Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN

INGEGNERIA ELETTRONICA, INFORMATICA E
DELLE TELECOMUNICAZIONI

Ciclo XXIV

Settore Concorsuale di afferenza: 09/F2 Telecomunicazioni
Settore Scientifico disciplinare: ING-INF/03 Telecomunicazioni

Wireless Sensor Networks for Monitoring Applications

Presentata da:
Cengiz Gezer

Relatore:
Chiar.mo Prof. Ing. Roberto Verdone

Tutore:
Chiar.mo Prof. Ing. Oreste Andrisano

Coordinatore Dottorato:
Chiar.mo Prof. Ing. Luca Benini

Esame finale anno 2012

Abstract

Wireless Sensor Networks (WSNs) are getting wide-spread attention since they became easily accessible with their low costs. One of the key elements of WSNs is distributed sensing. When the precise location of a signal of interest is unknown across the monitored region, distributing many sensors randomly/uniformly may yield with a better representation of the monitored random process than a traditional sensor deployment. In a typical WSN application the data sensed by nodes is usually sent to one (or more) central device, denoted as sink, which collects the information and can either act as a gateway towards other networks (e.g. Internet), where data can be stored, or be processed in order to command the actuators to perform special tasks. In such a scenario, a dense sensor deployment may create bottlenecks when many nodes competing to access the channel. Even though there are mitigation methods on the channel access, concurrent (parallel) transmissions may occur. In this study, always on the scope of monitoring applications, the involved development progress of two industrial projects with dense sensor deployments (eDIANA Project funded by European Commission and Centrale Adritica Project funded by Coop Italy) and the measurement results coming from several different test-beds evoked the necessity of a mathematical analysis on concurrent transmissions. To the best of our knowledge, in the literature there is no mathematical analysis of concurrent transmission in 2.4 GHz PHY of IEEE 802.15.4. In the thesis, experience stories of eDIANA and Centrale Adriatica Projects and a mathematical analysis of concurrent transmissions starting from O-QPSK chip demodulation to the packet reception rate with several different types of theoretical demodulators, are presented. There is a very good agreement between the measurements so far in the literature and the mathematical analysis.

Acknowledgements

This thesis would not have come to fruition without the guidance, help and support of a number of people. Foremost, I would like to express my gratitude to my PhD advisors Prof. Roberto Verdone, Dr. Chiara Buratti and Dr. Alberto Zanella for providing me insightful discussions and friendly working environment. My appreciation also goes to Prof. Adam Wolisz and Dr. Aitor Arriola for their valuable comments on the manuscript. I also owe thanks to undergraduate/graduate students, Sara Costanzini, Mirco Sapucci, Thomas Juhre, Flavio Fabbri, Marco Maretto, Yassine Nouir, Julian Alvarez, Alexandru Raileanu, and Danilo Abrignani who have contributed in various ways towards the making of the thesis. Furthermore to those nice people that I worked with through eDIANA, Centrale Adriatica, and SMAIL projects, thanks for your contributions on my personal and academical development. I also wish to thank to my colleagues, Flavia Martelli, Francesca Sorci, Silvia Zampese, Davide Visani, Ramona Rosini, Gianpiero Alcaro, Andrea Carniani, and Francesco Pantisano for the unforgettable moments in the office and the accompany to Garisenda at the lunch breaks. And special thanks go to all the rest of the people in CSITE and DEIS because of their positive attitudes towards me. Last, but not least, I would like to thank to my family for their unconditional support.

Contents

List of Figures **ix**

List of Tables **xiii**

1 Introduction **1**

I IEEE 802.15.4 **11**

2 Overview **13**

2.1 Device Types 14

2.2 Topologies 14

2.3 Physical Layer 15

2.4 Medium Access Layer 18

2.4.1 Operational Modes 19

2.4.2 CSMA-CA Algorithm 21

2.4.3 Data Transfer Model 22

2.4.4 Frame Structure 26

3 Preliminary Measurements **29**

3.1 Freescale MC13224 Platform 29

3.2 Received Signal Strength(RSS) and Energy Detection (ED) 30

3.3 Probability of Delay in slotted CSMA-CA 35

4 Capture Effect **41**

4.1 Related Work 42

4.2 Conditional Packet Capture Probability (CPCP) Measurements . . . 43
4.3 Mathematical Model . . . 46
4.3.1 The Model . . . 47
4.3.2 Extension: Capture Effect . . . 48
4.3.3 Numerical Results . . . 49
4.4 Validation . . . 51
4.5 Conclusion . . . 52

5 Concurrent Transmission **55**
5.1 Introduction . . . 55
5.2 System description . . . 57
5.3 Probability of Chip Error in Coherent O-QPSK . . . 59
5.3.1 Without Pulse Shaping . . . 60
5.3.1.1 Case: $\sqrt{I} < \sqrt{C/2}$ . . . 61
5.3.1.2 Case: $\sqrt{C/2} \leq \sqrt{I} < \sqrt{C}$ . . . 62
5.3.1.3 Case: $\sqrt{C} \leq \sqrt{I}$ . . . 63
5.3.2 Half-Sine Pulse Shaping . . . 63
5.3.2.1 case: $\sqrt{I} < \sqrt{C}$ . . . 67
5.3.2.2 case: $\sqrt{I} \geq \sqrt{C}$ . . . 67
5.3.3 Validation of the Analytical Model through Simulations . . . 68
5.4 Probability of Chip Error in Non-Coherent O-QPSK . . . 69
5.4.0.1 **Synchronous Symbols of Interferer** . . . 71
5.4.0.2 **Asynchronous Symbols of the Interferer** . . . 73
5.4.1 Validation of the Analytical Model through Simulations . . . 74
5.5 Alternative Demodulator 1 . . . 75
5.6 Alternative Demodulator 2 . . . 78
5.7 Probability of Data Error in IEEE 802.15.4 . . . 80
5.7.1 Chip Error Rate to Data Symbol Error Rate . . . 81
5.7.1.1 Coherent O-QPSK . . . 82
5.7.1.2 Non-coherent O-QPSK . . . 83
5.7.2 Packet Reception Rate . . . 84
5.8 Conclusion . . . 88

6 IEEE 802.11 Signal Strength Detection **89**

6.1 Framework and Methodology 90

6.2 Calibration 94

II ZigBee **99**

7 Overview **101**

7.1 ZigBee Basics 101

7.1.1 Device Types and Roles 103

7.1.2 Topologies 104

7.1.3 Self-Forming and Self-Healing 105

7.2 Network Layer 106

7.2.1 Addressing and Communication Mechanisms 108

7.2.2 Routing 110

7.2.2.1 Hierarchical (Tree) Routing 110

7.2.2.2 AODV Algorithm 111

7.2.2.3 Many-to-One 111

7.2.2.4 Source Routing 111

7.3 Application Layer 112

7.3.1 Application Support Sublayer 112

7.3.2 ZigBee Device Object (ZDO) 113

7.3.2.1 Device and Service Discovery 114

7.3.2.2 Binding Table Management 115

7.3.2.3 Network Management 115

7.3.3 Application Framework 115

7.3.4 ZigBee Cluster Library 117

7.3.4.1 Client / Server Model 117

7.3.4.2 Service Discovery 118

7.3.4.3 General Commands 120

7.3.5 Application Profiles 122

8 Mesh Routing in a Two Dimensional Grid: Centrale Adriatica Project **125**

8.1 System Description 125

8.2 Measurement Cycle 127

8.2.1 Synchronization 127

8.2.2 Link and Path Discovery 131

8.2.3 Reporting 135

8.3 Conclusion 135

9 Query Strategies in Application Layer: eDIANA Project **137**

9.1 eDIANA Scenario 138

9.2 eDIANA Reference Architecture 140

9.3 ZigBee in eDIANA Scenario 142

9.3.1 Intelligent Embedded Interface (iEi) 142

9.3.2 Classification of Applications 144

9.3.2.1 Monitoring Applications 144

9.3.2.2 Control Applications 146

9.3.3 ZigBee Profiles 147

9.4 ZigBee Driver on CDC 148

9.5 Demonstrator 151

9.6 Query Measurements 155

9.6.1 Hopping Time over the ZR 156

9.6.1.1 One-Hop Latency 157

9.6.1.2 Two-Hop Latency 157

9.6.2 QBS in Fixed Topology 159

9.6.3 EDS in Fixed Topology 160

9.6.4 Mesh Topology 162

9.7 Conclusions 164

10 Conclusion **165**

A 802.15.4 Transceiver Comparison **167**

A.1 TinyOS Compatible 167

A.1.1 Atmel Platforms compatible with TinyOS 168

A.1.2 Texas Instruments Platform compatible with TinyOS: TELOS . 169

A.2 Other Solutions 169

A.2.1 Ember and ST Microelectronics Platform 170

A.2.2 Freescale Platform 170

A.2.3 Texas Instruments Platforms 170

References **173**

List of Figures

1.1 Dense Sensor Deployments Result with Concurrent Transmissions 3
1.2 The Relationship between the Chapters of Part I 6
1.3 The Relationship between the Chapters of Part II 7

2.1 ZigBee/ IEEE 820.15.4 protocol stack architecture 13
2.2 Star Topology 15
2.3 Peer-to-peer Topology 16
2.4 PHY protocol data unit 17
2.5 Superframe 19
2.6 Superframe with Contention Free Period (CFP) 19
2.7 General Structure of the Superframe 20
2.8 Operational Modes 21
2.9 CSMA-CA Mechanism 23
2.10 Data Transfer to a Coordinator in Beacon-enabled Mode 24
2.11 Data Transfer to a Coordinator in non Beacon-enabled Mode 24
2.12 Data Transfer from a Coordinator in Beacon-enabled Mode 25
2.13 Data Transfer from a Coordinator in non Beacon-enabled Mode 25
2.14 MAC Protocol Data Unit in General 26
2.15 Beacon Frame 27
2.16 Acknowledgement Frame 28

3.1 MC1322x Family Block Diagram 30
3.2 MC1322x Transceiver 31
3.3 Measurement Setup 32
3.4 P_{RX} - Distance Relationship 33

LIST OF FIGURES

3.5 P_{RX} Measurements When Varying the Distance 34
3.6 P_{RX} Measurements 36
3.7 Slotted CSMA-CA Example 37
3.8 Probabilities of Having the Beginning of a Packet in a Backoff Slot with 20-byte Packets 37
3.9 Probabilities of Having the Beginning of a Packet in a Backoff Slot with 40-byte Packets 38
3.10 Probabilities of Having the Beginning of a Packet in a Backoff Slot with 60-byte Packets 38

4.1 Experimental Setup for Two Interferers. 44
4.2 Example Timing for Two Interferers. 45
4.3 Conditional Packet Capture Probability. 46
4.4 p_s as a function of N. 50
4.5 Measurement Setup with Four Nodes. 51
4.6 (a) p_s values for 20 bytes packets, (b) p_s values for 40 bytes packets, (c) p_s values for 60 bytes packets 53

5.1 Impact of Interferer 61
5.2 Cases: **(a)** when $\sqrt{I} < \sqrt{C/2}$; **(b)** when $\sqrt{C/2} \leq \sqrt{I} < \sqrt{C}$; **(c)** when $\sqrt{C} \leq \sqrt{I}$ 62
5.3 Impact of Carrier Phase of Interferer 64
5.4 Impact of Asynchronous Symbols of Interferer 65
5.5 Coherent O-QPSK Demodulator Chip Error Rate 68
5.6 Phase Transitions of Complex Envelope 70
5.7 Non-coherent Chip Error Rate 75
5.8 $R(\omega t)$ diagram when $\phi_C = 0$ 77
5.9 Chip Error Rate 80
5.10 Packet Reception Rate 86

6.1 A View of the Warehouse 90
6.2 Layout of the Warehouse 91
6.3 Coexistence of 802.15.4 and 802.11 in 2.4 GHz ISM Band 92
6.4 ED Scan 93

6.5 ED Scan Timing . . . 93
6.6 Spectrum Measured nearby Sensor A8 . . . 94
6.7 Spectrum Analyzer Measurement nearby Sensor 88 . . . 95
6.8 802.11 and 802.15.4 Channel Spacing . . . 96
6.9 Measurements . . . 96
6.10 Sensor 99 - Channel 1 . . . 97
6.11 Sensor 88 - Channel 1 . . . 98

7.1 ZigBee Stack Architecture . . . 102
7.2 A Star Topology in ZigBee . . . 104
7.3 A Tree Topology in ZigBee . . . 105
7.4 A Mesh Topology in ZigBee . . . 106
7.5 ZigBee NWK Layer . . . 106
7.6 Types of Primitives in ZigBee . . . 107
7.7 Application Support Sublayer . . . 112
7.8 Simplified Client/Server Model . . . 118
7.9 Simplified Relationship between ZigBee Devices and Descriptors . . . 119
7.10 Service Discovery Example . . . 120
7.11 The Relationship between ZCL and Application Profiles . . . 122

8.1 ZigBee Networks . . . 126
8.2 Diagram of the System . . . 128
8.3 ZigBee Coordinator (ZC) and Single Board Computer (SBC) . . . 129
8.4 Node . . . 129
8.5 Cycle Timing . . . 129
8.6 Average Received Signal Strength (RSS) of *Write Attributes* Command Broadcast . . . 130
8.7 Received Signal Strength to Link Cost Map in BeeStack . . . 131
8.8 Relays . . . 133
8.9 Average Path Cost and Number of Hops . . . 134
8.10 Packet Reception Rate . . . 134

9.1 The eDIANA Project . . . 138
9.2 The eDIANA Hierarchy . . . 140

9.3 eDIANA Reference Architecture 141
9.4 Intelligent Embedded Interface (iEi) 143
9.5 Home Automation and Smart Energy Profiles Together 146
9.6 Cell Device Concentrator (CDC) 148
9.7 ZigBee Driver 149
9.8 POSIX Threads Implementation of ZigBee Driver 150
9.9 A Snapshot of ZigBee Driver User Interface 152
9.10 End Point Configuration 153
9.11 Flow Chart of On/Off Appliances 153
9.12 Flow Chart of Programmable Appliances 154
9.13 Flow Chart of Heating or Cooling Capable Appliances 154
9.14 Analysis of Hopping Time - One Hop 157
9.15 Analysis of Hopping Time - Two Hops 158
9.16 Synchronized Query 159
9.17 Group Query 160
9.18 Periodic Traffic 161
9.19 Query in Periodic Reporting ($\Delta T = 3s$) 162
9.20 Query in Periodic Reporting ($\Delta T = 200ms$) 162
9.21 An Instance of Mesh Routes 163
9.22 Mesh Measurements 164

A.1 Atmel Platforms compatible with TinyOS [52] 168
A.2 Comparison of the Platforms 171

List of Tables

1.1 Analogy between the Projects 5

2.1 PHY Alternatives 17
2.2 SHR in Different PHY Alternatives 18

3.1 Outdoor Measurement Results 32
3.2 Probability of Success 39

5.1 Useful and Interferer Chip Phase Combinations 74
5.2 Symbol Point Decision 76
5.3 Probabilities in Equation 5.26 77
5.4 Combinations of Different Errors 78
5.5 Permitted Sequences 79
5.6 Feature Mapping 79
5.7 Permitted Feature Vectors (Duration: $3T_c$) 80
5.8 Chip Sequences in 2450 MHz PHY of IEEE 802.15.4 82
5.9 Hamming Distances between the Data Symbols in Coherent Demodulation 83
5.10 Reoccurrences of Unique Hamming Distance Values 83
5.11 Phase Transitions of Chip Sequences in 2450 MHz PHY of IEEE 802.15.4 84
5.12 Hamming Distances between the Phase Transition Vectors of Data Symbols 85
5.13 Reoccurrences of Unique Hamming Distance Values 85

6.1 802.11 Spectrum Peak and RSS in Channel 1 95

7.1 ZigBee Stack Comparison 103
7.2 Type of Devices in ZigBee 104

LIST OF TABLES

7.3 NWK Layer Primitives 107
7.4 Address Types 108
7.5 Broadcast Addresses 110
7.6 APS Primitives 113
7.7 ZCL General Commands 121

9.1 Energy Saving Strategies in eDIANA 139
9.2 eDIANA Appliances and ZigBee Devices 147
9.3 Hopping Time Measurements 158

Acronyms

ADC Analog-to-Digital Converter

AES Advanced Encryption Standard

AF Application Framework

AIB Application Support Layer Information Base

AO Application Object

AODV Ad-Hoc On-Demand Distance Vector

AP Access Point

APDU Application Support Sublayer Data Unit

API Application Programming Interface

APL Application Layer

APS Application Support Sublayer

APSDE Application Support Sublayer Data Entity

APSME Application Support Sublayer Management Entity

AWGN Additive White Gaussian Noise

BI Beacon Interval

BO Beacon Order

BPSK Binary Phase Shift Keying

BTR Broadcast Transaction Record

c2MCCi Cell to MacroCell Concentrator Interface

CA Control Application

CAP Contention Access Period

CCA Cell Control and Actuation

CDC Cell Device Concentrator

CDF Cumulative Distribution Function

CER Chip Error Rate

CFP Contention Free Period

CGS Cell Generation and Storage

CMM Cell Monitoring and Metering

CMOS Complementary MetalOxideSemiconductor

CPCP Conditional Packet Capture Probability

CRC Cyclic Redundancy Check

CSMA-CA Carrier Sense Multiple Access with Collision Avoidance

CUI Cell User Interface

DSER Data Symbol Error Rate

DSSS Direct Sequence Spread Spectrum

DV Distance Vector

ED Energy Detection

eDIANA Embedded Systems for Energy Efficient Buildings

EDP eDIANA Platform

EDS Event-Driven Strategy

FCS Frame Check Sequence

FFD Full Function Device

GPIO General Purpose Input Output

GTS Guaranteed Time Slot

HA Home Automation

HAN Home Area Network

HC Health Care

HVAC Heating, Ventilation, and Air Conditioning

I2C Inter-Integrated Circuit Interface

iEi Intelligent Embedded Interface

ISM Industrial, Scientific and Medical

LGA Land Grid Array

LNA Low Noise Amplifier

ACRONYMS

LQI Link Quality Indication

LR-WPAN Low-Rate Wireless Personal Area Network

MA Monitoring Application

MAC Medium Access Control

MCC Macro-Cell Concentrator

MCS MacroCell Control Strategies

MCU Micro Controller Unit

MDG MacroCell Data Gathering

MEMS Micro-Electro-Mechanical System

MFR MAC Footer

MLME MAC Sublayer Management Entity

MPDU MAC Protocol Data Unit

MPU Micro Processor Unit

MS Manufacturer Specific

MSK Minimum Shift Keying

MTO Many-to-One

MUI MCC User Interface

NIB NWK Information Base

NLDE Network Layer Data Entity

NLME Network Layer Management Entity

NPDU Network Protocol Data Unit

NTP Network Time Protocol

NWK Network

OEM Original Equipment Manufacturer

O-QPSK Offset Quadrature Phase Shift Keying

P2P Peer-to-Peer

PA Power Amplifier

PAN Personal Area Network

PCS Power Consumption Sensor

PDF Probability Density Function

PER Packet Error Rate

PHR PHY Header

PHY Physical Layer

PiP Platform-in-Package

PLME Physical Layer Management Entity

PPDU PHY Protocol Data Unit

PRR Packet Reception Rate

PSDU PHY Service Data Unit

PSP Packet Success Probability

QBS Query-Based Strategy

QoS Quality of Service

RF Radio Frequency

RFD Reduced Function Device

ROM Read Only Memory

RREP Route Reply

RREQ Route Request

RSS Received Signal Strength

SAP Service Access Point

SBC Single Board Computer

SD Superframe Duration

SE Smart Energy

SER Symbol Error Rate

SFD Start-of-frame Delimiter

SHR Synchronization Header

SINR Signal-to-Interference-Plus-Noise Ratio

SIR Signal-to-Interferer Ratio

SNR Signal-to-Noise Ratio

SO Superframe Order

SPI Serial Peripheral Interface

SRAM Static Random Access Memory

UART Universal Asynchronous Receiver/Transmitter

VoIP Voice over Internet Protocol

WSN Wireless Sensor Network

ZC ZigBee Coordinator

ZCL ZigBee Cluster Library

ZDO ZigBee Device Object

ZDP ZigBee Device Profile

ZED ZigBee End Device

ZR ZigBee Router

ZTC ZigBee Test Client

Chapter 1

Introduction

The thesis study, in the scope of two industrial projects, Embedded Systems for Energy Efficient Buildings (eDIANA) and Centrale Adriatica, investigates in general the implementation of monitoring applications by using Wireless Sensor Networks (WSNs) and in particular concurrent transmissions that may occur in dense sensor node deployments. A WSN can be described as a network consisting of densely distributed autonomous devices (nodes), using sensors to cooperatively monitor physical or environmental conditions and Radio Frequency (RF) waves to send the monitored data to the base stations or coordinators [34]. The nodes in WSNs contain RF components, actuators, sensors and Complementary MetalOxideSemiconductor (CMOS) type electronic devices (interface and data fusion circuitry, specialized and general purpose signal processing units, and microcontrollers). These components are named together as Micro-Electro-Mechanical System (MEMS). The latest development on the MEMSs allows the production of low-cost, low-power, multifunctional sensor nodes. Today the availability of cheap sensor nodes enables the applications of distributed wireless sensing to be realized commercially. When it is compared with the traditional sensors, WSNs yields improved line of sight and Signal-to-Noise Ratio (SNR) because of its way of deployment and processing. Traditional sensors are deployed in the following two ways [71]:

- Large, complex sensor systems are usually deployed very far away from the phenomena to be sensed, and employ complex signal processing algorithms to separate targets from environmental noise.

- Carefully engineered network of sensors is deployed in the field, but individual sensors do not possess computation capability, instead they transmit time series of the sensed phenomena to one or more nodes which perform the data reduction and filtering.

On the other hand, WSNs are densely deployed either inside the phenomenon or very close to it and they are capable of carrying out simple computations. Three important concepts that constitute the paradigm of WSN are; distributed sensing, wireless communication and distributed processing [56].

- **Distributed Sensing:** The positions of sensor nodes need not be pre-determined through engineering calculations when the precise location of a signal of interest is unknown across the monitored region. Distributing them randomly may yield higher SNR, and improved line of sight than having a single very sensitive sensor in one particular location. It is obvious that a distributed network of sensors will collect significantly more information than a system relying on a single sensor.

- **Wireless Communication:** In WSN applications the environment being monitored generally lacks of infrastructure for communications or energy, therefore untethered nodes must be supplied from local and finite energy sources, as well as relying on wireless communication channels to send data packets to each other.

- **Distributed Processing:** Finite energy budget of the untethered nodes restricts the design of WSNs. Communications is a key energy consumer as the radio signal power in sensor networks drops off due to ground reflections because of mostly short antenna heights [56]. Therefore, it is desired to process data as much as possible inside the nodes to reduce the number of bits that is transmitted. Distributed processing shows itself as a solution to energy constraints in WSN.

WSNs can be built up from many different types of sensors which are able to monitor different kinds of ambient conditions such as [57]: temperature, humidity, vehicular movement, lighting conditions, pressure, soil makeup, noise levels, presence or absence of certain kinds of objects, mechanical stress levels on attached objects, current characteristics such as speed, direction, and size of an object, etc. Application areas of WSNs can be categorized considering the application type [34]: Military applications,

Environmental applications [41, 46], Health Applications [58, 92], Home Applications [32, 86], etc. In a typical WSN application the data sensed by nodes is usually sent to one (or more) central device, denoted as sink, which collects the information and can either act as a gateway towards other networks (e.g. Internet), where data can be stored in order to be accessed by final users, or to be processed in order to command the actuators to perform specific tasks. When the network is densely deployed since many nodes are competing for the channel access, problems such as how to query the nodes to collect the data and how to route the packets to the sink, arise. In the thesis study we are trying to give some answers to these questions. For instance in Chapter 9, query strategies in the application scenario of eDIANA Project, and in Chapter 8, routing in a two dimensional grid deployment in Centrale Adriatica Project are considered. In fact, these application and network layer problems are closely related with a fundamental problem in the physical layer, which is called concurrent transmissions (Chapter 5). When there are many nodes that are periodically sending data to the direction of sinks, concurrent (parallel) transmissions can occur with a high probability. The relationship between these chapters can be summarized in Fig. 1.1.

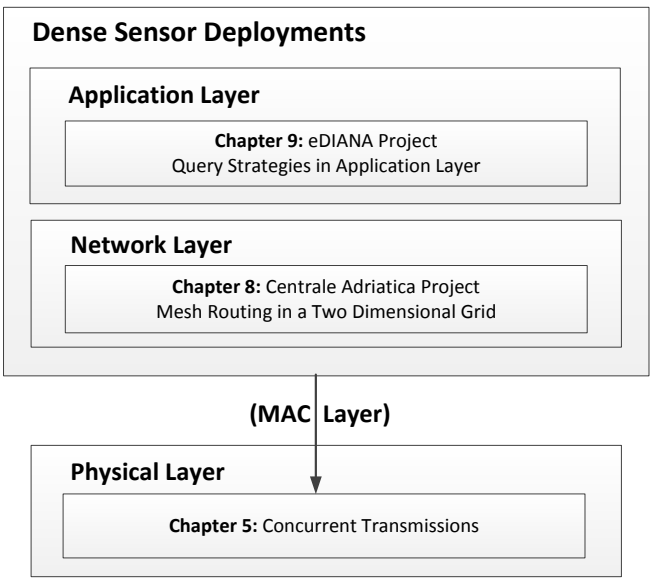

Figure 1.1: Dense Sensor Deployments Result with Concurrent Transmissions

We are not establishing a direct link between the concurrent transmission problem in the physical layer with the above mentioned problems in the upper layers since this kind of study highly depends on the implementation of the monitoring application, Network (NWK) and Medium Access Control (MAC) layers, therefore it is not straightforward and generic, but Chapter 8 and 9 strongly suggest us to separately investigate this fundamental problem as we did in Chapter 5.

The eDIANA Project aims to achieve more efficiency in the use of resources, by prioritizing energy as a scarce resource, through the deployment of WSNs in the building environments. In Europe, for instance, buildings are responsible for 40% of total energy consumption [51], which is more than the demand of industry or transportation. eDIANA Project [9], funded by the European Commission within FP7 through the ARTEMISIA framework, addresses the need of energy efficiency in buildings. Probably the most important innovation in the energy management put forward by eDIANA can be mentioned as the monitoring the real-time energy consumption of each appliance in order to implement smart algorithms. The eDIANA Platform (EDP) monitors the energy consumptions using a WSN. Each node embedded in the appliances periodically sends energy consumption of the appliance to a sink to be processed on the decision taking of energy saving algorithms. In this way, rather than having one fiscal electricity meter for each contract with utility provider, there will be power consumption sensors in each appliance in the premises with the intention that the high resolution will result in better algorithm implementation on the efficient use of energy. On the other hand, in the Centrale Adriatica Project, that is funded by Centrale Adriatica, the quality of the WiFi coverage in the high rack storage area of a warehouse is monitored by 228 battery-powered ZigBee-compliant nodes deployed as a planar grid throughout the warehouse. The WiFi network provides Voice over Internet Protocol (VoIP) and data services to the mobile operators working in the warehouse, therefore loss of coverage means loss of labor. The warehouse contains grocery stocks for supplying several stores, hence materials such as plastic, metal cans, bottles, liquids, all having different dielectric properties are continuously fed and withdrawn, so that the reflective/refractive characteristics of the environment change over time. This results in coverage quality being dynamic. Thanks to the two dimensional dense grid deployment of sensors, the regions in the warehouse where the signal quality is not sufficient can be identified in

real-time in order to take immediate actions against the loss of efficiency on the coordination of the workers. Both projects aim the efficient use of resources by exploiting the distributed sensing property, which allows better representation or better spatial distribution of the monitored phenomenon. Other two properties of WSNs, wireless communication and distributed processing, are tools on this purpose due to lack of infrastructure. Table 1.1 summarizes the analogy between these two projects.

Table 1.1: Analogy between the Projects

Project	**Distributed Sensing**	**Efficient Use of**
eDIANA	• Energy consumption of the appliances • Presence • Light Level • ...	Energy
Centrale Adriatica	• WiFi coverage	Labor

In the thesis study inside the conceptual borders of WSNs, with a particular focus on densely deployed monitoring applications, starting from Physical Layer (PHY) to Application Layer (APL), the performance determining factors such as Packet Error Rate (PER) and probability of success in channel access, capture effect, concurrent transmission, latency, routing and the different data query scenarios, were discussed. The thesis study is structured into two parts, namely IEEE 802.15.4 and ZigBee. These two parts complement each other resulting with a complete stack architecture, including medium access, routing and all principal tools on the application layer for a monitoring implementation. The performance of a monitoring application can not be isolated easily in a particular layer without considering the mutual impacts of all other layers, therefore the thesis study follows a step by step approach until the APL each time by adding another layer on the top of the previous layer. Part I starts with an overview of the IEEE 802.15.4 Standard with Chapter 2. Then, the hardware platform, Freescale MC1322x, that is used throughout the thesis, is introduced in Chapter 3 with the preliminary measurements on RSS, PER, and channel access probability. Then, Chapter 4 describes a mathematical model of Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) and an extension to it while including the impact of the capture effect

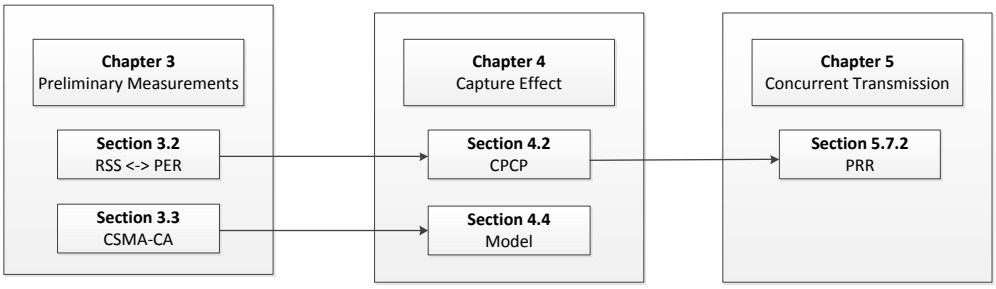

Figure 1.2: The Relationship between the Chapters of Part I

to the scenario. Chapter 5 mathematically investigates the concurrent transmission in Offset Quadrature Phase Shift Keying (O-QPSK) while giving coherent and non-coherent models, considering the effect of pulse shaping and spreading in IEEE 802.15.4. Finally Part I finishes with Chapter 6, which discusses how to measure the RSS in 802.11 by using the energy levels measured in 2.4 GHz PHY of IEEE 802.15.4 as a part of Centrale Adriatica Project. In Part I, there is a strong relationship between Chapter 3, Chapter 4, and Chapter 5 as shown in Fig. 1.2. The measurements in Section 3.2 are significant to demonstrate that RSS is a reliable metric on the PER in MC1322x Platform, which serves to Section 4.2 as the basis of the reliability of the measurement setup that is used in order to obtain the necessary Signal-to-Interferer Ratio (SIR) in concurrent transmission. This measurement setup later on in Section 5.7.2, compared with the concurrent transmission analysis in Chapter 5. The measurements coming from Section 4.2 and the concurrent transmission analysis in Chapter 5 agree on the minimum necessary SIR to be able to start receiving a packet even though the noise is neglected in the mathematical analysis for the sake of simplicity. Also measurement results coming from Section 3.3 are used to validate the model in Section 4.4.

The ZigBee Part, on the other hand, starts with an overview of ZigBee with Chapter 7. The ZigBee Cluster Library (ZCL) which provides tools in order to implement different kinds of monitoring applications are described in details in Section 7.3.4 of Chapter 7 since in the following two chapters, the application scenarios are implemented by using the ZCL. Therefore Section 7.3.4 is important as being the application framework of Chapter 8 and Chapter 9. Later in Chapter 8 Centrale Adriatica Project scenario

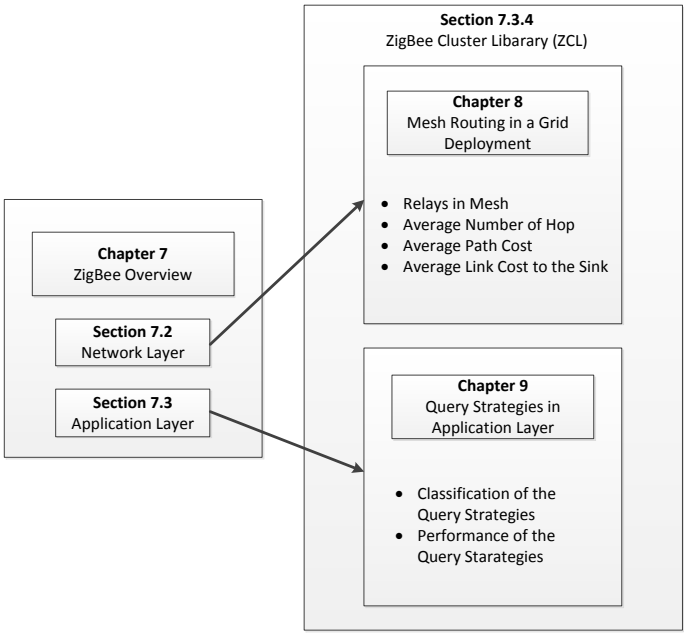

Figure 1.3: The Relationship between the Chapters of Part II

and the routing in ZigBee are covered as an example of the ZigBee NWK layer performance in a large scale (228 nodes) planar two dimensional grid deployment. Used relays, average number of hops and the average path costs for each node to reach the sink presented in Chapter 8 to give insights on the Distance Vector (DV) routes that can be formed in a WSN. Finally in Chapter 9 a typical real-life monitoring application scenario, eDIANA Project scenario on the energy saving and resource management of household/office appliances, is described. In the same chapter a systematical classification of the query strategies in monitoring applications is given in Section 9.3.2 and for each strategy performance is measured in a eDIANA compatible test-bed in Section 9.6. Furthermore in Chapter 9 the developed driver in order to give basic controls over the ZigBee network to a ZigBee gateway, which is refereed as Cell Device Concentrator (CDC) in the chapter, is explained. The relationship between the chapters of Part II can be summarized as in Fig. 1.3. In both Chapter 8 and Chapter 9, the conclusion is concurrent transmissions may become significant even though there are mitigation methods in MAC layer, therefore on the analysis of synchronized queries or dense sensor deployments, concurrent transmissions should be considered initially (see Fig. 1.1).

During the thesis activity I have collaborated with several graduate and undergraduate students. The embedded software for the nodes in Chapter 3, Chapter 4, Chapter 8 has been completely developed by me. Chapter 4 is an outcome of a conference paper published with Dr. Chiara Buratti and Prof. Roberto Verdone. One of my main efforts in the thesis study was the mathematical analysis in Chapter 5, under the supervision of Dr. Alberto Zanella and Prof. Roberto Verdone. The method used in Chapter 6 was an outcome of my activity in Centrale Adriatica Project under the supervision of Prof. Verdone. During the activity of eDIANA Project described in Chapter 9, Dr. Chiara Buratti supervised me. The software design and development for the related tasks of eDIANA Project has been performed by me autonomously. The ZigBee driver that I have developed was used by the partners in their energy saving demonstrator sites. In the last year of the thesis activity Centrale Adratica Project was the highest priority. The software development has been realized by me but for the software tests I received help from the thesis students, Marco Maretto, Alexandru Raileanu and Danilo Abrignani. During the three years activity of eDIANA Project, especially for the experimental activity I received help from the thesis students Thomas Juhre, Yassine Nouir and Julian Alvarez. Also for the measurements in Chapter 3, I received help from thesis students, Sara Costanzini and Mirco Sapucci. Some parts of the thesis is published in conferences papers and project deliverables but Chapter 5, Chapter 6, and Chapter 8 have not yet been published (2012). I believe the methodology on the analysis of concurrent transmission in Chapter 5, to the best of our knowledge is the first in the literature, besides quite general with a potential to be repeated for the other systems.

Conference Papers:

- F. Fabbri, C. Gezer, and R. Verdone, The Impact of Realistic Footprint Shapes on the Connectivity of Wireless Sensor Networks, in Proceedings of the 1st International Workshop on Performance Methodologies and Tools for Wireless Sensor Networks (WSNPerf), Oct. 23, 2009, Pisa, Italy.

- C. Gezer; C. Buratti; A. Visconti; R. Ukmar; and R. Verdone, Zigbee-Based Platform for Energy Efficient Buildings, 7th European Conference on Wireless Sensor Network, EWSN 2010, Feb. 17-19 2010, Coimbra, Portugal.

- C. Gezer, C. Buratti, and R. Verdone, Capture Effect in IEEE 802.15.4 Networks: Modelling and Experimentation,International Symposium on Wireless Pervasive Computing, ISWPC 2010 , May. 5-7, 2010, Modena, Italy.

- C. Gezer, M. Niccolini, and C. Buratti, An IEEE/ZigBee Based Wireless Sensor Network for Energy Efficient Buildings, International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob, 2010, Oct. 11-13, 2010, Niagara Falls, Canada.

- Flavio Fabbri, Chiara Buratti, Cengiz Gezer, Paolo Toppan, Andrea Toppan, Roberto Verdone, Enhancing Wi-Fi Coverage Through ZigBee Mesh Network of Energy Scan Devices , 8th ACM Conference on Embedded Networked Sensor Systems, SenSys 2010, Nov. 3-5 2010, Zurich, Switzerland.

- C. Gezer, C. Buratti, A ZigBee Smart Emergy Implementation for Energy Efficient Buildings, IEEE 73rd Vehicular Technology Conference, VTC-Spring, May. 15-18, 2011, Budapest, Hungary.

Project Deliverables:

- "D.1.1-A Baseline Analysis Result" deliverable for eDIANA Project, edited by R. Socorro (Acciona), May. 31, 2009

- "D2.1-B eDIANA Reference Architecture" deliverable for eDIANA Project, edited by R. Ukmar (ST Microelectronics), Jan. 31, 2010

- "D2.2-B Software Tools for Run-Time Discovery of Services and Devices" deliverable for eDIANA Project, edited by M. J. Martinez (I&IMS), Jan. 31, 2010.

- "D2.3-A Network Topology and Communications Architecture Definition", deliverable for eDIANA Project, edited by C. Buratti (UniBo), Jan. 31, 2010.

- "D3.2-A Intelligent Embedded Interface (iEi) - Concept Release", deliverable for eDIANA Project, edited by J.U. Garcia (Fagor), Apr. 1, 2010

- "D2.2-E Adapted Sensor Collaboration Middleware" deliverable for eDIANA Project, edited by M. J. Martinez (I&IMS), Jul. 31, 2010.

- "D2.3-B Communication Protocol Specification", deliverable for eDIANA Project, edited by C. Buratti (UniBo), Jul. 31, 2010.
- "D1.4-B Reference Scenarios", deliverable for eDIANA Project, edited by B. V. D. Heuvel (Philips Apptech), Jul. 31, 2010
- "D3.6-A Communication Middleware for Intelligent Embedded Interface", deliverable for eDIANA Project, edited by C.Gezer (UniBo), Oct. 31, 2010
- "D3.6-B High Level Communication Infrastructure Concept Report", deliverable for eDIANA Project, edited by I. Karls (Intel), Oct. 31, 2010
- "D7.1-A Demo Lab Testing Methodology", deliverable for eDIANA Project, edited by J. M. Marcos (Fagor), May. 31, 2011
- "D7.2-A Interim Test results", deliverable for eDIANA Project, edited by R. Socorro (Acciona), Jul. 31, 2011
- "D8.1-A Report on the eDIANA Platform Instantiation for Energy Saving", deliverable for eDIANA Project, edited by T. V. Craenendonck (Philips Consumer Lifestyle), Oct. 30, 2011
- "D8.1-B Report on the eDIANA Platform Instantiation for Energy Saving", deliverable for eDIANA Project, edited by T. V. Craenendonck (Philips Consumer Lifestyle), Jan. 31, 2012

Part I

IEEE 802.15.4

Chapter 2

Overview

The IEEE 802.15.4 [93] standard appeared in November 2003 to provide ultra low complexity, ultra low power consumption and low cost wireless networking solution in low data rate wireless networks. The IEEE 802.15.4 protocol specifies MAC and PHY for Low-Rate Wireless Personal Area Networks (LR-WPANs). Even though this standard was not specifically developed for WSNs, it is suitable for them, satisfying the requirements on power consumption and communications. The IEEE 802.15.4 protocol is deeply connected with the ZigBee specifications [36]. The ZibBee Alliance has been working together with IEEE (task group 4) in order to specify a full protocol stack for low cost, low power, low data rate wireless network protocol. The model of the ZigBee/IEEE 802.15.4 protocol architecture can be seen in Fig 2.1.

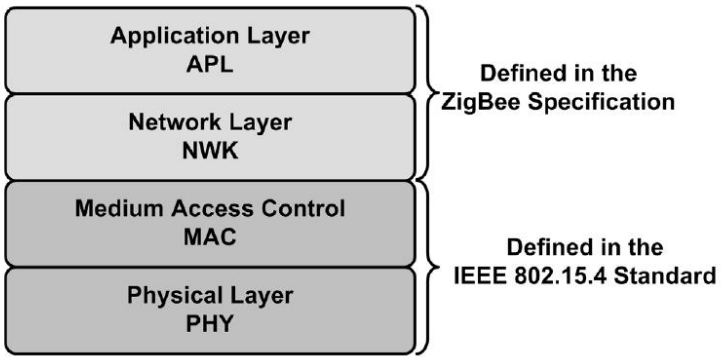

Figure 2.1: ZigBee/ IEEE 820.15.4 protocol stack architecture

2.1 Device Types

IEEE 802.15.4 defines two types of devices: the Full Function Device (FFD) and the Reduced Function Device (RFD). The FFD contains the complete set of MAC services and can operate in three modes:

- Personal Area Network (PAN) Coordinator: In this mode FFD forms its own network to where other devices can associate. Every PAN must include at least and at most one FFD acting as the PAN coordinator. PAN Coordinator may provide synchronization services to its own network and may act like a sink.
- Coordinator: The device that works in this mode acts like a local coordinator that must be a part of a PAN previously formed by a PAN Coordinator. The coordinator may provide synchronization services by transmitting beacons.
- Simple Device: The device that does not have the previously described two roles but having the capability to own one of these roles.

The RFD contains a reduced set of MAC services to allow simple devices to be part of the network with minimal resources and memory capacity. RFD can operate only as a simple device intended for applications that are extremely simple, such as a light switches, assive infrared sensors, etc. A RFD can only associate to a FFD at a time.

2.2 Topologies

Two basic topologies are allowed, but not completely described by the standard since definition of higher layer functionalities are out of the scope of 802.15.4. First one is the star topology, formed around an FFD acting as a PAN coordinator, which is the only node allowed to form links with more than one device as shown in Fig. 2.2. The communication can just be established between the devices and the coordinator; each device joined the network and willing to communicate with other devices must send its data to the PAN coordinator (Because of being always on, it is a good practice to employ mains powered devices as the PAN coordinator). Star topology is preferable in case the coverage area is small and low latency is required by the application. In the standard recommended applications for star topology are listed as: home automation, personal computer peripherals, toys and games.

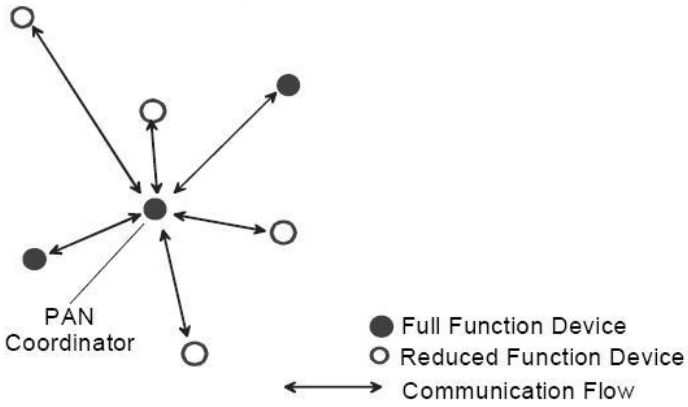

Figure 2.2: Star Topology

The Peer-to-Peer (P2P) topology as shown in Fig. 2.3 differs from the star topology since any device can communicate with each other as long as they are in their range of transmission. Every LR-WPAN must have a PAN coordinator; however PAN coordinator in P2P topology does not have centralized tasks such as in star topology. P2P topology allows more complex network formations to be implemented. P2P topology is preferable in case a large area should be covered and latency is not a critical issue. Applications such as industrial control and monitoring, wireless sensor networks, asset and inventory tracking, intelligent agriculture, and security, can benefit from such a network topology. This topology allows the formation of more complex networks and permits any FFD to communicate with any other FFD behind its transmission range via multiple hops. Each device in a P2P topology first needs to find the route to the destination device. Once the destination device is found, these two devices can exchange data by sending the packets passing over the relays. However, the multi-hop communications requires additional device memory for routing tables. IEEE 802.15.4 can also support other network topologies, such as cluster-trees which are not part of the IEEE 802.15.4 standard, but are described in the ZigBee Alliance specifications.

2.3 Physical Layer

The IEEE 802.15.4 physical layer uses spread spectrum techniques, and it comes in two ISM band alternative; one designed to operate in the 868 MHz and 915 MHz ISM

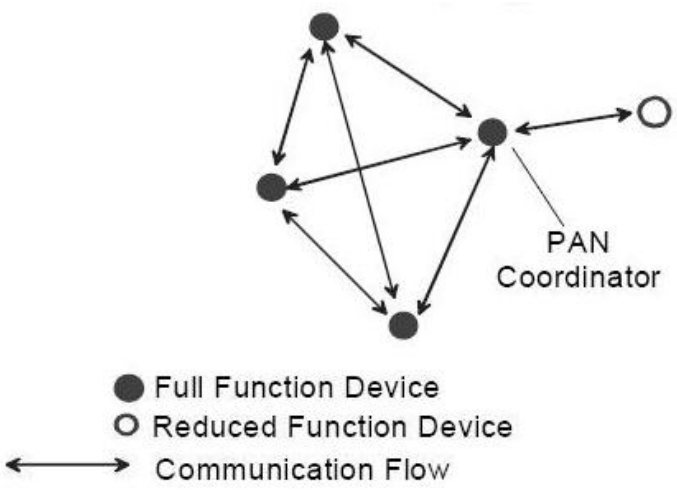

Figure 2.3: Peer-to-peer Topology

bands, and the other designed to operate in the 2.4 GHz ISM band. The lower frequency band utilizes Binary Phase Shift Keying (BPSK) modulation to achieve data rates of 20 kbps over a single channel, and 40 kbps over 10 channels. The higher frequency band utilizes O-QPSK modulation to achieve a data rate of 250 kbps over 16 channels. There are optional PHYs in 868/915 MHz Industrial, Scientific and Medical (ISM) band defined in 2006 revision of the standard. The 802.15.4 PHY operates in three different unlicensed bands as shown in Table 2.1. Common PHYs in both 2003 and 2006 versions are:

- The 868 MHz band, ranging from 868.0 and 868.6 MHz and used in the European area, implements a raised-cosine-shaped BPSK modulation format, with Direct Sequence Spread Spectrum (DSSS) at 300 Kchip/s. Only a single channel with a data rate of 20 kbps is available.

- The 915 MHz band, ranging between 902 and 928 MHz and used in the North American and Pacific area, implements a raised-cosine-shaped BPSK modulation format, with DSSS at 600 Kchip/s. Ten channels are available with a data rate of 40 kbps.

- The 2.4 GHz ISM band, which extends from 2400 to 2483.5 MHz and is used worldwide, implements a half-sine-shaped O-QPSK modulation format, with DSSS at 2000 Kchip/s. Sixteen channels are available with a data rate of 250 kbps.

Table 2.1: PHY Alternatives

Band (MHz)	Chip Rate (kchip/s)	Modulation (kb/s)	Bit Rate (ksymbol/s)	Symbol Rate (Ksymbol/s)	Number of Channels
868-868.6	300	BPSK	20	20	1
902-928	600	BPSK	40	40	10
868-868.6 (optional)	400	ASK	250	12.5	1
902-928 (optional)	1600	ASK	250	50	10
868-868.6 (optional)	400	O-QPSK	100	25	1
902-928 (optional)	1000	O-QPSK	250	62.5	10
2400-2483.5	2000	O-QPSK	250	62.5	16

PHY is the lowest protocol layer. It is in charge of activation and deactivation of the radio transceiver, Energy Detection (ED) within the current channel, Link Quality Indication (LQI) for received packets, Clear Channel Assessment (CCA) for CSMA-CA, channel frequency selection, data transmission and reception. Fields of PHY Protocol Data Unit (PPDU) are shown in Fig.2.4. Maximum PPDU size is specified as 133 bytes in the standard.

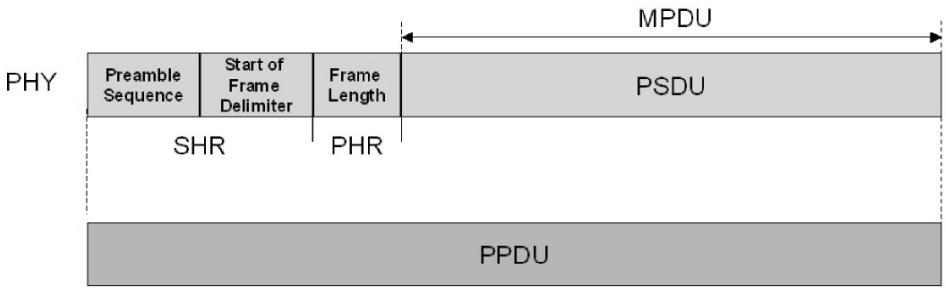

Figure 2.4: PHY protocol data unit

On the header of PPDU there are two parts named as Synchronization Header (SHR) and PHY Header (PHR). SHR has two fields namely preamble and Start-of-frame Delimiter (SFD). These two fields are dependent on the used band and the modulation as summarized in Table 2.2. The PHY Service Data Unit (PSDU) serves as the protocol data unit for the next higher layer, MAC.

Table 2.2: SHR in Different PHY Alternatives

PHY	Preamble Length		SFD Length	
868868.6 MHz BPSK	4 bytes	32 symbols	1 byte	8 symbols
902928 MHz BPSK	4 bytes	32 symbols	1 byte	8 symbols
868868.6 MHz ASK	5 bytes	2 symbols	2.5 bytes	1 symbol
902928 MHz ASK	3.75 bytes	6 symbols	0.625 bytes	1 symbol
868868.6 MHz O-QPSK	4 bytes	8 symbols	1 byte	2 symbols
902928 MHz O-QPSK	4 bytes	8 symbols	1 byte	2 symbols
24002483.5 MHz O-QPSK	4 bytes	8 symbols	1 byte	2 symbols

2.4 Medium Access Layer

The MAC layer of the IEEE 802.15.4 protocol provides an interface between the physical layer and the higher layer protocols. The MAC layer handles all access to the physical radio channel and is responsible for the tasks such as generating network beacons if the device is a coordinator, synchronizing to the beacons, supporting PAN association and disassociation, supporting device security, employing the CSMA-CA mechanism for channel access, handling and maintaining the Guaranteed Time Slot (GTS) mechanism, providing a reliable link between two peer MAC entities. The MAC layer defines two different access to the channel: beacon and non beacon-enabled. In both cases a contention-based protocol, based on CSMA-CA is implemented. MAC layer supports star and P2P topologies, giving flexibility to the application. Each PAN has a PAN coordinator and other devices may associate or disassociate at will. The 16-bit addressing mode allows the networks up to 65535 devices. IEEE 802.15.4 uses a protocol based on the CSMA-CA algorithm, which requires listening to the channel before transmitting to reduce the probability of collisions with other ongoing transmissions.

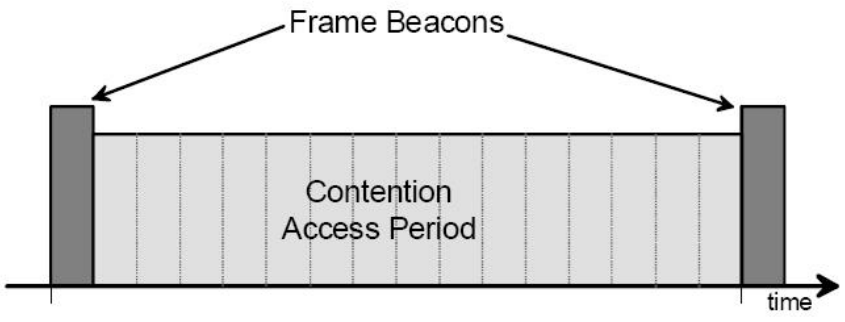

Figure 2.5: Superframe

2.4.1 Operational Modes

IEEE 802.15.4 defines two operational modes, namely the beacon-enabled and the non beacon-enabled mode, which correspond to two different channel access mechanisms. In the non beacon-enabled mode nodes use an unslotted CSMA-CA protocol to access the channel and transmit their packets. In the beacon-enabled mode, instead, the access to the channel is managed through the superframe structure, starting with the beacon transmitted by PAN Coordinator as shown in Fig. 2.5. When the beacon-enabled mode is selected, coordinator uses a periodic superframe structure to manage the communications between the devices. The superframe is bounded by frame beacons. Beacon transmissions can also occur during the association therefore to identify this use of beacon the name frame beacon is used. The format of this structure is determined by the coordinator and transmitted to other devices inside the beacon frame. The superframe is divided into 16 equally sized slots.

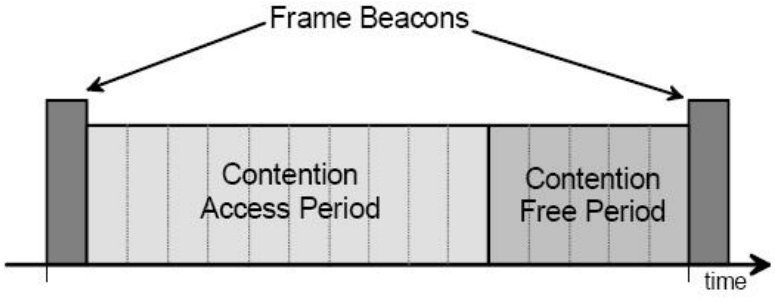

Figure 2.6: Superframe with Contention Free Period (CFP)

In order to offer some Quality of Service (QoS), a Contention Free Period (CFP) is defined by the Standard. The CFP consists of GTSs that may be allocated by the PAN coordinator to the devices that can need low-latency or specific data bandwidth. CFP is a part of the superframe and starts immediately after Contention Access Period (CAP), as shown is Fig.2.6. Use of GTS is optional and a maximum number of seven GTSs can be allocated.

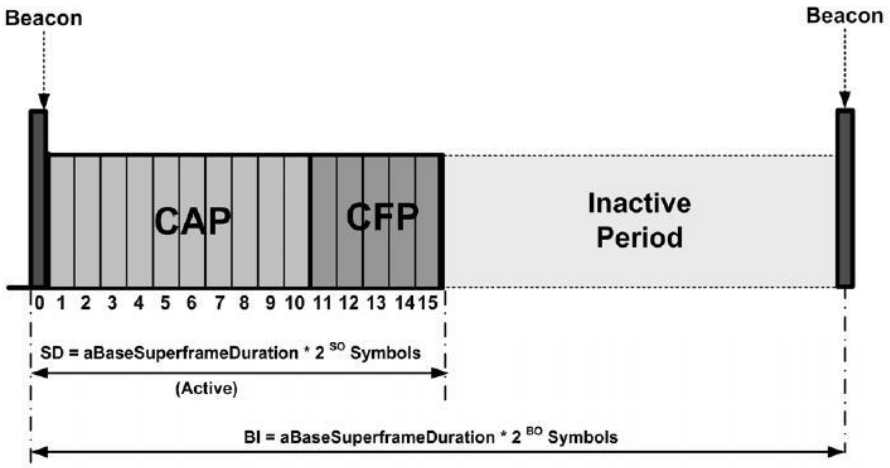

Figure 2.7: General Structure of the Superframe

The superframe may contain an inactive part as shown in Fig. 2.7, allowing nodes to go to sleeping mode. In the inactive part coordinator also goes to sleep like the nodes. The structure of a superframe is defined by two parameters called Beacon Order (BO) and Superframe Order (SO):

- BO (macBeaconOrder): describes the time interval between the beacon frames. The value of the BO and the Beacon Interval (BI) are related as the following:

$$BI = aBaseSuperframeDuration * 2^{BO}$$

- SO (macSuperframeOrder) : describes the length of the active portion of the beacon interval. The value of the macSuperframeOrder and the Superframe Duration (SD) are related as follows:

$$SD = aBaseSuperframeDuration * 2^{SO}$$

where aBaseSuperframeDuration is defined as $15,36ms$ in the standard. If $SO = BO$ then $SD = BI$, and there is no inactive period in the superframe structure. The permitted values for these two parameters should satisfy $0 \leq SO \leq BO < 15$ in the beacon enabled mode. if $SO = 15$ this means there is no superframe and the network operates in non beacon-enabled mode. In summary, the active part of the superframe is divided into two parts named CAP and CFP. CFP is composed of GTSs that can be allocated by the coordinator to specific nodes. All transmissions must be finished before the end of the CAP except the devices having a GTS. The device that will use the GTS must ensure that its packet transmission fits to the allocated slot or slots. In Fig. 2.8 all possible ways to access the channel is summarized.

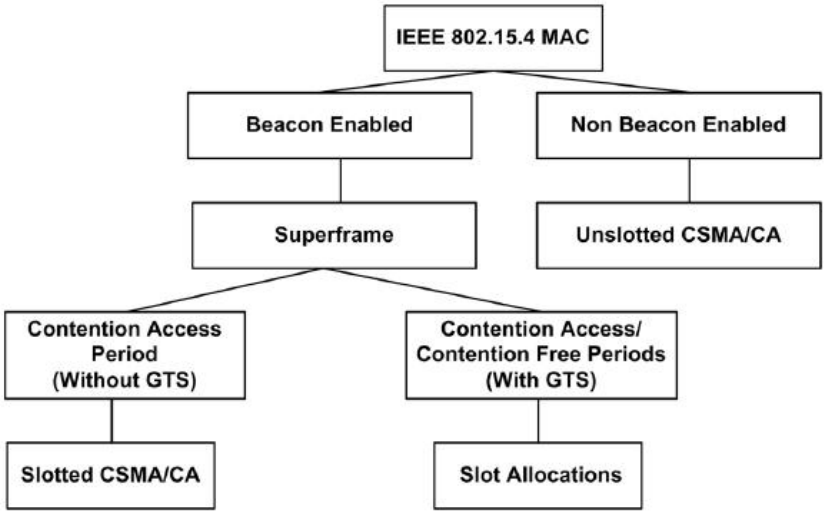

Figure 2.8: Operational Modes

2.4.2 CSMA-CA Algorithm

The CSMA-CA algorithm is implemented using units of time called backoff periods,T. There are slotted and unslotted versions but in general they are very similar except the differences on the synchronization, parameter initialization and the number of sensing phase. In both versions after the parameter initialization, each node waits for a random number of backoff periods. After that, channel sensing is performed, if the channel is found free the node immediately starts the transmission; if, instead, the channel is busy the node enters again the backoff state. There exists a maximum number of

attempt the node can try to sense the channel. When this maximum is reached the algorithm ends with a failure. Flowchart of the algorithm is found in Fig. 2.9. In the algorithm each node maintains two variables for each transmission attempt: NB and BE. NB is the counter for transmission attempts which is always initialized to 0 at the beginning and cannot be larger than $macMaxCSMABackoffs$ (default: 4). BE is the backoff exponent related to the maximum number of backoff periods a node will wait before attempting to assess the channel's availability. BE is initialized to the value of $macMinBE$ (default:3), and cannot assume values larger than $macMaxBE$ (default:5). Figure 2.9 illustrates the steps of the CSMA-CA algorithm. First, NB and BE are initialized and then the algorithm randomly picks an integer number form the (0, 2^{BE}-1) interval and waits as that many backoff slots [step (2)] , then channel sensing is performed in the following backoff slot [step (3)]. If the channel is assessed to be busy [step (4)], the MAC sublayer increases both NB and BE by one, ensuring that BE is not larger than $macMaxBE$. If the value of NB is less than or equal to $macMaxCSMABackoffs$, the CSMA-CA algorithm returns to step (2). If the value of NB is larger than $macMaxCSMABackoffs$, the CSMA-CA algorithm terminates with a failure. If the channel is assessed to be idle [step (5)], in unslotted version the MAC layer transmits the packet immediately with a return of success while in the slotted version it senses the channel one more backoff slot before sending the packet.

2.4.3 Data Transfer Model

In IEEE 802.15.4 three different ways of data transfer is possible; data transfer to a coordinator, data transfer from a coordinator, data transfer between two peer devices.

Data Transfer to a Coordinator: In a beacon-enabled network the device that wants to send a packet to the coordinator first listens to the network beacon. When the beacon is received, the device synchronizes itself to the superframe structure defined in this beacon. If the device has a GTS assigned, it waits until appropriate moment indicated in the beacon frame to transmit the data, if not the device transmits its data frame in the CAP with slotted CSMA-CA. If it is requested, the coordinator acknowledges the successful reception of the data by transmitting an acknowledgment frame. This sequence is shown in Fig. 2.10.

On the other hand, in a non beacon-enabled network the device transmits its data frame destinated to the coordinator, using unslotted CSMA-CA. If it is requested, the

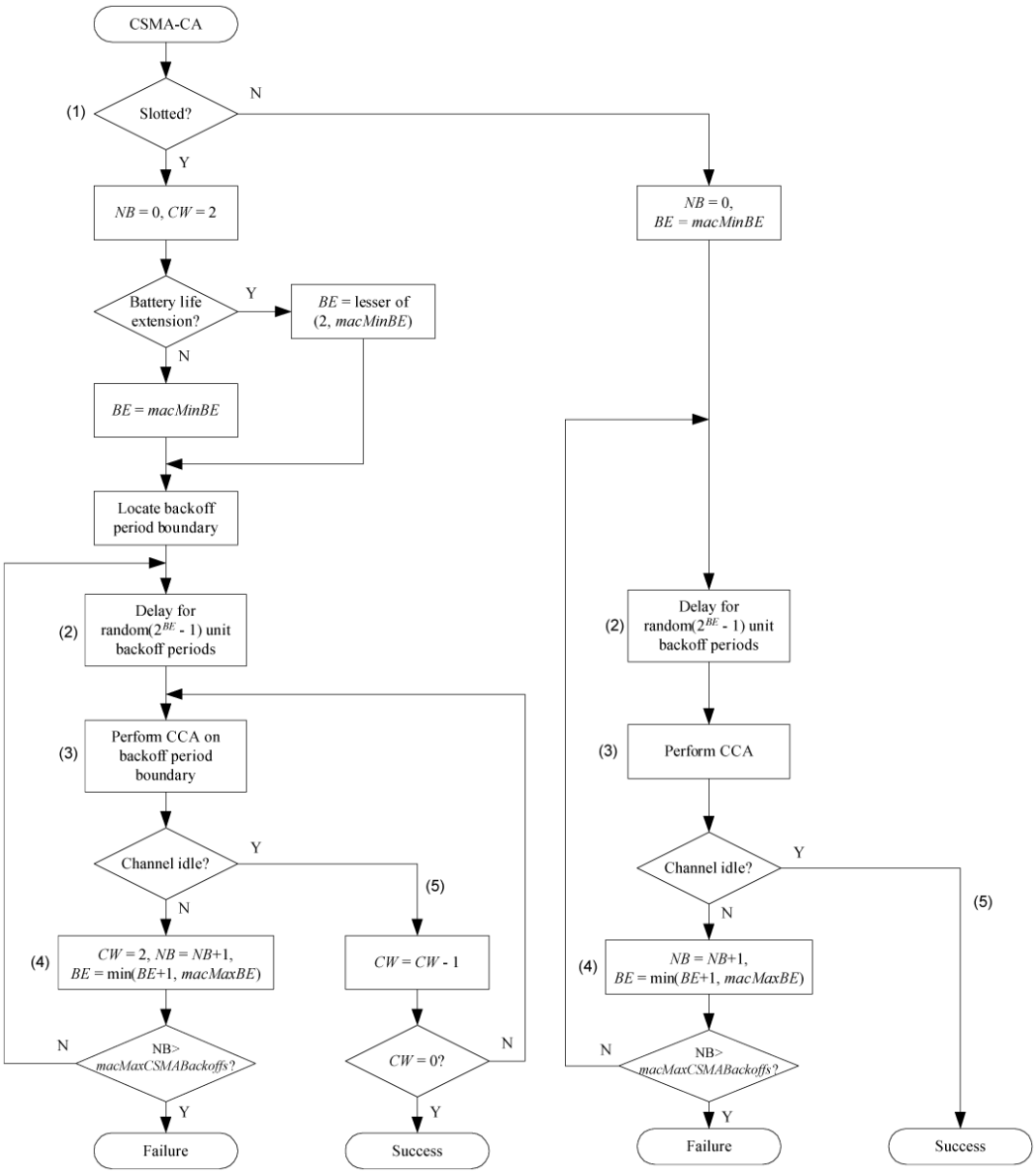

Figure 2.9: CSMA-CA Mechanism

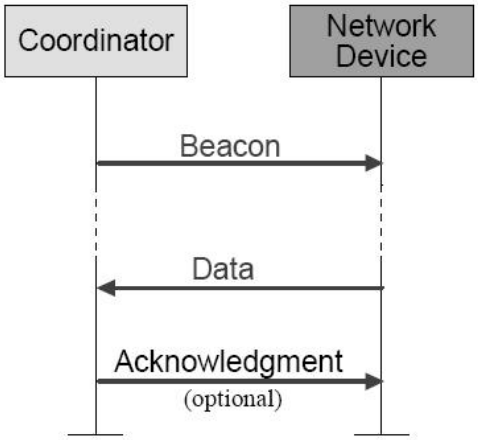

Figure 2.10: Data Transfer to a Coordinator in Beacon-enabled Mode

coordinator acknowledges the successful reception of the data by transmitting an acknowledgment frame. This sequence is shown in Fig. 2.11.

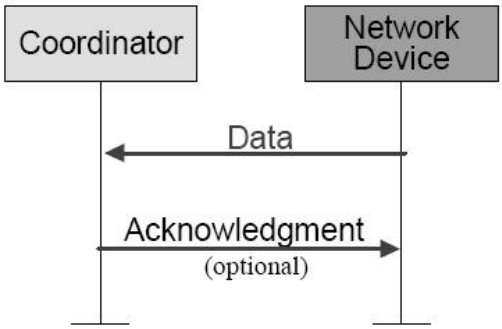

Figure 2.11: Data Transfer to a Coordinator in non Beacon-enabled Mode

Data Transfer from a Coordinator: In a beacon-enabled network the coordinator indicates in the network beacon that the data message is pending for the device. When the device receives this beacon, transmits a MAC command requesting the data, using slotted CSMA-CA. If it is requested, the coordinator acknowledges the successful reception of the data request by transmitting an acknowledgment frame. The pending data frame is then sent using slotted CSMA-CA by the coordinator. The device acknowledges the successful reception of the data by transmitting an acknowledgment frame. After receiving the acknowledgement, the message is removed from the list of pending messages in the beacon frame by the coordinator. This sequence is shown in

Fig. 2.12.

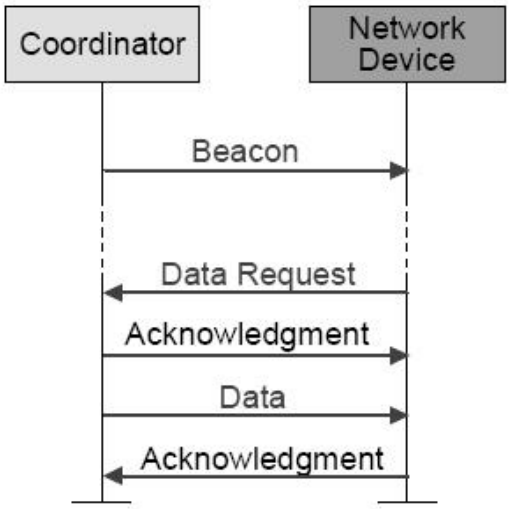

Figure 2.12: Data Transfer from a Coordinator in Beacon-enabled Mode

Whereas in a non beacon-enabled network in order to transfer data to a device, coordinator keeps the data for the device until a data request from the device arrives. Just after receiving the request, coordinator sends an acknowledgement frame to indicate the reception of the data request. Finally, coordinator transmits the data frame, using unslotted CSMA-CA. Once it receives the data the device sends an acknowledgement. This sequence is shown in Fig. 2.13.

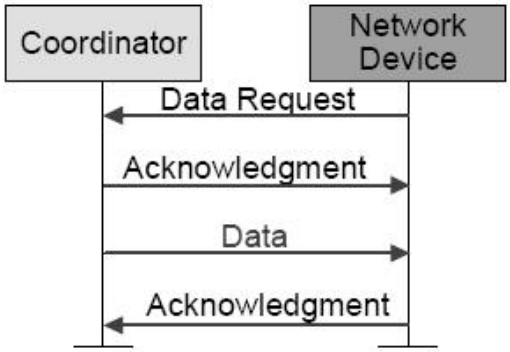

Figure 2.13: Data Transfer from a Coordinator in non Beacon-enabled Mode

Data Transfer between Peer Devices: In a P2P PAN, every FFD can communi-

cate with other FFDs in its radio range. To assure this the device may stay continuously in reception mode scanning the radio channel for on-going communications and when necessary it may simply transmits its data using unslotted CSMA-CA or devices may synchronize to each other in order to be able to go in sleeping mode in an organized way. For the above modes of operation intelligent algorithms are necessary on the upper layers which is beyond the scope of IEEE 802.15.4 Standard.

2.4.4 Frame Structure

In order to achieve robust transmission over a noisy channel, IEEE 802.15.4 Standard defines four frame types; beacon frame, data frame, acknowledgement frame, and MAC command frame. All communication inside the network is realized by using these frames. Frame control, sequence number and addressing fields are common in all frame types except the acknowledgement frame which doesn't have addressing field. The MAC Protocol Data Unit (MPDU) in genearal is shwon in Fig. 2.14.

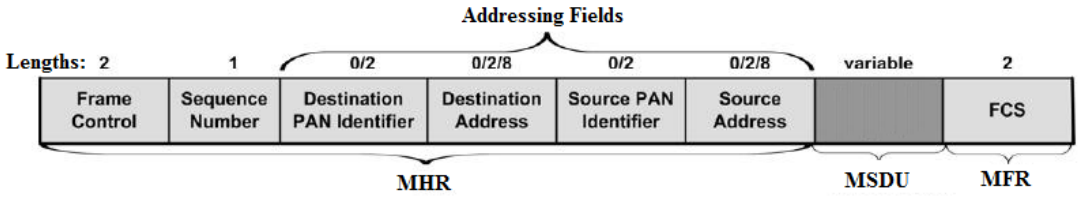

Figure 2.14: MAC Protocol Data Unit in General

The common fields are;

- **Frame Control:** It is a 16-bit long field that contains the information about the frame type and the control flags (i.e. Security Enabled, Frame Pending, Acknowledgment Request, Intra-PAN).
- **Sequence Number:** It is an 8-bit field that contains a unique number for each transmitted frame.
- **Destination PAN Identifier:** It is a 16-bit field that contains the PAN identifier of the recipient of the frame.
- **Destination Address:** It is either a 16-bit or a 64-bit field (depending on the used addressing mode) that contains the address of the recipient of the frame.

- **Source PAN Identifier:** It is a 16-bit field that contains the PAN identifier of the sender of the frame.
- **Source Address:** It is either a 16-bit or a 64-bit field (depending on the used addressing mode) that contains the address of the sender of the frame.
- **MAC Footer (MFR):** It contains the Frame Check Sequence (FCS) field. The FCS is a 16 bit Cyclic Redundancy Check (CRC) sequence.

Beacon Frame: It is the frame that is responsible for the synchronization in the PAN when the beacon-enabled mode is used. During the association of a device each FFD sends a beacon responding to the device with the information about the network. Fig. 2.15 demonstrates the fields in the beacon frame. The lengths of each field are given on the top of the field. Beacon Frame contains four distinct fields; Superframe Specification, GTS, Pending Addresses, and Beacon Payload.

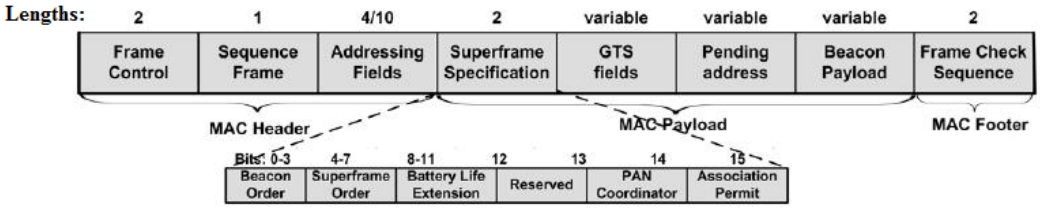

Figure 2.15: Beacon Frame

- **Superframe Specification:** It is a 16-bit field that specifies parameters such as Beacon Order, Superframe Order, Battery Life Extension, PAN coordinator, Association Permit.
- **GTS field:** It is a variable size field that contains information about allocated GTSs.
- **Pending Address:** It is a variable size field that contains addresses of the devices that having pending messages on the coordinator.
- **Beacon Payload:** It is an optional field reserved for upper layer to transmit data in the beacon frame.

Acknowledgement Frame: It is the frame that is used for acknowledgement. There is no addressing field in the frame so the sender follows the acknowledgement frames by checking the sequence numbers.

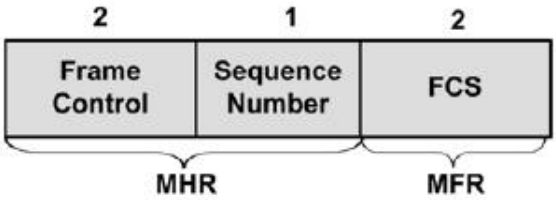

Figure 2.16: Acknowledgement Frame

Data and Command Frames: Data and Command frames are slightly different from the general frame structure shown in Fig. 2.14 because of just having different payloads. In the payload, command frame contains information specific to command types (i.g. Data Request Frame, Purge Request Frame, Associate Request Frame, GTS Request Frame, etc.) while the data frame payload is completely depends on the application running on upper layers.

Chapter 3

Preliminary Measurements

In this chapter the transceiver that is employed throughout the thesis is introduced and some measurements used as a baseline for the rest of the thesis are presented. Firstly in the chapter the relationship between RSS and LQI is investigated. Then RSS values at varying distances are measured. The comparison with a simple path loss model is verified the linearity of these results reported by the transceiver. Especially reliability of the reported RSS value for each packet reception is essential for the measurements in Chapter 4. Moreover in this Chapter PER conditioned to RSS is obtained which then leaded us to the receiver sensitivity in 1% PER. Finally at last section, probabilities of delay in CSMA-CA are given.

3.1 Freescale MC13224 Platform

The MC1322x family [62] is Freescales third-generation ZigBee platform which incorporates a complete, low power, 2.4 GHz radio frequency transceiver, 32-bit ARM7 core based Micro Controller Unit (MCU), hardware acceleration both for IEEE 802.15.4 MAC and Advanced Encryption Standard (AES), and a full set of MCU peripherals into a 99-pin Land Grid Array (LGA) Platform-in-Package (PiP). MC13224 comes with a typical receiver sensitivity better than -96 dBm, and it is equipped with 128 Kbyte serial FLASH memory, 96 Kbyte Static Random Access Memory (SRAM), 80 Kbyte Read Only Memory (ROM). Typical current draw is around 29 mA and 22 mA in TX and RX modes respectively. In the hibernate mode the current draw can be as less

as 0.85μA (In Appendix A there is a comparison of 802.15.4 compatible transceivers). The block diagram of MC1322x family is shown in Fig.3.1.

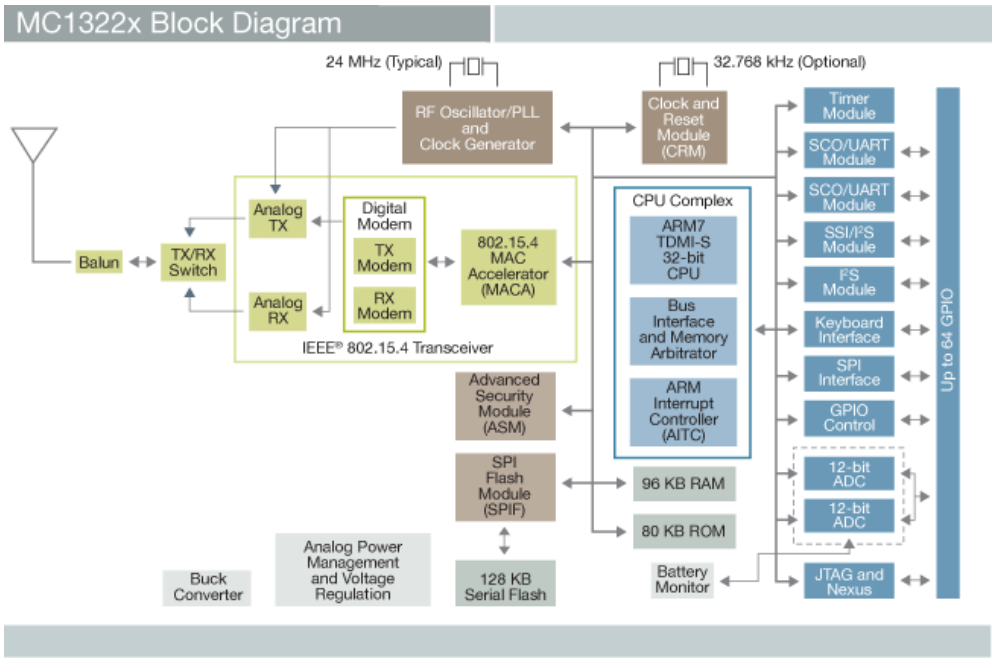

Figure 3.1: MC1322x Family Block Diagram

In the RF receive signal path, as shown in Fig. 3.2 in more details, firstly the RF input is amplified with a Low Noise Amplifier (LNA) then converted to In-phase and Quadrature-phase (I , Q) components through a single down-conversion stage. After these components are amplified, filtered, and digitally sampled to transform the received signal to the digital domain. In the digital domain the modem demodulates the digitized data, a correlator de-spreads O-QPSK signal and determines the symbols and packets.

3.2 Received Signal Strength(RSS) and Energy Detection (ED)

RSS and ED measurements are two fundamental ways for the received signal quality and the assessment of the channel. In this section the capabilities of the transceiver that is used are investigated. In IEEE 802.15.4 Standard [93] LQI is defined as the strength and/or the quality of a received packet. The standard defines three options

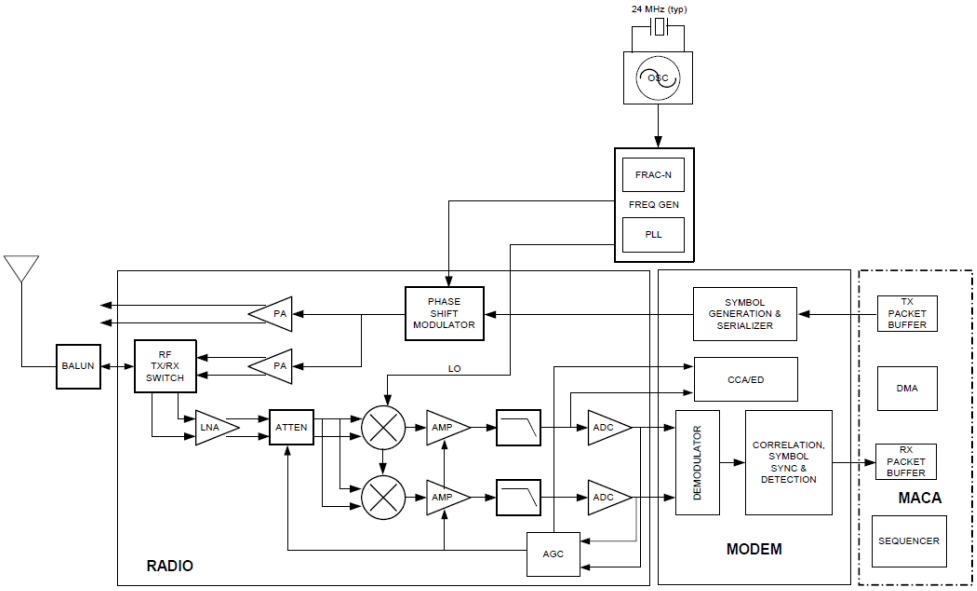

Figure 3.2: MC1322x Transceiver

to measure LQI: RSS estimation, the SNR estimation or a combination of these two methods. In MC1322x family of transceivers for every packet reception receiver block in the transceiver reports a LQI as an 8-bit value from 0 to 255. The value 0 means -100dBm and the value 255 means -15dBm and the values in between are distributed linearly. A simple conversion formula for the RSS given in [63] is:

$$P_{RX}(dBm) = (LQI/3) - 100 \tag{3.1}$$

Regarding the ED in MC1322 family, as required by the 802.15.4 Standard, a value from 0 to 255 is reported. The hardware measured values are scaled and normalized for this range with the minimum value of 0 set to -100dBm and the maximum value of 255 set to -15dBm exactly the same of RSS. Measured values between -15dBm and -100dBm are scaled linearly between 0 and 255 [64]. Therefore the energy level in a channel is:

$$E_Z(dBm) = (ED/3) - 100 \tag{3.2}$$

In the garden of Engineering Faculty of University of Bologna (Fig. 3.3) several sets of measurements are conducted at different distances in order to obtain RSS and

PER. In the measurements Freescale 1322x USB Dongle [60] is used as the receiver while Freescale 1322x-LPN [59] is used as the transmitter. During the measurements transmit power was set to -30 dBm in order keep the maximum radio range of the transceivers at a reasonable distance and in each set of measurement at least 10k packets have been sent. In order to calculate the number of losses, sequence numbers of the packets are controlled. The time interval between two packets was 30ms and the packet size was 20 bytes. Measurement results are given in Table 3.1.

Figure 3.3: Measurement Setup

Table 3.1: Outdoor Measurement Results

Distance[m]	P_{RX}[dBm]	Lost Packets	Received packets	PER
1.5	-85	0	13820	0.00
2.5	-88	1	10185	0.00
3	-91	7	10386	0.00
3.5	-92	14	10117	0.00
4	-93	17	10175	0.00
4.5	-95	44	11728	0.00
4.8	-96	161	9319	0.02
5	-97	5720	15587	0.37
5.5	-98	20293	28308	0.72

In the IEEE 802.15.4 Standard the receiver sensitivity is defined as the received

signal strength that yields a PER less than 1% with 20 bytes packets. In the light of this definition the receiver sensitivity for our device is -95 dBm. In fact, it is the reported typical receiver sensitivity of the USB dongle[60] even though the transceiver itself has better receiver sensitivity, -96dBm, as stated in [62].

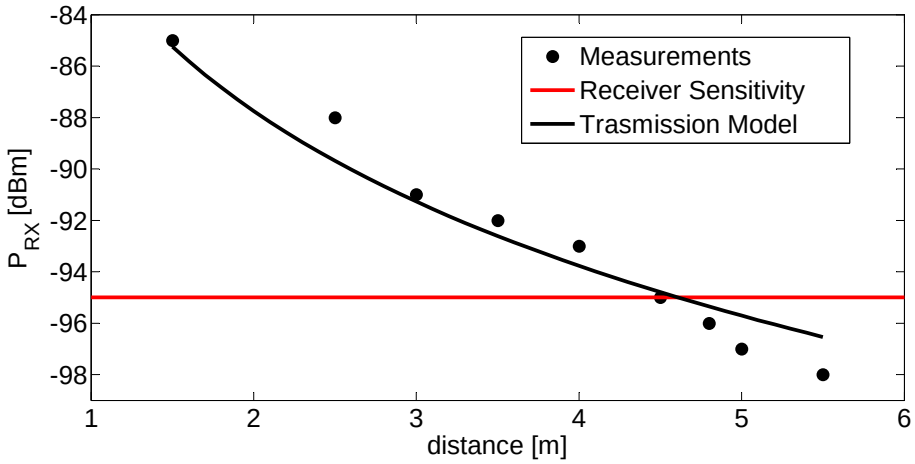

Figure 3.4: P_{RX} - Distance Relationship

Furthermore, in order to reveal the validness of the simple linear map in (3.1) the measured P_{RX} values is tested on the transmission equation given in [81]. The transmission equation can be written in a piecewise form as:

$$P_{RX}(d) = P_{TX}G_{RX}G_{TX}(\frac{\lambda}{4\pi d})^2, d < d_{break} \tag{3.3}$$

$$P_{RX}(d) = P_{RX}(d_{break})(\frac{d}{d_{break}})^{-n}, d > d_{break} \tag{3.4}$$

In the equation, P_{TX} is the transmitted power P_{RX}, is the received power, G_{TX} and G_{RX} are the antenna gain factors, d is the distance between the receiver and the transmitter, λ is the wavelength, d_{break} is the distance where the power is not proportional anymore to d^{-2} and n is an empirical constant typically lying between 3 and 4. In order to obtain a simpler version of the transmission equation, (3.3) can be modified as:

$$P_{RX}(d) = P_{TX}C_l(\frac{\lambda}{4\pi d})^2 \tag{3.5}$$

where C_l represents together the gain components and the losses in the system. In the logarithmic form (3.5) becomes:

$$P_{RX}(d) = P_{TX} + C_l + 10log10(\frac{\lambda}{4\pi d})^2$$

C_l is an unknown constant but by using the measurements in the Table 3.1 a least square estimation to C_l is found as $\hat{C}_l = -11.7dB$. The simple transmission model that is used in (3.5), indeed, fits the measurement points as shown in Fig.3.4 where $P_{TX} = -30dBm$. Consequently it can be stated that the mapping in (3.1) captures the real-life behavior quite good.

In addition, in Fig.3.5 there is the time localized representation of a measurement with the same configuration but when the transmitter is mobile. Asterisks indicate the time of the lost packets on the horizontal axis and P_{RX} of the last received packet on the vertical axis. It is clearly seen that under the receiver sensitivity the graph is populated with the packet loses.

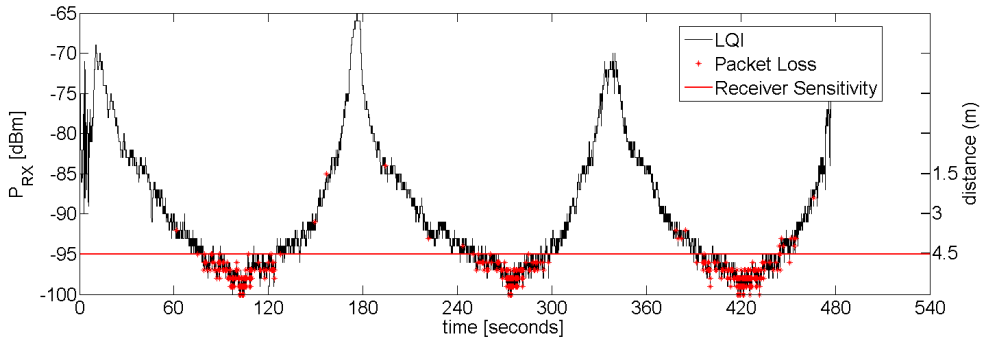

Figure 3.5: P_{RX} Measurements When Varying the Distance

Finally in this section there are examples of indoor measurements done in the first floor of CSITE Building at University of Bologna in Fig. 3.6. In the environment there are different types of obstacles (e.g. windows, concrete walls, plasterboard wall, wooden wall, cupboards) as represented with different colors. In such an environment modeling the path loss and obtaining the transmission equation as done in outdoor measurements is cumbersome but to give some insights only the measurements results are presented. The measurement configuration (30 ms packet interval, 20 bytes packets) is the same like the outdoor measurements except the transmit power is now 0dBm. Each different color in the figure represents a set of measurement. The rectangles are the positions of

the receivers and the circles indicate the positions of the transmitters. In the position of each circle, the transmitter that is located sent at least 10k packets in three different times to the receiver which is indicated with the same color. The measured P_{RX} values in three sets are averaged in order to obtain the average P_{RX} from each transmitter indicated with circles. The diameters of the circles are proportional to the average P_{RX} and there is a legend on the bottom of the figure which shows the dBm values respect to the circle diameters.

3.3 Probability of Delay in slotted CSMA-CA

In this section there are channel access measurements of the slotted CSMA-CA in IEEE 802.15.4 when there is a number of nodes competing with each other in order to access the channel. Detailed information about CSMA-CA can be found in Section 2.4.2. The slotted CSMA-CA algorithm is implemented using units of time called backoff periods, T. In slotted CSMA-CA every node synchronizes itself to the beacons coming from the coordinator. Before transmitting a packet each node waits for a random number of backoff periods. After that, channel sensing is performed; if the channel is found free the node immediately starts the transmission; if, instead, the channel is busy the node enters again the backoff state.

An example of slotted CSMA-CA is given in Fig. 3.7. In the example, firstly, the node initializes the BE to the default value, 3, then randomly picks a number of backoff slot from the interval [0, 2^{BE}] (In the example it is 2 backoff slots) and waits that much before sensing the channel and founding it busy. After discovering the channel busy the node increments the BE value in order attempt again to transmit the packet by picking up another random number of backoff slots but this time from a larger interval [0,2^{BE+1}]. In the example after the second backoff stage (four backoff slots), the node finds the channel idle and transmits its packet.

In a beacon-enabled network with different setups by varying the numbers of nodes, each node is forced to send a packet using slotted CSMA-CA with default parameters in IEEE 802.15.4 Standard just after the reception of the beacon, by this way nodes stat following the steps of CSMA-CA algorithm exactly at the same time. The packets in the network are traced using MC1322 USB dongle. Probabilities of having the beginning of a packet at a backoff slot are given in Fig.3.8 to Fig. 3.10 for different packet

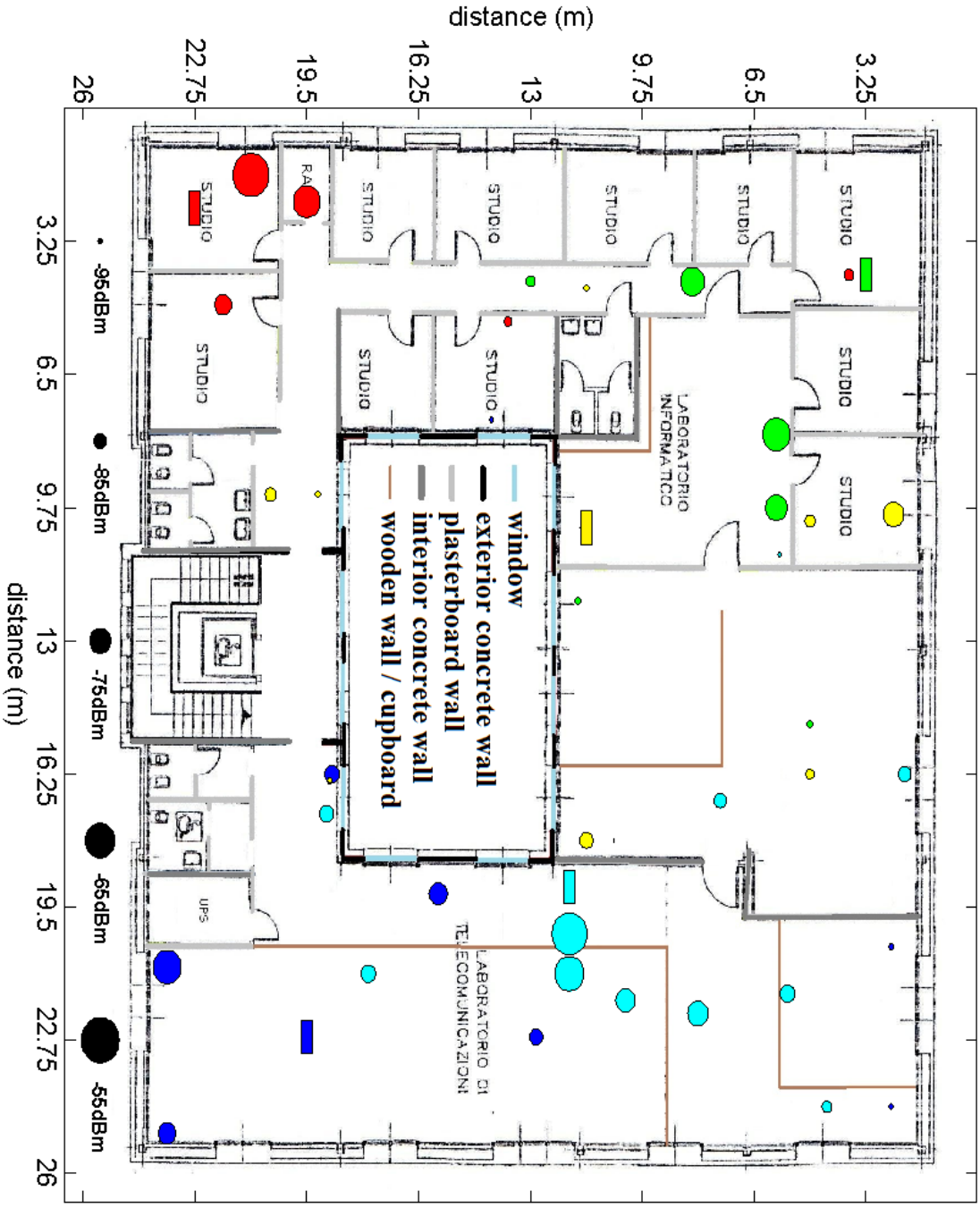

Figure 3.6: P_{RX} Measurements

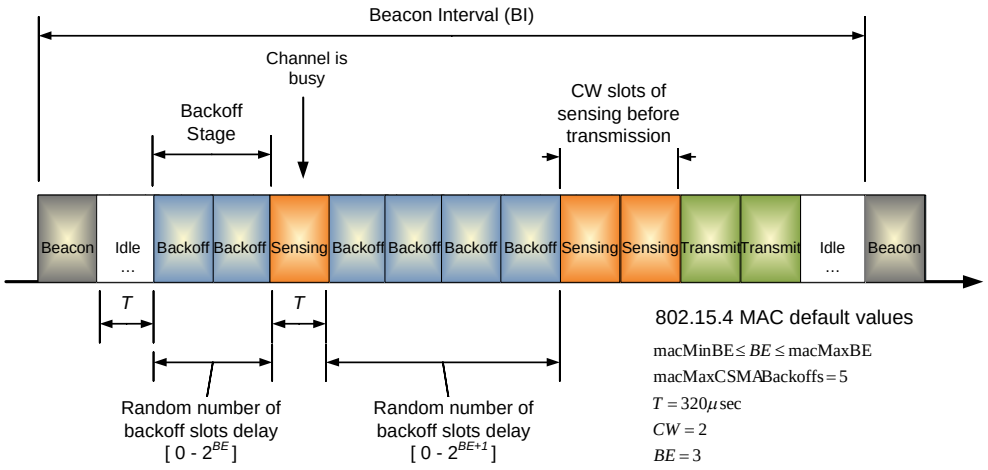

Figure 3.7: Slotted CSMA-CA Example

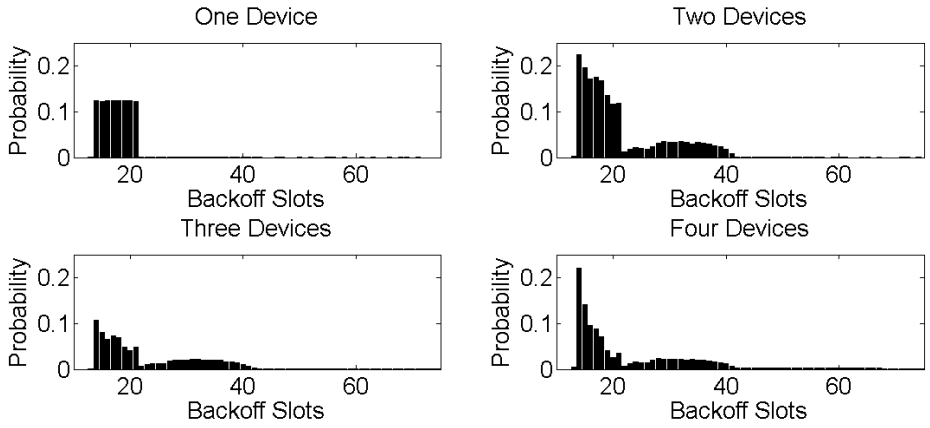

Figure 3.8: Probabilities of Having the Beginning of a Packet in a Backoff Slot with 20-byte Packets

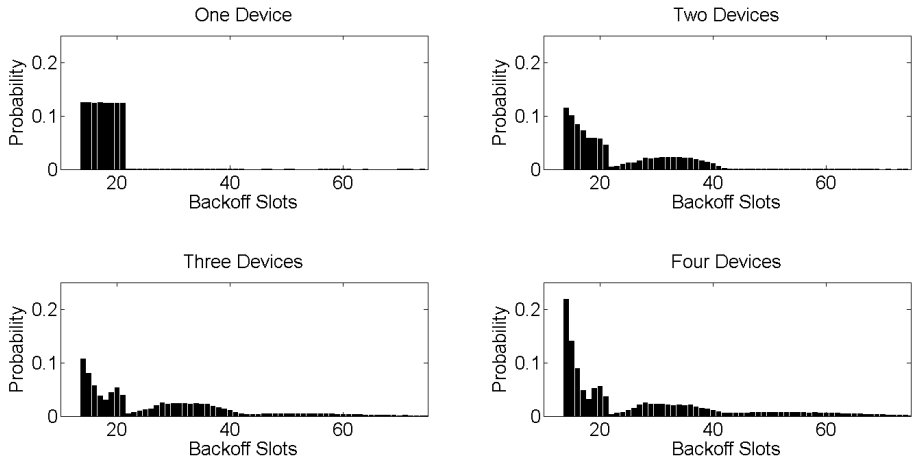

Figure 3.9: Probabilities of Having the Beginning of a Packet in a Backoff Slot with 40-byte Packets

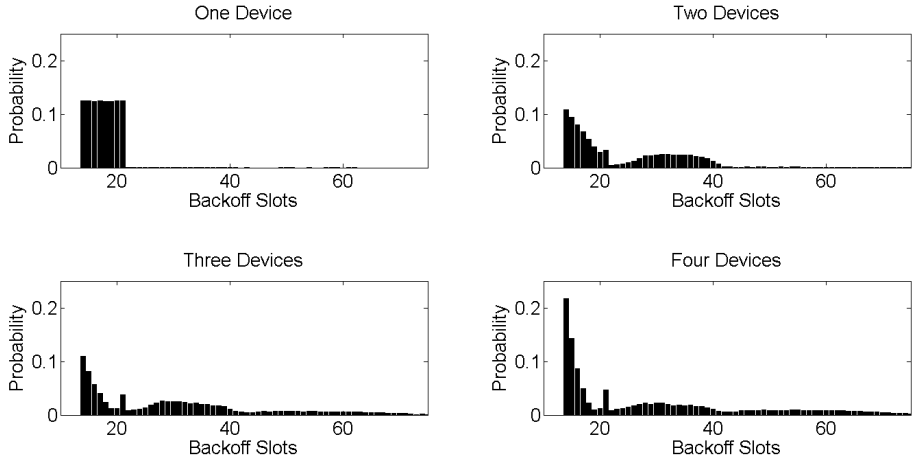

Figure 3.10: Probabilities of Having the Beginning of a Packet in a Backoff Slot with 60-byte Packets

sizes. When there is only one device, the device always finds the channel idle after waiting a random number of backoff slots therefore backoff stage results with a uniform distribution but when the number of devices in the network is increased, the probability of having the beginning of a packet in the farther slots also becomes significant.

Furthermore in the Table 3.2 the probabilities of successfully receiving all the packets from all the nodes in a beacon interval are given.

Table 3.2: Probability of Success

	20-byte	40-byte	60-byte
One Node	≈ 1	≈ 1	≈ 1
Two Nodes	0.91	0.92	0.91
Three Nodes	0.89	0.89	0.91
Four Nodes	0.86	0.88	0.88

Chapter 4

Capture Effect

The notion of "capture effect" refers to the ability of a receiver to correctly receive a packet from a transmitter in the presence of simultaneous transmissions at the same carrier frequency (collisions) [82, 84], even at low values of the SIR. Packet capture may happen inside the receiver; if the useful carrier power, C, is sufficiently larger than the sum of interfering carrier powers, I, the packet is successfully decoded with large probability. We denote this probability as Conditional Packet Capture Probability (CPCP), which is the probability that a packet is successfully decoded conditioned to a given value of the ratio C/I. The behavior of CPCP versus C/I sometimes shows a step-wise behavior, taking value close to one if the SIR is larger than a given value, denoted as protection ratio (i.e., the minimum SIR ensuring the correct reception of a packet), and close to zero otherwise. This happens depending on the type of modulation format, coding scheme and receiver architecture. Under such circumstances, the Packet Success Probability (PSP), which is the probability that a packet is correctly received at the receiver, is just the probability that C/I is larger than the protection ratio. On the other hand, the protection ratio value, in general depends on the number of interferers, N_i, because interference acts as additional noise since for larger N_i the variance of the interference contribution increases, in a given C/I, and therefore CPCP gets smaller.

In this chapter the packet capture in IEEE 802.15.4 is investigated through experimentation and mathematical modeling. A PAN composed of multiple devices (hereafter denoted as *nodes*) working in beacon-enabled mode is considered, and a query-based application: each node upon reception of a periodic query from the coordinator (sent through the beacon packet), attempts to access the channel in order to transmit its

data to a sink, through a direct link. In such scenario, the characteristics of capture effect in IEEE 802.15.4, which uses O-QPSK modulation with DSSS, is difficult to study through mathematical analysis. A test-bed composed of a number of 802.15.4 standard-compliant Freescale devices is used to evaluate the behavior of the CPCP and to measure the protection ratio that allows the correct reception of a packet. These results are used to improve a mathematical model developed for the evaluation of the PSP. In contrast to the lack of considerations of capture effect in literature on 802.15.4 MAC modeling, the model described in [44] is extended also to account for capture effect and to provide more realistic performance in terms of PSP. Results show that capture effect strongly improves performance.

Therefore, in this chapter: (i) we rely on experimental measurements at the link level to show that the step-wise behavior of the CPCP versus C/I curve holds, while no evident dependence of the protection ratio on N_i is found (Section 4.2); then, (ii) we use the determined protection ratio, and evaluate the performance in terms of PSP through an extension to the mathematical model developed in [44] (Section 4.3); finally, (iii) we validate the extension through comparison with simulations, and experimental measurements at the network layer: a very good agreement is found (Section 4.4).

4.1 Related Work

To the best of our knowledge, even though it has been shown that capture effect exists in a wide variety of transceivers including 802.11, Bluetooth radios and cellular systems, there are not many experimental studies in the literature related to capture effect in 802.15.4 transceivers. [96] presents an experimental study using coaxial media, allowing to realize very precise SIR measurements. However, due to the different nature of wireless channel with respect to coaxial medium, these results might give wrong insights on the performance achieved in real-word conditions. In [102] the packet capture probability, when packets are not fully overlapped and received with different signal strengths, is provided. On the other hand our packets are almost perfectly overlapped because of the slotted mechanism in beacon-enabled mode. Finally, [94] presents capture probability values for one interferer, and unstable results for multiple interferers using Chipcon C1000 transceivers[31]. But in our experiments with Freescale MC1322

Plaform [63], we found stable values for packet capture probabilities for one and more than one interferer.

Regarding the model for the 802.15.4 MAC protocol, there exist different works in the literature [49, 80, 89]. However, as stated in [44], none of these models could be applied to query-based applications, where nodes have only one packet per query to be transmitted. All these works, in fact, are based on Bianchi's model and they assume that each packet collides with constant and independent probability, regardless of the backoff stage [40]. This approximation is proper for the above mentioned studies, being the number of nodes competing for the channel constant in time, but it is not suitable to our application, where the number of nodes accessing the channel decreases in time, which results as a decrease in the probability of collision. Moreover some of these models (e.g., [49, 80]) do not show a good agreement with simulation results and none of them compares the model results with experimental measurements. The model presented in [44] and extended here, on the contrary, matches simulations and also experimental measurements, as shown in the following sections. Finally, worth to note that none of the above mentioned models take into account the capture effect.

4.2 Conditional Packet Capture Probability (CPCP) Measurements

A series of measurements have been conducted in order to find CPCP with respect to varying SIR values. In an office environment, IEEE 802.15.4 standard-compliant devices produced by Freescale have been used in the experiments.A PAN composed of a coordinator and a number of nodes is formed at the beginning of each experiment: the coordinator performs energy detection scan in all the available channels and selects the less noisy one. After the PAN formation, nodes start transmissions in each beacon interval. The sniffer receives the 20-byte long packets coming from nodes and sends them to a PC to be processed. The measurement setup for the two interferers is shown in Figure 4.1. 13192-SARD [26] and 13192-EVB [27] boards are used for the coordinator and the nodes of the PAN. These two boards comes with MC13192 [30] transceivers while the sniffer used, instead, is the 1322XUSB board [66], which has a acMCU and a transceiver platform that is explained in Section 3.1, namely MC13224V[62]. 13192 boards are only used with the purpose of generating the traffic but the measurements

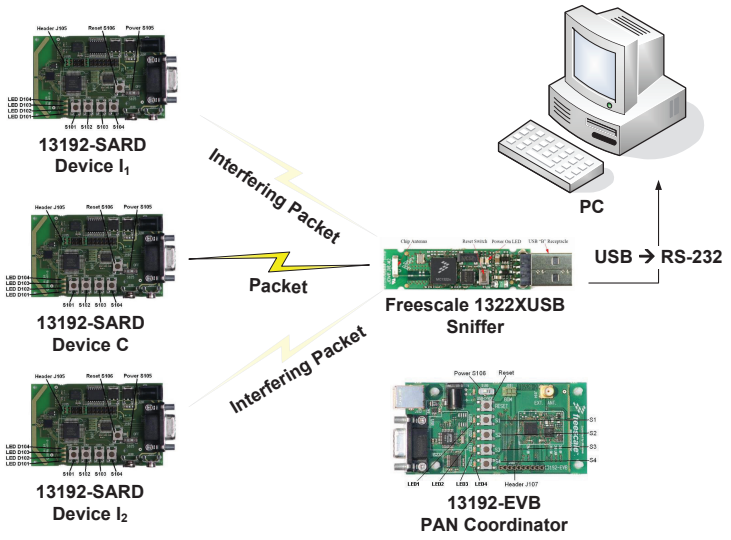

Figure 4.1: Experimental Setup for Two Interferers.

are done on the 1322XUSB board side therefore the measured performance reflects characteristics of MC13224V. All the transceivers used in the experiments operate in 2.4 GHz ISM band and the maximum transmit power of 3.6 dBm is used. The distance between transmitters and sniffer were around 70 cm and they were located over a office table 60 cm above the ground.

In the IEEE 802.15.4 standard [93] LQI is defined as the strength and/or the quality of a received packet. The standard defines three options to measure LQI: RSS estimation, SNR estimation or a combination of these two methods. The Freescale devices used in our test bed estimate the RSS as we have discussed in Section 3.2. The LQI is described as an integer value from 0x00 to 0xFF, which is linearly mapped to the RSS as being 0x00=-100dBm and 0xFF=-15 dBm. Therefore, to derive the RSS value from the LQI, we used the Eq.3.1

In order to measure CPCP a deliberate collision is forced in each superframe. Upon reception of the beacon, each node transmitted a packet with $BE_{min} = 0$, resulting a transmission at the same time. Just after this collision, nodes sent another packet, but this time setting $BE_{min} = BE_{max} = 3$ and $NB_{max} = 0$, to avoid a collision. Since RSS values of collided packets,except the captured, can not be obtained these second transmissions are introduced in order to estimate the C/I at the instance of collision. In Figure 4.2 an example of transmissions performed in each superframe when two

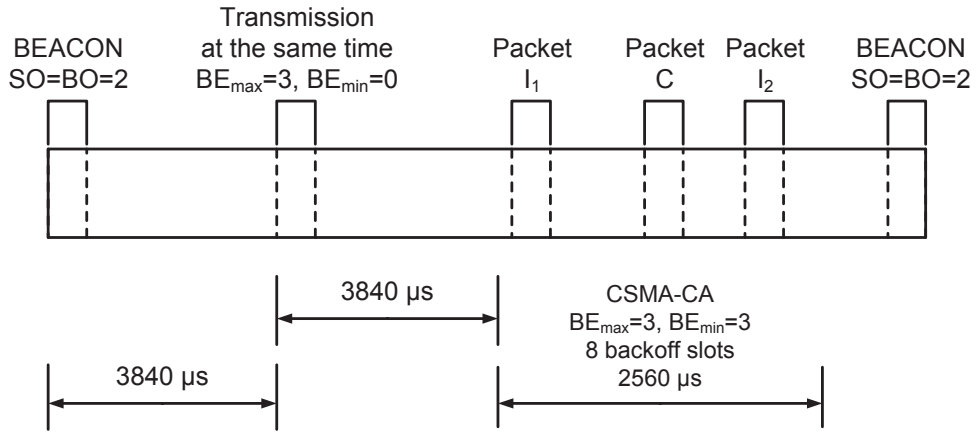

Figure 4.2: Example Timing for Two Interferers.

interferers are present is shown. The collision happens 3840 μs after the beacon then second packets come from the device C and the interferer devices I_1 and I_2 in random backoff slots. In case another collision during the second transmissions happens, this superframe was not considered to evaluate the CPCP. It is worth to note that, having the time gap between the collision and the transmission of the second packets less than 6.4 ms, very reasonable to use the RSS values measured after the collision to compute the SIR, since RSSs cannot significantly change in such a short interval of time.

For each superframe the C/I value was computed (being $C = RSS_C$ and $I = 10\ log10(\sum_{i=1}^{N} 10^{RSS_{I_i}/10}$ where N is the number of interferers) and the information whether the packet coming from device C is captured or not in the collision was stored. Finally, for each particular C/I value the CPCP was evaluated as the ratio between the number of packets captured and the number of superframes which resulted with that C/I. Experiments were conducted until at least 10^4 samples were transmitted for each particular value of C/I. The CPCP as a function of C/I is presented in Figure 4.3 up to four interferers.

No significant dependence between the CPCP curve and the number of interferers is observed. In order to visualize the general behavior we averaged the values for different interferers and applied a piece-wise linear fit. The piece-wise average curve can be seen in the graph as a dashed line. Furthermore, to simply the capture effect model, CPCP curve can be collapsed to a step function which corresponds to the protection ratio of

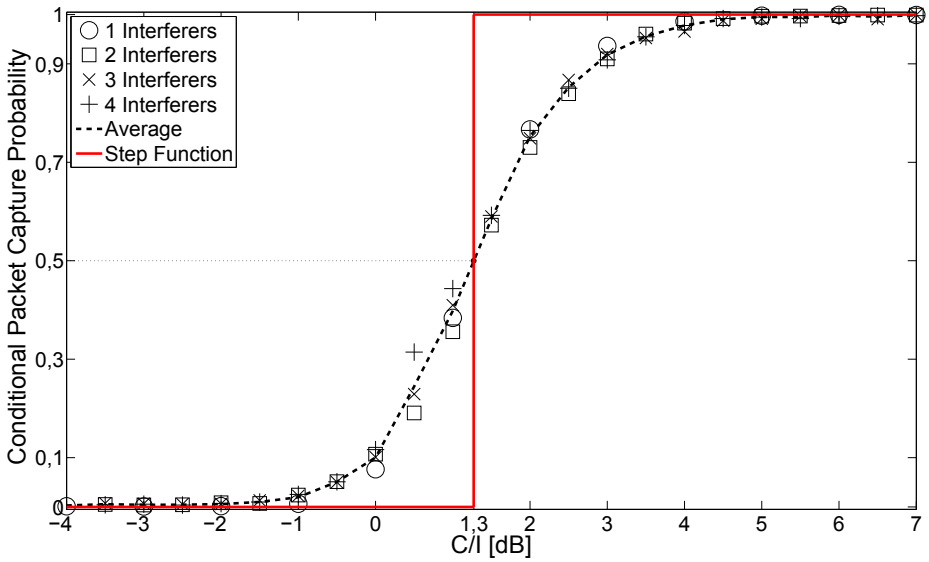

Figure 4.3: Conditional Packet Capture Probability.

1.3 dB at where correct reception of the majority of the packets are guaranteed. In the mathematical model, this value is used.

During the experiments locations of the sniffer and the nodes were random places over a table, quite often we have changed the positions: no significant correlation between the location of the devices and the calculated CPCP, is found. This justifies the use of the RSS values of the packets coming after the collision as an estimate to the C/I: RSS values are stable enough in such a short time interval.

4.3 Mathematical Model

We consider an 802.15.4 network, composed of N nodes transmitting to the sink. We assumed that nodes are uniformly distributed over a circle, having a radius of R, and the sink is located in the center of the circle. Let us also assume that each node can reach the sink (no connectivity losses) and can hear every other transmission occurring in the channel. Nodes transmit packets having size $D{\cdot}10$ bytes, where D is an integer.

In this section the mathematical model for the 802.15.4 MAC protocol in beacon-enabled mode, introduced in [44], is extended to take into account the capture effect. In particular, the model describes the behavior of a node accessing the channel by

using the slotted CSMA-CA algorithm and allows the evaluation of the PSP, which is the probability that a node successfully transmits its packet to the sink by the end of the superframe. According to our application scenario each node has one packet to be transmitted in each beacon interval. In the following sections details about the model developed in [44] are reported and the extension to the model that takes the capture effect into account is provided. Finally, numerical results are shown.

4.3.1 The Model

The PSP, denoted hereafter as p_s, depends on the probability that a node transmits its packet, which is a function of the probability that the node senses the channel and finds it free for two subsequent slots. To derive the above mentioned probabilities, we use the mathematical model in [44] which models the possible generic states in a node and develops a finite-state transmission diagram to evaluate the probability that a node is in a given state (i.e., backoff, sensing, transmission and idle) at a given slot. A slot coincides with the backoff period and all the probabilities are evaluated for all the slots in the superframe (i.e., T_q/d_b slots).

From the diagram in [44] we can derive the probability that a node is in the second sensing phase (CW=2) at the j-th slot and in the i-th backoff stage, denoted as $P\{S2_i^j\}$, where $P\{.\}$ is the probability that a given event happens. It is worth to note that this probability is evaluated for all the slots j and for all the $NB_{max}+1$ backoff stages. For the sake of conciseness, we refer to [44] for the formulas of $P\{S2_i^j\}$.

By denoting $P\{Z^j\}$ as the probability that a successful transmission ends in slot j, we have

$$p_s = \sum_{j=0}^{T_q/d_b-1} P\{Z^j\}\,. \tag{4.1}$$

Before introducing the derivation of $P\{Z^j\}$, we need to underline that in the following we will denote N_c^j as the number of nodes accessing the channel in slot j. It is a random variable, binomially distributed, difficult to model. Hence, to reduce the computational complexity it can be assumed that $N_c^j = N_c = N$ [44, 45].

If the capture effect is not taken into account, the probability that the coordinator correctly receives the tail of a packet in slot j, is the probability that one and only one transmission starts in $j-D+1$. This is due to the fact that $D \cdot 10$ bytes packets occupy D slots (bit-rate is 250 kbit/sec). Only one transmission starts in slot $j-D+1$ if only

one node, over N_c, senses the channel in slot $j-D$ and if the channel is free in $j-D$ and $j-D-1$. The probability $P\{Z^j\}$ is given by:

$$P\{Z^j\}_{nCE} = f^{j-D} \cdot P\{C^{j-D}\} \cdot \prod_{k=0}^{NB_{max}} (1 - P\{S2_k^{j-D}\})^{N_c-1}, \tag{4.2}$$

where the second multiplier gives the probability that one node senses the channel in $j-D$, whatever is the backoff stage, and the third multiplier gives the probability that the remaining N_c-1 nodes do not sense the channel in the slot $j-D$. $P\{C^j\}$ is the probability of being in the second sensing phase (i.e., when $CW=1$) at the j-th slot whatever is the backoff stage, given by $P\{C^j\} = \sum_{k=0}^{NB_{max}} P\{S2_k^j\}$. Finally, f^j is the joint probability of finding the channel idle in slot j and in slot $j-1$, which means the node senses the channel idle for two subsequent slots. This happens if no transmissions start in slots from j to $j-D$. f^j is given by [44]:

$$f^j = 1 - \sum_{v=j-D}^{j} f^{j-1} \cdot \left[1 - \prod_{k=0}^{NB_{max}} (1 - P\{S2_k^{j-2}\})^{N_c-1}\right], \tag{4.3}$$

where $f^{j-1}\left[1 - \prod_{k=0}^{NB_{max}}(1 - P\{S2_k^{j-2}\})^{N_c-1}\right]$ is the probability that at least one transmission starts in slot j.

4.3.2 Extension: Capture Effect

When capture effect is taken into account, a packet may be captured even if a collision occurs, therefore the probability $P\{Z^j\}$ becomes:

$$\begin{aligned} P\{Z^j\}_{CE} = f^{j-D} \cdot P\{C^{j-D}\} \cdot p_{c,N_i} \cdot \binom{N_c-1}{N_i} \\ \cdot P\{C^{j-D}\}^{N_i} \prod_{k=0}^{NB_{max}} (1 - P\{S2_k^{j-D}\})^{N_c-N_i-1}, \end{aligned} \tag{4.4}$$

where p_{c,N_i} is the probability that a packet is captured, when N_i interfering nodes are present. According to the results found in Section 4.2, if C/I is larger than the protection ratio, denoted as α , the useful packet is received.

Therefore, p_{c,N_i} is the probability that the above condition is verified. This probability will be evaluated for $N_i = 1$, whereas an approximated model will be used for $N_i > 1$.

In particular, in case of one interferer ($N_i = 1$), $p_{c,1}$ is:

$$p_{c,1} = P\left\{\frac{C}{I} \geq \alpha\right\} = P\left\{\frac{d_u}{d_i} \leq \alpha^{-\frac{1}{\beta}}\right\}, \tag{4.5}$$

where β is the propagation constant and d_u and d_i are the distances between the coordinator and the useful and interfering nodes, respectively.

Using the relation in [85] (page 186):

$$P\left\{\frac{x}{y} \geq z\right\} = \int_{y=0}^{+\infty} \int_{x=0}^{yz} f_{xy}(x,y)dx\,dy \tag{4.6}$$

in a scenario where the nodes are uniformly distributed over a circle we achieve:

$$p_{c,1} = \frac{z^2}{2} = \frac{\alpha^{-\frac{2}{\beta}}}{2}. \tag{4.7}$$

In general, when there are more than one interferer:

$$p_{c,N_i} = P\left\{\frac{C}{\sum_{i=1}^{N_i} I_i} \geq \alpha\right\}. \tag{4.8}$$

By increasing N_i the complexity in the exact evaluation of Eq. (4.8) increases. To simplify the analysis, we assume that the interferers are all at the same distance d_i, which brings:

$$p_{c,N_i} \cong \frac{(N_i\alpha)^{-\frac{2}{\beta}}}{2}. \tag{4.9}$$

4.3.3 Numerical Results

In this section the extension to the model introduced in Section 4.3.2 is validated through the simulations. For the purpose of numerical comparison, a simulation tool written in C language, has been used. The scenario simulated consists of N nodes distributed over a circle having the radius R, and a sink, located in the center of the circle receiving packets from the nodes. In each superframe, nodes start CSMA-CA algorithm at the same time just after receiving the beacon. We assume that each node can hear all other nodes. In the simulations a collision results with a capture if $C/I \geq \alpha$ and results are obtained over 10^4 rounds.

The comparison is provided in Figure 4.4, where p_s as a function of N is shown. Curves represent mathematical model, whereas points are the simulation results. The results presented in the figure are obtained by setting $SO = 2$, $D = 2$ and $R = 10$ m.

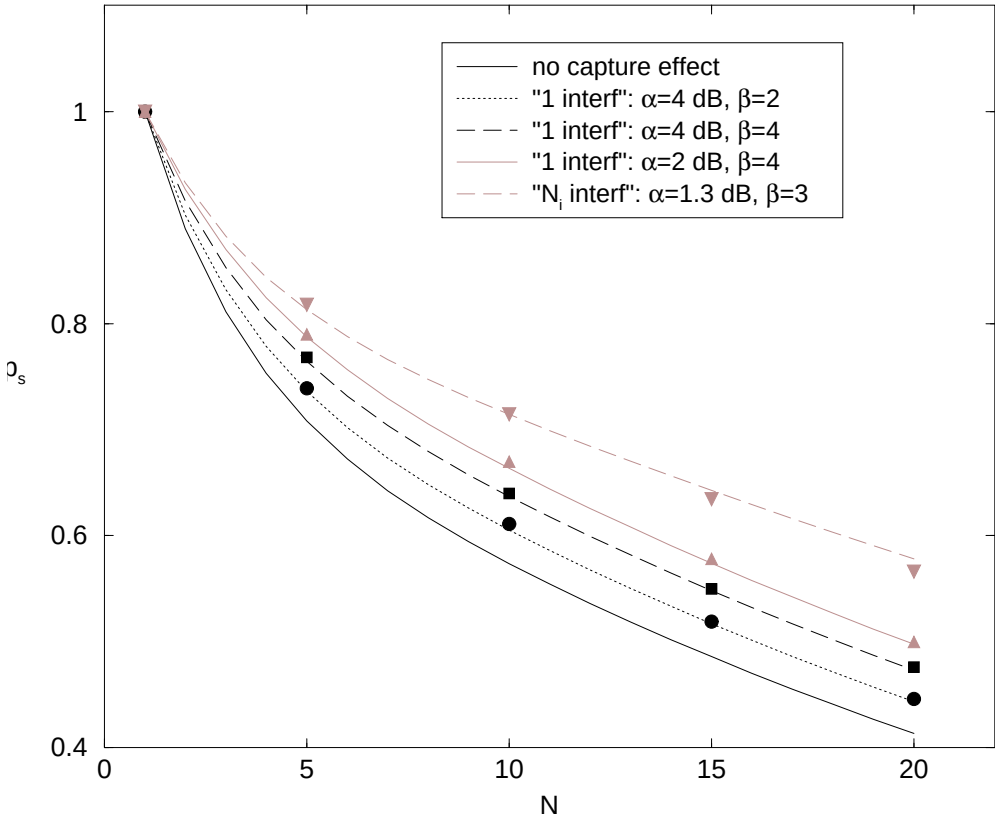

Figure 4.4: p_s as a function of N.

To underline the improvement comes by capturing the useful packet in a collision, the curve achieved with the model discussed in Section 4.3.1 which doesn't consider the capture effect is shown as well.

In particular, we start with a non realistic scenario to show the precision of the model for $N_i = 1$, in both simulation and mathematical results assuming that a packet can only be captured if $N_i \leq 1$ and the condition $C/I \geq \alpha$ is satisfied. We denote this case as "1 interf". Then, for the sake of completeness, we also consider the general case when there is no restriction on the number of interfering nodes. In this case the approximated formula in Eq.(4.9) is used and we set $\alpha = 1.3$ dB, which is the value achieved from the measurements in Section 4.2. We denote this case as "N_i interf". As expected, p_s increases with the impact of capture effect and gets larger by decreasing α and increasing β. The agreement between simulation and mathematical model results, in case of "1 interf", is perfect. In "N_i interf" case when there is the approximation

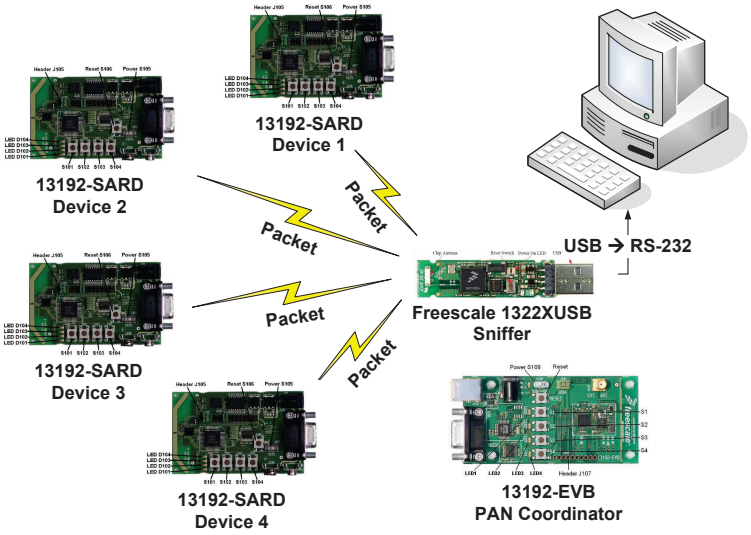

Figure 4.5: Measurement Setup with Four Nodes.

it is quite good with $\beta = 3$. p_s, in the "N_i interf" case, higher than all since a packet may still be captured even if more than one interfering nodes are present.

4.4 Validation

In order to verify the mathematical model results, another measurement is set up. Similar to the previous measurement the MC1322XUSB used to sense the packets in a star topology. The same transmission power and environment described in Section 4.2 is used. In Figure 4.5 the test bed with four devices, is shown. During the experiments, immediately after the reception of the beacon, nodes used CSMA/CA algorithm to send a packet. As in the model assumption, only one packet for each superframe is generated by nodes and in case a node cannot access the channel by the end of the superframe the packet is considered lost. We set $BE_{min} = 3$, $BE_{max} = 5$, $NB_{max} = 4$ and $SO = 2$ and generated at least 10^4 superframes. For up to four devices and packet lengths of 20, 40, 60 bytes, p_s values are computed.

In order to compare the experimental results with the model during the experiments the p_s was evaluated in two different ways: (i) packet capture is considered as a failure ("no CE" case in the Figures); (ii) packet capture is considered as a successful reception ("CE" in the Figures). Therefore the model described in [44] is used in the former case,

whereas the extension model described in Section 4.3.2 is used in the later case.

Regarding the experiments, when packet capture is considered as successful reception, the PSP is simply computed by dividing the number of received packets by the number of sent packets. When, instead, captures considered as failures ("no CE" case), we removed all the collided packets from the number of received packets and divided the result by the number of packets sent. When dealing with such an evaluation some ambiguities, in the determination of collided packets may arise. As an example, in a network of three nodes if only one of the three packets is received in a beacon interval two situations might have happened: (i) one collision between the two packets (receiver does not capture any of the packets) and one reception without collision; (ii) collision between all the three packets, with a capture of one of them. To resolve this kind of ambiguities following assumptions have been done. It has been assumed that packets are lost only in the case of collisions (no physical layer losses). This assumption is very likely since the measured packet receive rate for one device is more than 99.9% in the same environment. Furthermore it is assumed that a collision always ends up with the capture of a packet: this can be achieved by locating the transmitters at different distances to the sniffer to have a C/I value which is always higher than 1.3dBm (protection ratio). Therefore, in the three nodes example above, to exclude capture effect we considered all the packets lost, even though one of the collided packets is received. In the experiments the P_s is computed by averaging over the N nodes. Nodes were in random positions during the experiments. In Figure 4.6 the P_s as a function of N for different values of D, when capture effect is considered as success or failure is shown. There is a good agreement between the mathematical analysis and experimental measurement results. Results validate the model and show that the approximation introduced to simplify the analysis (i.e., the assumption that interfering nodes are at the same distance from the sink) does not significantly affect the performance of the model.

4.5 Conclusion

In this chapter an extension to the mathematical model in [44] which takes into account the capture effect has been introduced. The model is validated by simulations and experiments. To do so we found via experimental activity the CPCP versus C/I

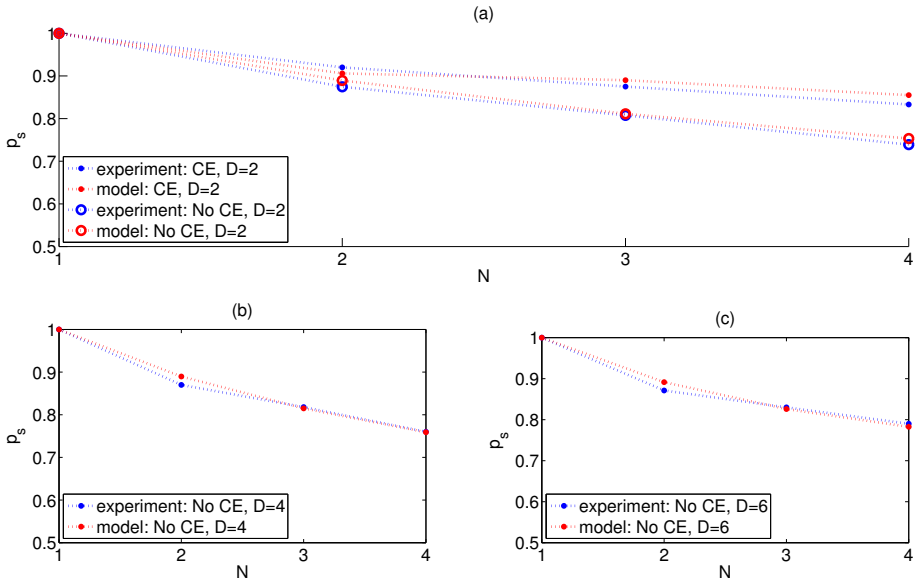

Figure 4.6: (a) p_s values for 20 bytes packets, (b) p_s values for 40 bytes packets, (c) p_s values for 60 bytes packets

curve in order to find the protection ratio, then, we used this value to evaluate the performance in terms of packet success probability; finally, we validated the extension through simulations, and experimental measurements at the network layer.

Chapter 5

Concurrent Transmission

5.1 Introduction

In a shared medium like the wireless channel, even though there are mitigation methods on the MAC Layer, collisions are not completely avoidable. However considering collisions as total failures is not exactly the right assumption. Certain radios can have the ability to correctly receive a signal despite the significant co-channel interference. As we have discussed before in Chapter 4 this co-channel interference tolerance is called capture effect. In the presence of concurrent transmissions at the same carrier frequency (collisions), even at low values of the SIR, packet capture may happen inside the receiver. If the useful carrier power, C, is sufficiently larger than the sum of interfering carrier powers, I, the packet can be successfully decoded. Some of the first papers in the literature that mention capture effect are mostly about the FM demodulators [38, 39, 74]. In addition to the FM demodulators, capture effect property has been shown in a wide variety of transceivers including Aloha networks [82, 84, 90], 802.11 radios [50, 72, 83], Bluetooth radios [54] and cellular systems [42]. On the 802.15.4 side, several experimental activities with different transceivers can be found in the literature. For instance in [102] with the aim of exploiting the capture effect for collision detection and recovery, packet capture probabilities are measured with Chipcon CC1000 transceiver [31]. Another study [94] again with CC1000 obtains capture probability threshold for one interferer but unstable results for multiple interferers and concludes that number of interferers might have an important effect on the capture probability with CC1000. With a more sophisticated transceiver Chipcon CC2420 [97],

a comprehensive study in [76] shows that capture effect is independent to the number of interferers but depends on the total interfering power in contrast to the previous CC1000 measurements. A separated study again from the same authors [77] highlights the independent behavior while demonstrating that the interference is additive. In Section 4.2 with a different transceiver, Freescale MC1224 [62], we have found similar packet capture characteristics as previously discussed in [76, 94]. In fact, two different transceivers in four different study, that are in [48, 76, 94] and Section 4.2, show similar behavior: the receiver starts capturing the useful packet when SIR goes beyond 0dB and around after 4dB Packet Reception Rate (PRR) reaches to 1. On the other hand to the best of our knowledge there is no theoretical model purely based on mathematical analysis of 2.4 GHz PHY of IEEE 802.15.4. In this chapter starting from O-QPSK chip demodulation until the PRR we'll mathematically analyze the receiver path on several different types of theoretical transceivers with the intention of finding explanations to the the real-live behavior of IEEE 802.15.4 2.4 GHz PHY.

In a typical WSN application the data sensed by nodes is usually sent to one (or more) central device, denoted as sink, which collects the information and can either act as a gateway towards other networks (e.g. Internet), where data can be stored in order to be accessed by final users, or to be processed in order to command the actuators to perform specific tasks. When there are many nodes which are periodically sending data to the direction of sinks, concurrent transmissions may occur. In this chapter we investigate the concurrent transmissions in a 802.15.4 network utilized in such scenarios. On the other hand, since IEEE 802.15.4 uses CSMA-CA channel access the probability of having a collision because of more than two concurrent transmissions is relatively low, besides as reported in [76] and Section 4.2, the capture performance of a O-QPSK receiver does not change with the number of interferers. Therefore, the chapter investigates the impact of one interferer on the capture probability, without loss of generality.

The rest of the chapter is organized as follows: Section 5.2 describes CSMA-CA and the 2.4 GHz PHY of IEEE 802.15.4 Standard; Section 5.3 and 5.4 evaluate Chip Error Rate (CER) for coherent and non-coherent O-QPSK respectively; Section 5.5 and Section 5.6 analyze performances of two alternative demodulators which perform in between the coherent and non-coherent demodulators; Section 5.7, obtains PRR in concurrent

transmission by finding an upper bound which transforms CER to Data Symbol Error Rate (DSER).

5.2 System description

In 802.15.4 there can be two different approaches in order to coordinate the data traffic: beacon enabled mode and non beacon-enabled mode [93]. In the former one periodic beacons are sent by the coordinators to synchronize the channel access while in the latter one there is no synchronization. In both approaches CSMA-CA algorithm is used but in slightly different ways. In the CSMA-CA algorithm a unit of time called backoff period is used. In the beacon enabled mode every node synchronizes itself to the backoff slots determined by the coordinator, therefore the channel access is named as slotted CSMA-CA, whereas in non beacon enabled mode every node has its own backoff slot timing, so it is called unslotted CSMA-CA. Default values of parameters and the number of sensing phases before assessing the channel idle is different in slotted and unslotted versions as the result of synchronization. In both versions after the parameter initialization, each node waits for a random number of backoff periods, then channel sensing is performed. If the channel is found free the node immediately starts the transmission, instead, if the channel is busy the node turns back to the backoff state. There is a maximum number of attempt a node can try to sense the channel. When this maximum is reached the algorithm ends with a failure. Even though CSMA-CA avoids collisions by sensing the channel before transmitting a packet still there can be collisions because of hidden terminal effect or in the worst case the channel can be found idle at the same time by two or more nodes. Actually a collision is not a total loss. One of the packets, most probably the one with better signal strength, can be captured as mentioned before.

The IEEE 802.15.4 PHY uses DSSS, and it comes with two ISM band alternatives: one designed to operate in the 868 MHz (EU) and 915 MHz (US) ISM bands, and the other designed to operate in the 2.4 GHz global ISM band. Because of the global availability of the higher frequency band, currently transceiver manufacturers mostly have products working in this band (i.e. Chipcon CC2420, Freescale MC13224, etc.). Following the trend, we are also focusing on the 2.4 GHz frequency band, which utilizes half-sine shaped O-QPSK modulation. In the PHY layer of 2.4 GHz band, the signal is

modulated first by forming a data symbol of four bits and then mapping this symbol to a 32 chip-sequence (see Table 5.8), therefore there are 16 data symbols. The data symbol set modulated at the carrier frequency f_c and carrier phase, ϕ_C can be written as

$$s_{Ci}(t) = I_i(t)\cos(2\pi f_c t + \phi_C) - Q_i(t)\sin(2\pi f_c t + \phi_C), \quad i = 0, 1, ..., 15$$

For each chip sequence even-indexed chips are modulated to the in-phase (I) carrier and odd-indexed chips are modulated to the quadrature-phase (Q) carrier of O-QPSK. The offset between I-phase and Q-phase is formed by delaying the Q-phase symbols by one chip duration with respect to I-phase, therefore $I(t)$ and $Q(t)$ for each data symbol can be written as in [96]

$$I_i(t) = \sum_{n=0}^{15} c_{2n}^i h(t - 2nT_c)$$
$$Q_i(t) = \sum_{n=0}^{15} c_{2n+1}^i h(t - (2n+1)T_c)$$

where $c_{(.)}^i$ is ± 1 respect to the chip value and i is the index of the data symbols in Table 5.8. At the final stage before the multiplication with the carrier each symbol of I and Q phases are half-sine shaped as

$$h(t) = \begin{cases} \cos(\frac{\pi t}{2T_c}), & if\ -T_c \leq t < T_c \\ 0, & otherwise \end{cases} \tag{5.1}$$

In the mathematical representation of half-sine pulse shaping we preferred to use a shifted cosine for the simplicity of the analysis. Finally the complex baseband signal can be written as

$$\tilde{s_{Ci}}(t) = I_i(t) + jQ_i(t)$$

The data symbol set in time domain can also be expressed by using the complex envelope as

$$s_{Ci}(t) = \Re\Big[\tilde{s_{Ci}}(t)\exp\Big(j(2\pi f_c t + \phi_C)\Big)\Big] \tag{5.2}$$

where $\Re[.]$ is the real part of the complex expression.

5.3 Probability of Chip Error in Coherent O-QPSK

In this section, we consider coherent communication and evaluate the chip error probability when there is one interferer with an unknown carrier phase and symbol timing. As the experiments in [48, 76, 94] and Section 4.2 suggest we will assume that the the signal of the interferer is additive and the probability of error is independent to the number of interferers, hence, we'll limit the border of the analysis with one interferer. In this case the received signal in the time domain throughout a O-QPSK symbol similar to (5.2) can be written as

$$r(t) = \Re\Big[\tilde{s}_C(t)\exp\Big(j(2\pi f_c t + \phi_C)\Big)\Big] + \Re\Big[\tilde{s}_I(t)\exp\Big(j(2\pi f_c t + \phi_I)\Big)\Big], \quad -T_c \leq t < T_c \tag{5.3}$$

where $\tilde{s}_C(t)$ and $\tilde{s}_I(t)$ are the complex envelops of the useful and interferer signals respectively, ϕ_C and ϕ_I are the carrier phases of the useful and interferer signals respectively. In fact, there is no phase difference between useful transmitter and the receiver since the demodulation is coherent hence $\phi_C = 0$. But the carrier phase of interferer is unknown. After the simplifications, (5.3) becomes

$$r(t) = \Re\Big[\Big(\tilde{s}_C(t) + \tilde{s}_I(t)\exp(j\phi_I)\Big)\exp(j2\pi f_c t)\Big], \quad -T_c \leq t < T_c$$

which can be expressed in complex baseband form as

$$\tilde{r}(t) = \tilde{s}_C(t) + \tilde{s}_I(t)\exp(j\phi_I), \quad -T_c \leq t < T_c \tag{5.4}$$

$\tilde{r}(t)$ is the function of I and Q components of the useful and interferer signals. In a coherent demodulator I and Q phases should be sampled at the instance when the symbol amplitude is maximum, therefore, because of the offset between I and Q phases in the modulation, in (5.4), I phase should be sampled at $t_I = 0$ and Q phase at $t = Tc$. In general for a sequence of chips I phase should be sampled at each $t_I = 2nT_c$ and Q phase at each $t_Q = (2n+1)T_c$ where $n = 0, 1, 2, \ldots$. This results sampling the I and Q phases of the useful signal in every T_s at the indicated sampling instances in Fig.5.1(a) and Fig.5.4(a) rather than both phases at the same time. Since such a delayed sampling will result with a Q phase delayed projection of the signal on the Q-I constellation plane, we named the resulting constellation plane as Q_d-I to emphasize

the difference. In the following two subsections we will obtain the CER for the received signal in (5.4) with respect to the SIR considering two cases: without pulse shaping and with pulse shaping. Since the intention is to obtain solely relations between SIR and CER, we neglect the noise but in Section 5.7 we'll take into account the noise in the simulations in order to compare the analytical analyzes with the experimental results.

5.3.1 Without Pulse Shaping

We start the coherent analysis without pulse shaping hence, the envelope of the pulse shaping function is always constant at whatever instance the symbol is sampled, as given in (5.5)

$$h(t) = \begin{cases} 1, & -T_c \leq t < T_c \\ 0, & otherwise \end{cases} \tag{5.5}$$

In this case the complex envelope of the received signal in (5.4) on Q_d-I constellation plane will be

$$\tilde{r}(t) = \left(\pm\sqrt{\frac{C}{2}} \pm j\sqrt{\frac{C}{2}} \right) + \left(\pm\sqrt{\frac{I}{2}} \pm j\sqrt{\frac{I}{2}} \right) \exp(j\phi_I), \quad 0 \leq \phi_I < 2\pi \tag{5.6}$$

where C and I are the energies of the useful and interferer signals respectively and ϕ_I is the carrier phase of the interferer. Since the demodulator is coherent, perfect knowledge of carrier phase and symbol timing is assumed for the useful signal while these parameters are supposed to be uniformly distributed random variables for the interfering signal. Symbol timing of the interferer is not required to be analyzed since the pulse shaping function is constant but the carrier phase of the interferer results with a significant impact on the received signal because, ϕ_I rotates the interfering signal on Q_d-I domain as expressed in the second addend of (5.6). The geometric representation of Eq. 5.6 on the Q_d-I constellation plain is shown in Fig. 5.1.

With respect to the transmitted symbol, the first addend in (5.6) can be any of the message points on the Q_d-I domain as shown in Fig. 5.1(b) while the second addend in the equation represents the interfering signal which can be any message points shown in Fig. 5.1(d) but rotated respect to the interferer carrier phase, ϕ_I. The final sum in the equation can be interpreted as in Fig. 5.1(e); respect to the transmitted useful symbol the complex envelope of the received signal will be on the one of the quadrants

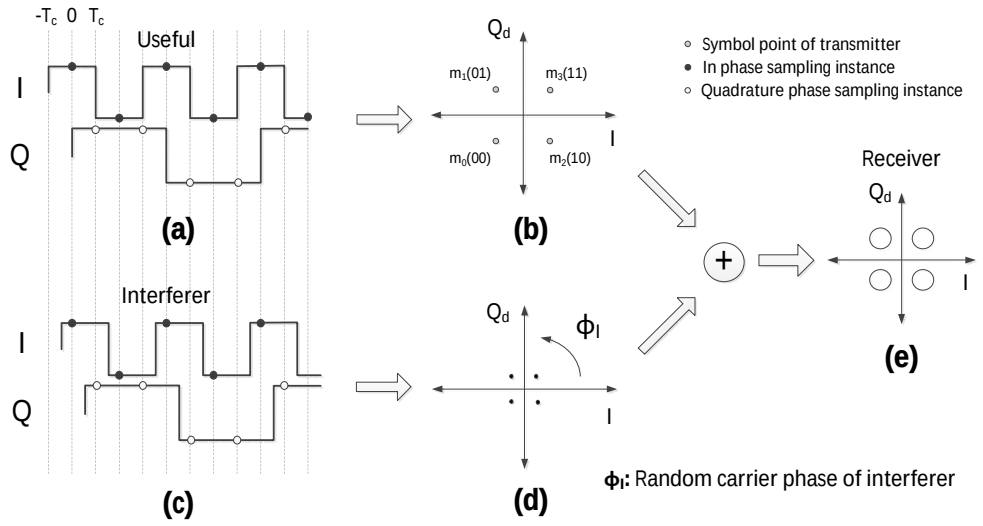

Figure 5.1: Impact of Interferer

in Q_d-I plane but with the impact of the interferer it will be shifted to a random location over the circle which has the radius of $\sqrt{I}$ and the center of $\pm\sqrt{\frac{C}{2}} \pm j\sqrt{\frac{C}{2}}$. Since C is the useful signal energy and I is the interferer signal energy, SIR is defined as C/I. Amplitude of the useful signal on the Q_d-I plane is $\sqrt{C}$ and the amplitude of interferer signal is $\sqrt{I}$. In the following subsections, P_c, and P_s denote the probability of chip error and the probability of O-QPSK symbol error, respectively. After explaining the notation and having (5.6) we can find P_c using a ideal maximum likelihood demodulator [70] on the complex envelope domain. According to the different SIR values there will be three distinct cases that need to be analyzed; **a)** $\sqrt{I} < \sqrt{C/2}$, **b)** $\sqrt{C/2} \leq \sqrt{I} < \sqrt{C}$, **c)** $\sqrt{C} \leq \sqrt{I}$

5.3.1.1 Case: $\sqrt{I} < \sqrt{C/2}$

When $\sqrt{I}$ is smaller than $\sqrt{C/2}$ as shown in Fig. 5.2(a), the demodulator will never make an error since the received signal can never pass to other symbol regions therefore

$$P_c = 0$$

$$P_s = 0$$

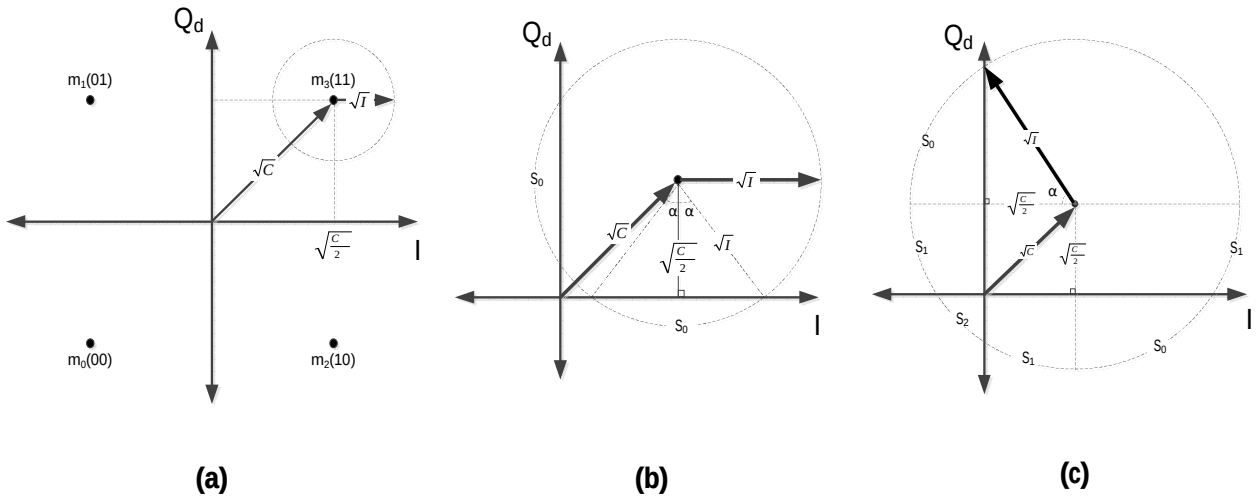

Figure 5.2: Cases: **(a)** when $\sqrt{I} < \sqrt{C/2}$; **(b)** when $\sqrt{C/2} \leq \sqrt{I} < \sqrt{C}$; **(c)** when $\sqrt{C} \leq \sqrt{I}$

5.3.1.2 Case: $\sqrt{C/2} \leq \sqrt{I} < \sqrt{C}$

When $\sqrt{I}$ becomes larger than (or equal to) $\sqrt{C/2}$, the demodulator will start deciding wrongly since interferer energy is sufficient enough to send the usful signal to the adjacent quadrants on the constellation plane, as shown in Fig.5.2(b). In such condition, the probability of symbol error will be given by

$$P_s = \frac{S_0}{\pi} = 2\frac{2\alpha}{2\pi}$$

where S_O is the arc of the circumference shown in Fig.5.2(b), and the angle α in radians is equal to $S_0/2$. α can be written as

$$\alpha = \arctan\sqrt{\frac{I - (C/2)}{\sqrt{C/2}}} \tag{5.7}$$

Since the circle can only pass to the adjacent quadrants P_c will be the half of the P_s

$$P_s = \frac{4\alpha}{2\pi} = \frac{2\alpha}{\pi}$$
$$P_c = \frac{4\alpha}{4\pi} = \frac{\alpha}{\pi}$$

5.3.1.3 Case: $\sqrt{C} \leq \sqrt{I}$

For increasing values of $\sqrt{I}$, the circle as indicated in Fig.5.2(c) will pass also to the other side of the origin. α can still be written as in (5.7), but now

$$S_0 = \alpha$$
$$S_1 = \pi/2 - \alpha$$
$$S_2 = 2\alpha - \pi/2.$$

When the received signal is on S_2 the receiver makes two chip errors over two transmitted chips but when it is on S_0 or S_1 the receiver makes one chip error over two transmitted chips. Probabilities of chip and O-QPSK symbol errors can be written as

$$P_s = \frac{2(S_0 + S_1) + S_2}{2\pi} = \frac{\alpha}{\pi} + \frac{1}{4}$$
$$P_c = \frac{2(S_0 + S_1)}{2\pi}\frac{1}{2} + \frac{S_2}{2\pi} = \frac{\alpha}{\pi}.$$

5.3.2 Half-Sine Pulse Shaping

In this section, we follow an approach similar to that of Sec.5.3.1 besides including the half-sine pulse shaping into the analysis of CER thus, received signal in (5.4) on Q_d-I plane becomes

$$\tilde{r}(t) = \left(\pm\sqrt{\frac{C}{2}} \pm j\sqrt{\frac{C}{2}}\right) + \left(\pm\sqrt{\frac{I}{2}} \pm j\sqrt{\frac{I}{2}}\right) cos(\frac{\pi t}{2T_c} + \phi_I), \quad -T_c \leq t < T_c, \qquad 0 \leq \phi_I < 2\pi \tag{5.8}$$

In (5.8) first addend represents the useful symbol and the second addend represents the interferer symbol. The cosine in the second addend is the pulse shaping function in (5.1). Since we are sampling the Q-phase in delay, there is only one sampling instance on Q_d-I domain both for useful and interferer symbols which is donated as t. The other parameter ϕ_I in (5.8) is the random carrier phase of the interferer which shifts the pulse shaping function on the Q_d-I domain. In order to clarify this random shift suppose that we are observing the interferer, which sends the message $m_3(11)$, throughout a chip duration, T_c, as indicated in Fig. 5.3(a), the trajectory of the signal

in Q-I plane will be the arrow shown in Fig. 5.3(b). Increasing carrier phase will shift the beginning and the end of the arrow to the points indicated by numbers from 1 to 5. White circles represent different ends of the arrow while the blacks are different beginnings of the arrow respect to the different values of carrier phase. The essential point here is how carrier phase shifts the symbol points on Q_d-I plane because it is the plane that we conduct the analysis. In Fig. 5.3(c) there is the Q_d-I plane and the corresponding sampling instances are given with the same numbers and colors. Indeed the symbol point of interferer does a simple harmonic motion which is indicated with gray circles between the points $(\sqrt{\frac{I}{2}}, \sqrt{\frac{I}{2}})$ and $(-\sqrt{\frac{I}{2}}, -\sqrt{\frac{I}{2}})$ on the Q_d-I plane with respect to the carrier phase form 0 to 2π. The amplitude of this motion on 1^{st} and 3^{rd} quadrants of Q_d-I plane can be seen in Fig. 5.3(d).

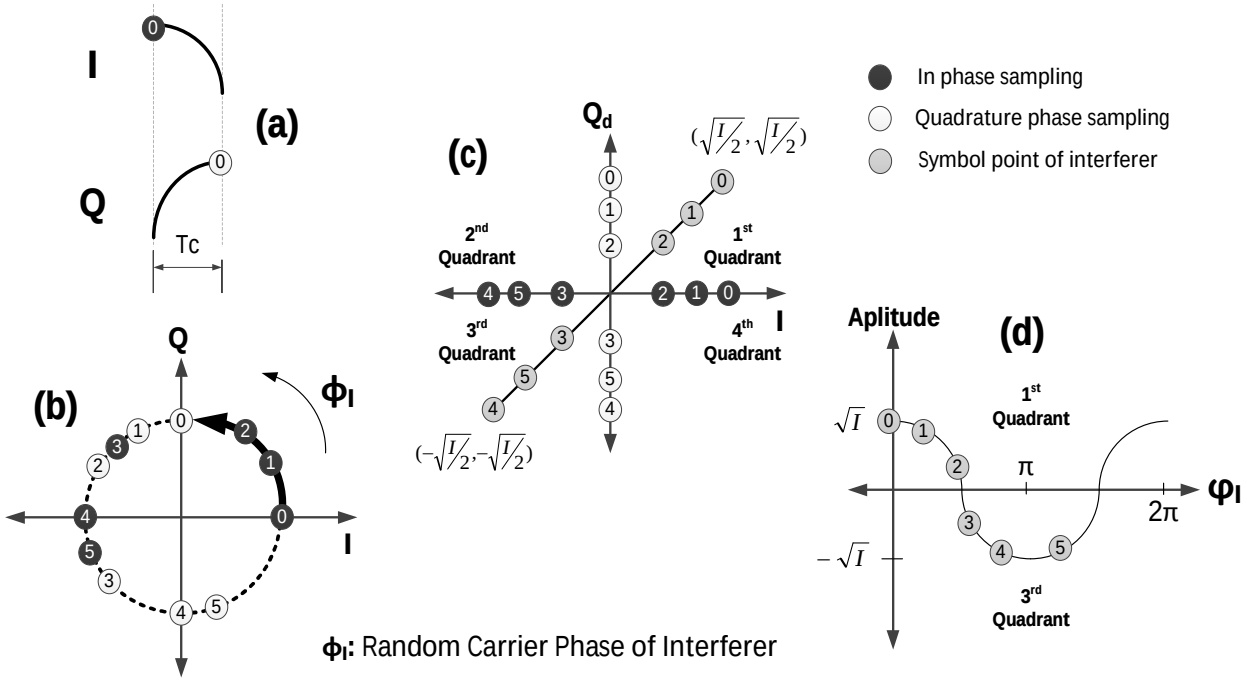

Figure 5.3: Impact of Carrier Phase of Interferer

When there is the perfect knowledge of carrier phase and symbol timing, from the demodulator point of view there is no difference between half-sine shaped or not shaped useful symbols since the sampling is done at the maximum amplitude instance. So the

first addends at (5.6) and (5.8) are the same, which is geometrically represented in Fig.5.1(b) and Fig.5.4(b). But when the pulse shaping function, $\cos(\frac{\pi t}{2T_c} + \phi_I)$, of the interferer is included to the scenario now at the sampling instance the interferer signal can be anywhere on the diagonal lines shown in Fig. 5.4(d). So the received signal disturbed by the interferer will be somewhere on the dashed lines in Fig. 5.4(e). As the next step, in order to find the error probability, we'll obtain the probability distribution of the amplitude of the interferer at the sampling instance.

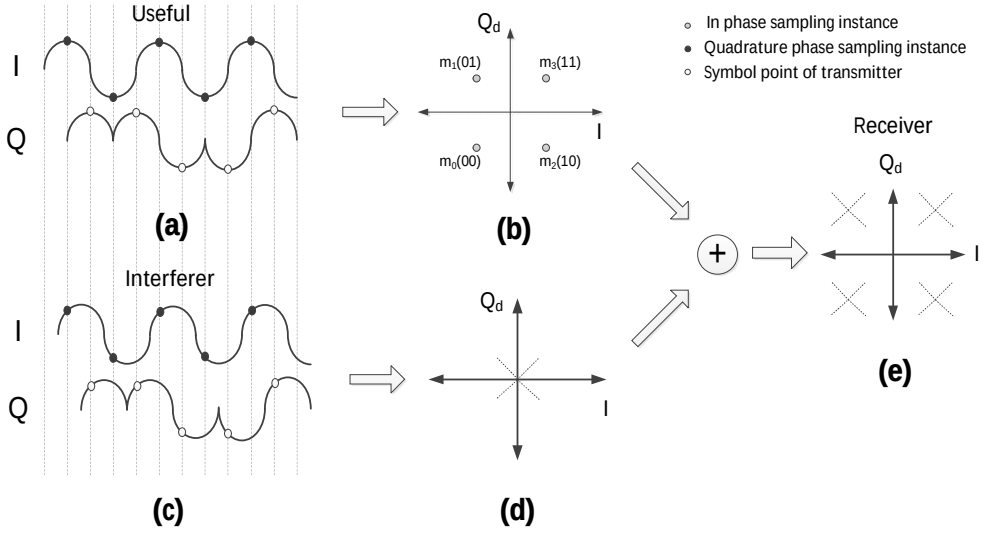

Figure 5.4: Impact of Asynchronous Symbols of Interferer

In order to simplify the notation let's write the uniformly distributed random variables in the pulse shaping function as $X = \frac{\pi t}{2T_c}$, $Y = \phi_I$ and $Z = X + Y$ and denote the Probability Density Functions (PDFs) of X and Y as $f_X(x)$ and $f_Y(y)$; then the PDF of Z will be the convolution integral [85] given as

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z - x) dx$$

where

$$f_X(x) = \begin{cases} \frac{1}{\pi}, & \frac{-\pi}{2} \leq x < \frac{\pi}{2} \\ 0, & otherwise \end{cases} \qquad f_Y(y) = \begin{cases} \frac{1}{2\pi}, & 0 \leq y < 2\pi \\ 0, & otherwise \end{cases}$$

After solving the integral

$$f_Z(z) = \begin{cases} \frac{\pi+2z}{4\pi^2}, & \frac{-\pi}{2} \leq z < \frac{\pi}{2} \\ \frac{1}{2\pi}, & \frac{\pi}{2} \leq z < \frac{3\pi}{2} \\ \frac{5\pi-2z}{4\pi^2}, & \frac{3\pi}{2} \leq z < \frac{5\pi}{2} \\ 0, & otherwise \end{cases}$$

Thanks to the periodicity, in the definition domain of cosine function (i.e. $0 \leq z < 2\pi$), $f_Z(z)$ is actually uniformly distributed as given below

$$f_Z(z) = 1/2\pi, \quad 0 \leq z < 2\pi$$

Therefore, whatever the sampling instance of the interferer and carrier phase of the interferer, Z is uniformly distributed in the interval $0 \leq z < 2\pi$. In each quadrant on the Q_d-I plane the interferer signal will behave the same, hence, the amplitude of the interferer signal in a quadrant on the Q_d-I plane can be written as

$$i = \sqrt{I}\cos(z), \quad -\pi/2 \leq z < \pi/2 \tag{5.9}$$

If $z = 0$, this means the interferer symbol was sampled at the maximum amplitude and the carrier phase was zero. The length of the definition domain is π long since we are observing the amplitude throughout a quadrant. Then the probability of the amplitude can be obtained by using PDF of (5.9) which is:

$$f(i) = \frac{2}{\pi\sqrt{1-(i/\sqrt{I})^2}}, \quad 0 \leq i \leq \sqrt{I} \tag{5.10}$$

where $\sqrt{I}$ is the maximum possible amplitude of the interferer at the sampling instance. Moreover, in order to find the probability of the interferer amplitude lying in an interval defined by the borders a and b, Cumulative Distribution Function (CDF) of (5.9) can be written as:

$$F(a < i \leq b) = \int_{\frac{a}{\sqrt{I}}}^{\frac{b}{\sqrt{I}}} f(i)di = \frac{2}{\pi}(\sin^{-1}(\frac{b}{\sqrt{I}}) - \sin^{-1}(\frac{a}{\sqrt{I}})) \tag{5.11}$$

Without loss of generality as we did in Sec.5.3.1, we can assume that the transmitter is transmitting m_3, then in order to find the P_c and P_s there will be two different cases; $\sqrt{I} < \sqrt{C}$ and $\sqrt{I} \geq \sqrt{C}$.

5.3.2.1 case: $\sqrt{I} < \sqrt{C}$

When $\sqrt{I}$ is smaller than $\sqrt{C}$ the demodulator will never make an error since the received signal will never pass to other symbol quadrants. Therefore P_c and P_s will be equal to zero.

$$P_s = 0$$
$$P_c = 0$$

5.3.2.2 case: $\sqrt{I} \geq \sqrt{C}$

However when i, which is the amplitude of the interferer signal at the sampling instance, is greater than or equal to $\sqrt{C/2}$ the receiver can decide to the wrong symbol with the probability of $\frac{3}{4}$. Regarding the P_c, a transition to the m_2 or m_1 will result with one chip error over two transmitted chips while a transition to the m_0 will result with two chips error over two. Therefore the error rates when $i > \sqrt{C/2}$:

$$P_s = \frac{3}{4}$$
$$P_c = \frac{1}{4}\frac{1}{2} + \frac{1}{4}\frac{1}{2} + \frac{1}{4} = \frac{1}{2}$$

In more general we can write the P_s and P_c as the functions of i like the following:

$$P_s(i) = \begin{cases} 0, & i < \sqrt{C} \\ \frac{3}{4}, & \sqrt{C} \leq i \leq \sqrt{I} \end{cases}$$

$$P_c(i) = \begin{cases} 0, & i < \sqrt{C} \\ \frac{1}{2} & \sqrt{C} \leq i \leq \sqrt{I} \end{cases}$$

Finally we can derive error probabilities when $\sqrt{I}$ is greater than $\sqrt{C}$ by evaluating the expected values of $P_s(i)$ and $P_c(i)$. In the equations below $f(.)$ is the PDF given in (5.10) and $F(.)$ is the CDF given in (5.11).

$$P_s = E[P_s(i)] = \int_{\sqrt{C}}^{\sqrt{I}} \frac{3}{4} f(i)di = \frac{3}{4}.F(\sqrt{C} < i \leq \sqrt{I}) = \frac{3}{4}\frac{2}{\pi}(\frac{\pi}{2} - \sin^{-1}(\sqrt{\frac{C}{I}})) = \frac{3}{4} - \frac{6}{4\pi}\sin^{-1}(\sqrt{\frac{C}{I}})$$

$$P_c = E[P_c(i)] = \int_{\sqrt{C}}^{\sqrt{I}} \frac{1}{2} f(i)di = \frac{1}{2}.F(\sqrt{C} < i \leq \sqrt{I}) = \frac{1}{2}\frac{2}{\pi}(\frac{\pi}{2} - \sin^{-1}(\sqrt{\frac{C}{I}})) = \frac{1}{2} - \frac{1}{\pi}\sin^{-1}(\sqrt{\frac{C}{I}})$$

5.3.3 Validation of the Analytical Model through Simulations

After having the analytical expressions for P_c now we will check validness of the expressions through simulations. Like the other simulations in the rest of the chapter, also here in this part we used MATLAB. A modular simulator which is able to simulate different types of demodulators and shape of pulses is developed. Simulations are done with the complex envelopes of the signals. Asynchronous symbols are obtained by shifting the complex envelope in time axis while random carrier phase is obtained through rotating the complex envelope on the complex plane. SIR values with a step of $0.3dB$ from $-10dB$ to $5dB$ are simulated. At each specific SIR value, 10000 random O-QPSK symbol points both for interferer and useful transmitter are generated. The complex baseband forms of O-QPSK symbols are represented with 100 sample points, therefore the resolution of the time for the asynchronous O-QPSK symbols was $1\mu s/100 = 10ns$, on the other hand for the carrier phase the resolution was $2\pi/100$. For each interferer symbol, random asynchronous symbols with random carrier phase are generated with the mentioned resolutions. All random parameters that are mentioned are uniformly distributed. In Fig.5.5 results are given with the perfect agreement between simulation points and analytical curves. Asymptotically there is a 3dB difference between the results of half-sine shaped and non-shaped symbols. In half-sine pulse shaping P_c goes to zero at $0dB$ while without pulse shaping at $3dB$.

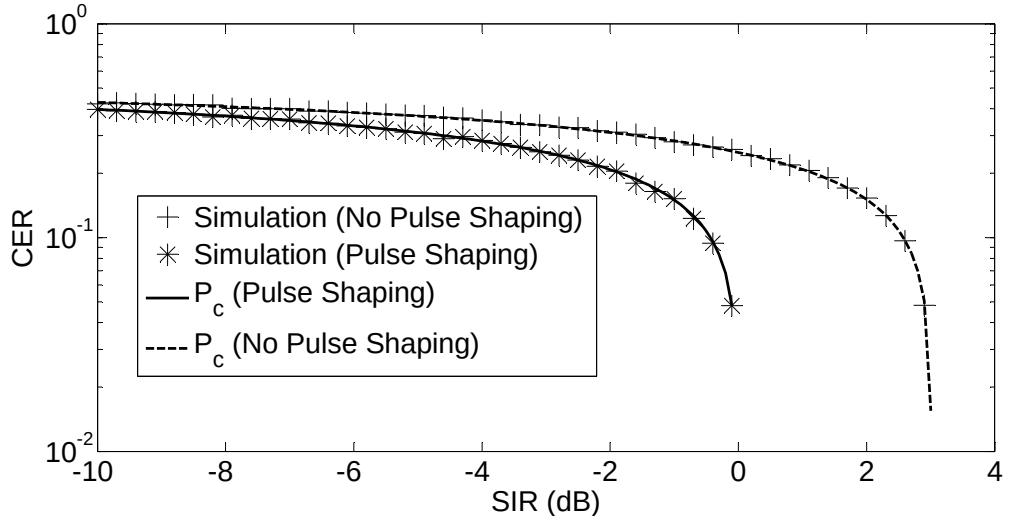

Figure 5.5: Coherent O-QPSK Demodulator Chip Error Rate

5.4 Probability of Chip Error in Non-Coherent O-QPSK

In the previous section we used Q_d-I plane in order to simplify the calculations but from now on we will turn back to the Q-I plane since in this section we will observe the phase change of the complex-envelope throughout a chip duration rather than sampling the transmitted symbol at the maximum amplitude. Normally it is not possible to demodulate O-QPSK non-coherently [70], however thanks to the half-sine pulse shaping in IEEE 802.15.4, information-bearing part of the signal is not only the carrier phase but also the complex-envelope of the signal. In each chip duration, T_c, the phase of the complex envelope increases or decreases at an amount of $\pi/2$. In this sense it is similar to Minimum Shift Keying (MSK) signal thus phase transitions can be detected in order to demodulate the signal. Instead, the difference is, in MSK $+\pi/2$ phase change means chip 1 is transmitted and $-\pi/2$ means chip 0 is transmitted but symbol mapping in IEEE 802.15.4 is different as shown in Fig.5.6(a). It can be though that, in every even chip duration, T_c, a symbol is transmitted and in the odd chip durations there are phase transitions in order to turn back to the 0 and π radians on the complex plane. Because of the random carrier phase, ϕ_C, the Q-I plane rotates, for this reason, as mentioned before, it is not possible to detect which O-QPSK symbol is transmitted. Only phase changes of the complex envelope can be detectable. For instance in Fig.5.6(b) phase change of complex envelope respect to the ϕ_C and Q/I components of transmitted signal are shown.

In this section we will evaluate the probability of erroneous detection of the phase transition during a chip interval and name this probability as P_c. In fact, with this notation, P_c also becomes the chip error rate for MSK in the non-coherent demodulation. We can represent the complex envelope of the received chip through a chip duration in Q-I plane as:

$$R(\omega t) = \sqrt{C}e^{j(\pm\omega t+\phi_C)}, \quad \omega t \epsilon [0, \pi/2], \quad \phi_c \epsilon [0, 2\pi]$$

where C is the energy of the signal and ϕ_C is the uniformly distributed carrier phase. Definition domain of ωt is from 0 to $\pi/2$ because throughout a chip interval signal can only go one quadrant far on the Q-I plane. The instantaneous phase of the $R(\omega t)$ will

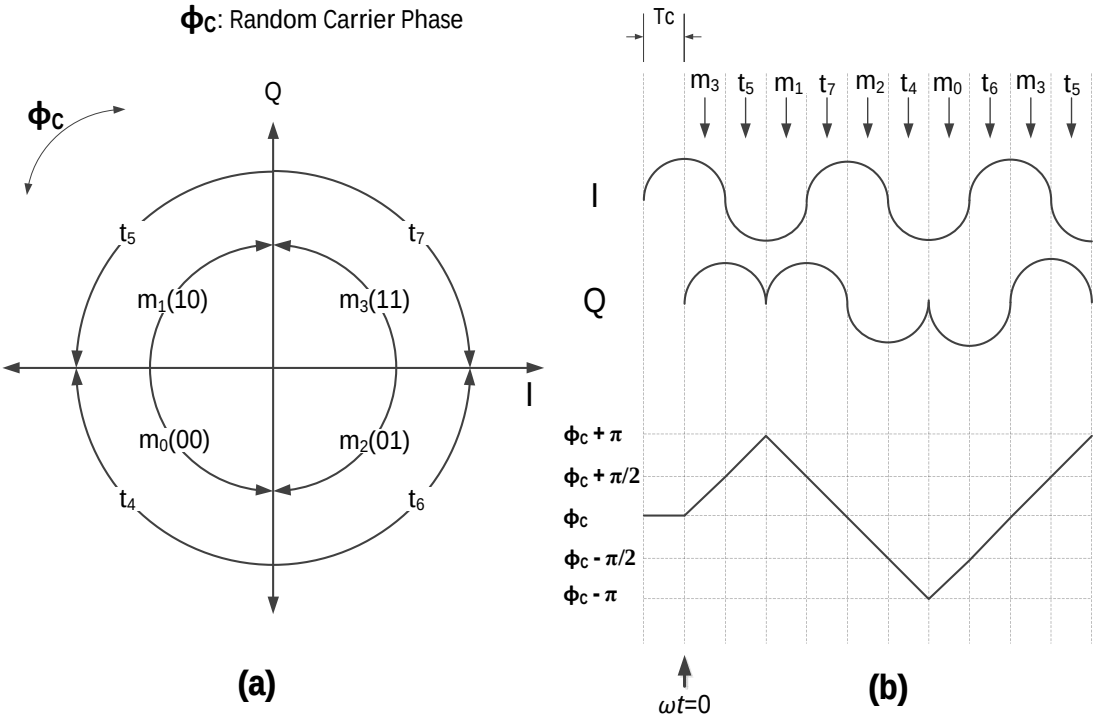

Figure 5.6: Phase Transitions of Complex Envelope

be

$$\Theta_R(\omega t) = \pm\omega t + \phi_C \tag{5.12}$$

therefore, the phase of the received signal at the beginning and at the end of the chip interval can be written as:

$$\Theta_R(0) = \phi_c$$
$$\Theta_R(\pi/2) = \pm\pi/2 + \phi_c$$

Since the carrier phase will be eliminated by subtracting $\Theta_R(\pi/2)$ and $\Theta_R(0)$, a discriminator demodulator similar to [78] can be defined as:

$$\Theta_R(\pi/2) \underset{-j}{\overset{+j}{\gtrless}} \Theta_R(0) \tag{5.13}$$

In (5.13) the decision is based on the comparison of the phases of the complex envelope of received signal at the beginning and at the end of the chip interval. The

demodulator decides on the positive phase change, $+j$, if the phase at the end of the chip interval is higher than the beginning, when it is vice versa it decides on negative phase change, $-j$. Firstly, we will analyze the case where the symbols of the interferer are synchronous to the useful symbols, which is not realistic but helps the mathematical analysis, then the analysis will be extended to the realistic case where the symbols of the interferer are asynchronous.

5.4.0.1 Synchronous Symbols of Interferer

Another transmission with the energy of I and the carrier phase of ϕ_I can act as interferer; the received signal in this case will be:

$$R(\omega t) = \sqrt{C}e^{j(b_C\omega t+\phi_C)} + \sqrt{I}e^{j(b_I\omega t+\phi_I)} \tag{5.14}$$

where $\omega t = [0, \pi/2]$, $\phi_C = [0, 2\pi]$ and $\phi_I = [0, 2\pi]$. In (5.14) b_C and b_I represent turning directions of the useful and the interferer chips on the complex envelope domain (i.e. $b_C, b_I = +1$ means a clockwise turn whereas $b_C, b_I = -1$ means counter-clockwise turn). We name the probability of detection error as P_{eq} when the useful and interferer phase transitions are the same whereas when they are different we name the probability of detection error as P_{dif}. Half of the all possible events are P_{eq} and the other half is P_{dif} so the chip error probability is:

$$P_c = \frac{1}{2}P_{eq} + \frac{1}{2}P_{dif} \tag{5.15}$$

P_{eq} will be always zero since in the Q-I plane received signal always turns at the direction of useful signal if the interferer signal is also turning at the same direction. In order to obtain P_{dif} we can analyze the case where $b_C = +1$ and $b_I = -1$. With this choice the useful chip will have the phase transition $+j$ thus, we can define a success/error rule for the demodulator based on the phase values at the beginning and end of the chip interval as

$$\Theta_R(\pi/2) \underset{e}{\overset{s}{\gtrless}} \Theta_R(0) \tag{5.16}$$

where s means success and e means error. If $\Theta_R(\pi/2)$ is higher than $\Theta_R(0)$, the demodulator will successfully receive the useful signal otherwise it will make an error. The received signal will be

$$R(\omega t) = \sqrt{C}e^{j(\omega t+\phi_C)} + \sqrt{I}e^{j(-\omega t+\phi_I)} \tag{5.17}$$

with the instantaneous phase function of $\Theta_R(\omega t)$. However now obtaining an analytical expression to $\Theta_R(\omega t)$ is not straightforward like in (5.12) because of the summation in (5.17). Let's assume that during a chip interval $R(\omega t)$ always stays on the definition domain of arctan (i.e. $[-\pi/1\ \ \pi/2]$) then the instantaneous phase function can be written as

$$\Theta_R(\omega t) = \arctan(\frac{\sqrt{C}\sin(\omega t+\phi_C) + \sqrt{I}\sin(-\omega t+\phi_I)}{\sqrt{C}\cos(\omega t+\phi_C) + \sqrt{I}\cos(-\omega t+\phi_I)}) \tag{5.18}$$

In order to obtain the P_c we can assume that $\phi_c = 0$ with no loss of generality. Now after the trigonometric simplifications the phases of the received signal at the beginning and at the end of the chip interval will be

$$\Theta_R(0) = \arctan(\frac{\sqrt{I}sin(\phi_I)}{\sqrt{C}+\sqrt{I}\cos(\phi_I)})$$
$$\Theta_R(\pi/2) = \arctan(\frac{\sqrt{C}-\sqrt{(I)}\cos(\phi_I)}{\sqrt{I}\sin(\phi_I)}))$$

After replacing $\Theta_R(0)$ and $\Theta_R(\pi/2)$ in (5.16) test becomes

$$\arctan(\frac{\sqrt{C}-\sqrt{I}\cos(\phi_I)}{\sqrt{I}\sin(\phi_I)}) \underset{e}{\overset{s}{\gtrless}} \arctan(\frac{\sqrt{I}\sin(\phi_I)}{\sqrt{C}+\sqrt{I}\cos(\phi_I)})$$

We can also remove the arctan since it is monotonically increasing at $[-\pi/1\ \ \pi/2]$.

$$\frac{\sqrt{C}-\sqrt{I}\cos(\phi_I)}{\sqrt{I}\sin(\phi_I)} \underset{e}{\overset{s}{\gtrless}} \frac{\sqrt{I}\sin(\phi_I)}{\sqrt{C}+\sqrt{I}\cos(\phi_I)} \tag{5.19}$$

in order to keep the trajectory of $R(\omega t)$ on the definition domain of arctan we can restrict the definition domain of ϕ_I as $[0, \pi/2]$ then $\sqrt{C}+\sqrt{I}\cos(\phi_I)$ and $\sqrt{I}\sin(\phi_I)$ will always be greater than zero. After trigonometric identity substitutions and simplifications (5.19) becomes

$$C - I\cos^2(\phi_I) \underset{e}{\overset{s}{\gtrless}} I\sin^2(\phi_I)$$

which is equal to

$$C \underset{e}{\overset{s}{\gtrless}} I \tag{5.20}$$

It can be showed that for the other domain definitions of $R(\omega t)$, (5.20) holds, therefore P_{dif} is

$$P_{dif} = 1, \qquad C < I$$
$$P_{dif} = 0, \qquad C > I$$

After finding out P_{dif} now we can evaluate chip error probability in (5.15) as :

$$P_c = 1/2, \qquad C < I$$
$$P_c = 0 \qquad C > I$$

5.4.0.2 Asynchronous Symbols of the Interferer

Asynchronous symbols of the interferer will increase the possible number of cases, for instance now the interferer signal can change its turning direction at a random instance that we call ϕ_t. Most generalized form of the received signal during a chip interval is:

$$R(\omega t) = \begin{cases} \sqrt{C}e^{j(b_C\omega t+\phi_c)} + \sqrt{I}e^{j(b_{I_1}\omega t+\phi_I)}, & \omega t = [0, \phi_t] \\ \sqrt{C}e^{j(b_C\omega t+\phi_c)} + \sqrt{I}e^{j(b_{I_2}\omega t+d)}, & \omega t = [\phi_t, \pi/2] \end{cases}$$

where b_C defines the turning direction of transmitted chip while b_{I_1} and b_{I_2} are turning directions of sequential chips of the interferer. In the interval $[0, \phi_t]$ interferer transmits a chip then in the following interval $[0, \phi_t]$ it transmits another chip.

We already found the probabilities of P_{eq} and P_{dif} in Sec. 5.4.0.1 the new one is P_{mix} which corresponds to the different sequential chips of the interferer. All combinations are shown in Table 5.1. With respect to the occurrences of the probability of error values at the table P_c will be:

$$P_c = 1/4P_{eq} + 1/4P_{dif} + 1/2P_{mix} \tag{5.21}$$

Table 5.1: Useful and Interferer Chip Phase Combinations

b_C	b_{I_1}	b_{I_2}	d	Probability of Error
+1	+1	+1	ϕ_I	$P_{eq} = 0$
+1	+1	-1	$\phi_I + 2\phi_t$	P_{mix}
+1	-1	+1	$\phi_I - 2\phi_t$	P_{mix}
+1	-1	-1	ϕ_I	P_{dif}
-1	+1	+1	ϕ_I	P_{dif}
-1	+1	-1	$\phi_I + 2\phi_t$	P_{mix}
-1	-1	+1	$\phi_I - 2\phi_t$	P_{mix}
-1	-1	-1	ϕ_I	$P_{eq} = 0$

To find out P_c we have to obtain P_{mix}. Let's analyze the $b_C = +1$, $b_{I_1} = +1$, $b_{I_2} = -1$, and $d = \phi_I + 2\phi_t$ case. Following the similar way in Section 5.4.0.1 we can obtain the test function as

$$\frac{\sqrt{C} - \sqrt{I}\cos(\phi_I + 2\phi_t)}{\sqrt{I}\sin(\phi_I + 2\phi_t)} \underset{e}{\overset{s}{\gtrless}} \frac{\sqrt{I}\sin(\phi_I)}{\sqrt{C} + \sqrt{I}\cos(\phi_I)} \tag{5.22}$$

(5.22) can be deduced to

$$C + \sqrt{CI}(\cos(\phi_I) - \cos(\phi_I + 2\phi_t)) - I\cos(2\phi_t) \underset{e/s}{\overset{s/e}{\gtrless}} 0 \tag{5.23}$$

In (5.23) now the direction of the inequality depends on the signs of denominators in (5.22). Since we are interested on how (5.23) results respect to C/I we can set $C = 1$ then (5.23) becomes

$$I\cos(2\phi_t) - \sqrt{I}(\cos(\phi_I) - \cos(\phi_I + 2\phi_t)) - 1 \underset{e/s}{\overset{s/e}{\gtrless}} 0 \tag{5.24}$$

P_{mix} can be obtained by using the test in (5.24) considering the different signs of the denominators in (5.22) and the definition domain of arctan function. Finally P_c will be as in (5.21).

5.4.1 Validation of the Analytical Model through Simulations

After obtaining P_c for synchronous and asynchronous interferers we have done simulations in MATLAB as described in Sec.5.4.1. Simulation results are perfectly aligned

with the analytical results shown in Fig. 5.7. In our systematical approach first we have obtained P_c in synchronous chips of interferer, in fact it is not a realistic scenario but it served to verify and simplify the calculations, since synchronous symbols of interferer is not realistic at the rest of the chapter we'll just refer to the asynchronous scenario.

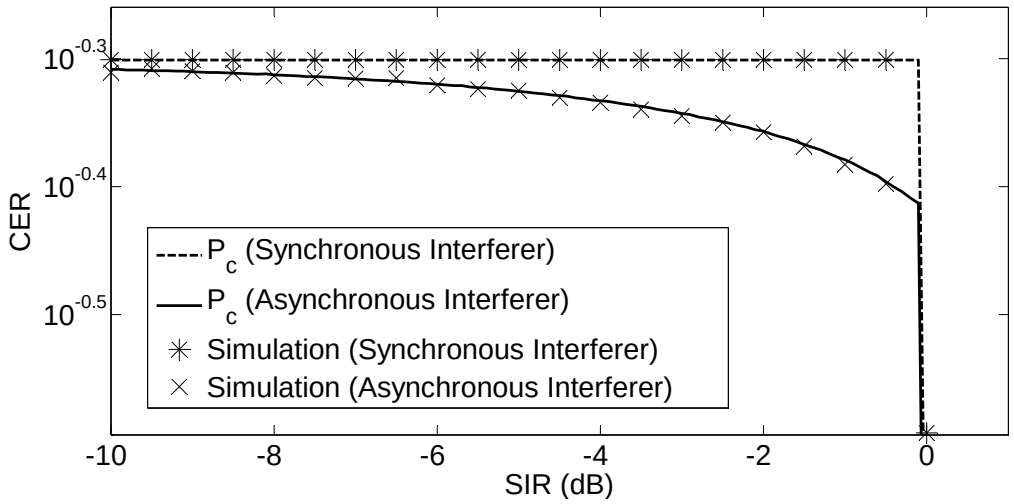

Figure 5.7: Non-coherent Chip Error Rate

5.5 Alternative Demodulator 1

In Sec. 5.3.2 we have obtained the performance of an ideal maximum likelihood demodulator in coherent case but implementing such an optimum demodulator can be complex from the hardware point of view. Here in this section and the following one we will introduce two alternative demodulators constructed by using the blocks that we have found out so far and we will analyze these demodulators' performance with one interferer. For instance we'll start reconsidering the demodulator block in Sec. 5.4. There can be defined a coherent version of this demodulator by using the initial phase, $\Theta_R(0)$, to differ between the symbol points rather than just detecting the phase transitions. As shown in Fig. 5.6(a), in fact, symbol points (i.e. m_0, m_1, m_2, m_3) are transmitted in even chip intervals and always the angle between I-phase and Q-phase starts from 0 and π radians and goes to $\pi/2$ and $3\pi/2$ radians then in the odd chip interval the angle turns back to 0 and π radians again. Therefore the transmitted signal

in the even chip duration can be written like:

$$T(\omega t) = \sqrt{C}e^{\pm j(\omega t+\phi_C)}, \omega t = [0, \pi/2], \phi_C = \{0, \pi\} \tag{5.25}$$

Now the maximum likelihood estimation for ϕ_C can be defined as in (5.26a) which is basically checking the received ϕ_C is whether in the left or in the right half side of the Q-I plane in the beginning of even chip duration. Then we can combine this information with (5.16) to demodulate the symbol point transmitted at that chip interval.

$$\hat{\phi_C} = \begin{cases} 0, & \Theta_R(0) = (-\pi/2, \pi/2) \\ \pi, & \Theta_R(0) = (\pi/2, 3\pi/2) \end{cases} \tag{5.26a}$$

$$\Theta_R(\pi/2) \underset{-j}{\overset{+j}{\gtrless}} \Theta_R(0) \tag{5.26b}$$

If the estimation $\hat{\phi_C}$ is 0 then according to the phase transition, m_3 or m_2 can be demodulated instead if $\hat{\phi_C}$ is 1 then m_0 or m_1 can be demodulated as shown in Table 5.2.

Table 5.2: Symbol Point Decision

$\hat{\phi_C}$	phase transition	symbol point
0	$+j$	$m_3(11)$
0	$-j$	$m_2(10)$
π	$+j$	$m_0(00)$
π	$-j$	$m_1(01)$

Let's obtain the probability of error in (5.26a). As stated in (5.27) probability of estimation error when $\phi_C = 0$ or $\phi_C = \pi$ is equal because of the symmetry.

$$P(e|\phi_C) = P(\Theta_{R(\omega t)} = (\pi/2, 3\pi/2)|\phi_C = 0) = P(\Theta_{R(\omega t)} = (-\pi/2, \pi/2)|\phi_C = \pi) \tag{5.27}$$

So that we can analyze transmitted initial phase of $\phi_C = 0$. As represented in Fig. 5.8 the transmitted signal is shown with the arrow from origin to $\sqrt{C}$ then the received signal respect to the phase of the interferer at the sampling instance can go anywhere over the circle with the center at $(0, \sqrt{(C)})$.

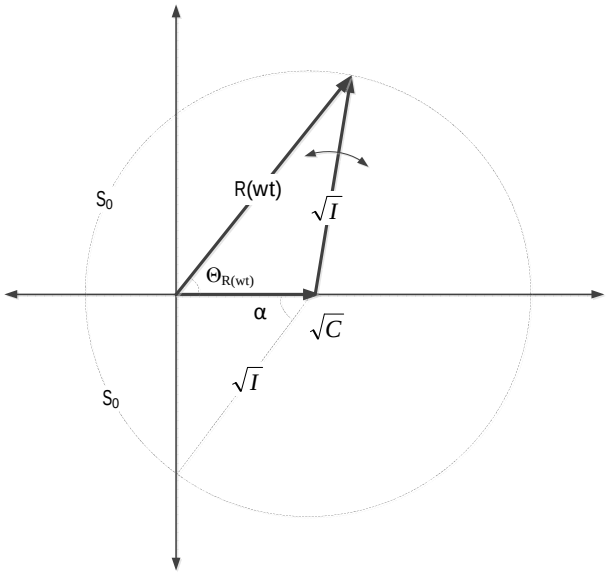

Figure 5.8: $R(\omega t)$ diagram when $\phi_C = 0$

Therefore the probability of error will be the ratio between $2S_0$ and 2π

$$P(e|\phi_C) = \frac{2S_0}{2\pi} = \frac{2\alpha}{2\pi} = \frac{\alpha}{\pi} \tag{5.28}$$

where

$$\alpha = \arctan(\sqrt{\frac{I}{C} - 1}) \tag{5.29}$$

We have already obtained the error rate of decision test in (5.26b) in Sec. 5.4 and named it as P_c. Normally probability of error in (5.26a) and in (5.26b) are dependent because of the common term $\Theta_R(0)$ but as seen in Fig.5.8 when $\sqrt{C/I}$ goes to 1, S_0 goes to 0 while diminishing the dependency between them. So in order to obtain a upper bound for the joint probability of error we can simply treat these two equations as if they are independent. In Table 5.3 there is a summary of probabilities in 5.26.

Table 5.3: Probabilities in Equation 5.26

	probability of error	probability of success
(5.26a)	$P(e\|\phi_C)$	$1 - P(e\|\phi_C)$
(5.26b)	P_c	$1 - P_c$

Therefore we can obtain a upper bound for Symbol Error Rate (SER) in alternative demodulator 1 as:

$$P_{s_1} \geq 1 - (1 - P_c)(1 - P(e|\phi_C)) \tag{5.30}$$

On the other hand for the CER we have to consider number of chip errors respect to the symbol point errors. Possible errors are shown in Table 5.4

Table 5.4: Combinations of Different Errors

(5.26a)	(5.26b)	symbol point errors	number of chip errors
success	success	No	0
success	fail	$m_3(11) \leftrightarrows m_2(10)$	1
		$m_1(01) \leftrightarrows m_0(00)$	
fail	success	$m_3(11) \leftrightarrows m_0(00)$	2
		$m_1(01) \leftrightarrows m_2(10)$	
fail	fail	$m_3(11) \leftrightarrows m_1(01)$	1
		$m_0(00) \leftrightarrows m_2(10)$	

Using the probabilities in Table 5.3 and the number of chip errors in Table 5.4 the lower bound for the CER in alternative demodulator 1 will be:

$$P_{c_1} \geq \frac{1}{2}P_c(1 - P(e|\phi_C)) + (1 - P_c)P(e|\phi_C) + \frac{1}{2}P(e|\phi_C)P_c \tag{5.31}$$

5.6 Alternative Demodulator 2

Actually there is still room for the improvement we can correct some erroneous receptions by extending the observation interval to the previous and next chips instead of observing only the even chip interval. For this reason here in this section just to show that it can be defined better demodulators that can perform similar to the optimal coherent demodulator of Sec.5.3.2 we will define another demodulator which is not very sophisticated but performing better than the alternative demodulator 1 because of extended observation duration which covers three chips duration. The reason why we are extending the observation to three chips interval, is because of the offset in the modulation. A single symbol point takes place in three sequential chip intervals. Since

the purpose is just to show that there is still room for the improvement we will not analyze this new demodulator mathematically but we will just use it in the simulation.

As we have seen in Fig. 5.6 not every phase change can happen after or before a chip interval, permitted combinations are given in Table 5.5.Every t_i or m_i in Table 5.5 has its own features like initial phase and the phase change, we can create unique feature vectors by mapping these features to binary numbers as shown in Table 5.6

Table 5.5: Permitted Sequences

m_0			m_1			m_2			m_3		
c_{-1}	c_0	c_{+1}	c_{-1}	c_0	c_{+1}	c_{-1}	c_0	c_{+1}	c_{-1}	c_0	c_{+1}
t_5	m_0	t_4	t_4	m_1	t_5	t_7	m_2	t_6	t_7	m_3	t_5
t_5	m_0	t_6	t_4	m_1	t_7	t_7	m_2	t_4	t_7	m_3	t_7
t_4	m_0	t_4	t_5	m_1	t_5	t_6	m_2	t_6	t_6	m_3	t_5
t_4	m_0	t_6	t_5	m_1	t_7	t_6	m_2	t_4	t_6	m_3	t_7

Table 5.6: Feature Mapping

	Feature	Binary Number
Phase Change	$+j$	1
	$-j$	0
Initial Phase	0	00
	$\pi/2$	01
	pi	10
	$3\pi/2$	11

Now we can rewrite Table 5.5 as shown in Table 5.7 by using the mapping in Table 5.6. Accordingly we can extract the features from the received signal observing through three chip intervals and we can find the correlation between the received features and the permitted feature vectors. For each symbol point (i.e m_0, m_1, m_2, m_3) there are four permitted feature vector so there will be four correlation values. The demodulator sums this four correlation values and obtains a value for each symbol point then it chooses the maximum.

In Fig. 5.9 there are CER values for the all half-sine shaped demodulators that we have analyzed. As expected non-coherent demodulator performs worse while coherent

Table 5.7: Permitted Feature Vectors (Duration: $3T_c$)

m_0	m_1	m_2	m_3
011100110	110101011	010000111	010001011
011100111	110101010	010000110	010001010
110100110	011101011	111000111	111001011
110100111	011101010	111000110	111001010

is performing the best. Our alternative demodulators are performing in between these two. On the right side of 0dB there is no chip error.

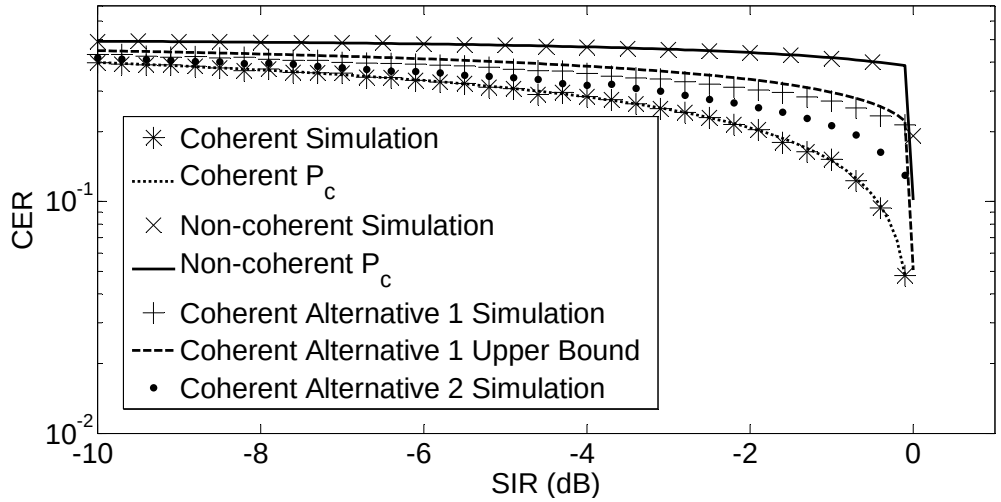

Figure 5.9: Chip Error Rate

5.7 Probability of Data Error in IEEE 802.15.4

In the last section we will find an upper bound for P_d which stands for the probability of error for the data symbol in IEEE 802.15.4. Commercial transceivers like TI CC2420 [29] and Freescale MC13224 [65] first demodulate O-QPSK signal then digitally de-spread it to the data symbols through correlation. In Sec.5.3 and Sec.5.4 we have obtained the chip error rate, P_c, here in this section we will map this to the probability of error in data symbols by finding out a union upper bound and at the final stage we will use this bound to find a lower bound to the packet reception rate, P_r.

5.7.1 Chip Error Rate to Data Symbol Error Rate

In the 2450 MHz PHY of IEEE 802.15.4 data symbols are mapped to the chips as shown in Table 5.8. Because of the symmetry in the spreading symbol set probability of data symbol, P_d, when S_0 is transmitted is equal to the probability of error for any other data symbol, S_i, in the set as formulated in (5.32) where $i\epsilon\{1, 2, ..15\}$. (A brief discussion about properties of spreading set of 802.15.4 2.4GHz PHY can be found in [68])

$$P_d = P(e|S_i) = P(e|S_0) \tag{5.32}$$

Therefore for the error analysis we can assume that S_0 is transmitted without loss of generality. There should be a union upper bound for $P(e|S_0)$ which can be calculated as the sum of each probability of error that transforms S_0 to any S_i.

$$P(e|S_0) \leq \sum_{i=1}^{15} P(S_i|S_0) \tag{5.33}$$

Each probability in the summation of (5.33) can be formulated as a function of hamming distance, h_i, and chip error rate, P_c, where h_i is the hamming distance between S_0 and S_i.

$$P(S_i|S_0) = P(h_i, P_c) = \begin{cases} \frac{1}{2}\binom{h_i}{h_i/2} P_c^{h_i/2}(1-P_c)^{h_i/2} + \sum_{i=\frac{h_i+2}{2}}^{h_i} \binom{h_i}{i} {P_c}^i (1-P_c)^{h_i-i} & , \ h_i \text{ is even} \\ \sum_{i=\frac{h_i+1}{2}}^{h_i} \binom{h_i}{i} {P_c}^i (1-P_c)^{h_i-i} & , \ h_i \text{ is odd} \end{cases} \tag{5.34}$$

In (5.34) we are summing all the probabilities of having a given number of chip error in the received signal which makes it more close to S_i in the hamming space. If h_i is even number, the chip sequence can be exactly at the same distance to both S_0 and S_i when there are $\frac{h_i}{2}$ chip errors in the received chip sequence. This is the reason why there is $\frac{1}{2}$ at the first addend of (5.34) when h_i is even, on the other hand when h_i is an odd number there will never be such a case.

Now we have a generic upper bound which can be applied to any symmetric chip sequence set. In the following two sub-sections we will use it in order to find the upper bounds of data symbols of coherent and non-coherent demodulators that we analyzed in Sec.5.3.2 and Sec.5.4 respectively.

Table 5.8: Chip Sequences in 2450 MHz PHY of IEEE 802.15.4

Data symbol	Data bits	Chip values
S_0	0000	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
S_1	1000	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
S_2	0100	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
S_3	1100	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
S_4	0010	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
S_5	1010	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
S_6	0110	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
S_7	1110	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
S_8	0001	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
S_9	1001	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
S_{10}	0101	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
S_{11}	1101	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
S_{12}	0011	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
S_{13}	1011	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
S_{14}	0111	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
S_{15}	1111	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

5.7.1.1 Coherent O-QPSK

In coherent case we are able to identify each chip so in order to find the hamming distances between the data symbols we simply use the chip sequences in Table 5.8. All the hamming distances between the data symbols are given in Table 5.9. It is worth to note that in Table 5.9 the quantity of the different values in each column and row is the same because of the symmetry in the set. The reoccurrences of these values for each column or row are given in the Table 5.10.

We can write the upper bound for the probability of P_d as a function of P_c as the following:

$$P_d \leq 2P(12, P_c) + 2P(14, P_c) + 3P(16, P_c) + 2P(18, P_c) + 6P(20, P_c) \tag{5.35}$$

where we have used $P(h_i, P_c)$ in the (5.34) with the coefficients coming from the reoccurrences in the Table 5.10

Table 5.9: Hamming Distances between the Data Symbols in Coherent Demodulation

Data Symbol	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
S_0	0	16	18	20	20	20	18	16	16	12	14	20	20	20	14	12
S_1	16	0	16	18	20	20	20	18	12	16	12	14	20	20	20	14
S_2	18	16	0	16	18	20	20	20	14	12	16	12	14	20	20	20
S_3	20	18	16	0	16	18	20	20	20	14	12	16	12	14	20	20
S_4	20	20	18	16	0	16	18	20	20	20	14	12	16	12	14	20
S_5	20	20	20	18	16	0	16	18	20	20	20	14	12	16	12	14
S_6	18	20	20	20	18	16	0	16	14	20	20	20	14	12	16	12
S_7	16	18	20	20	20	18	16	0	12	14	20	20	20	14	12	16
S_8	16	12	14	20	20	20	14	12	0	16	18	20	20	20	18	16
S_9	12	16	12	14	20	20	20	14	16	0	16	18	20	20	20	18
S_{10}	14	12	16	12	14	20	20	20	18	16	0	16	18	20	20	20
S_{11}	20	14	12	16	12	14	20	20	20	18	16	0	16	18	20	20
S_{12}	20	20	14	12	16	12	14	20	20	20	18	16	0	16	18	20
S_{13}	20	20	20	14	12	16	12	14	20	20	20	18	16	0	16	18
S_{14}	14	20	20	20	14	12	16	12	18	20	20	20	18	16	0	16
S_{15}	12	14	20	20	20	14	12	16	16	18	20	20	20	18	16	0

Table 5.10: Reoccurrences of Unique Hamming Distance Values

h_i:	12	14	16	18	20
reoccurrence:	2	2	3	2	6

5.7.1.2 Non-coherent O-QPSK

Regarding to the non-coherent case the calculation is different because in non-coherent demodulation we can only identify the phase transitions during the chip intervals. For 32-chip long sequence there will be 31 phase transitions as shown in Table 5.11. Where $+$ means $+\pi/2$ phase change and $-$ means $-\pi/2$ phase change in the transmitted signal.

We can think the phase transition sequence of a data symbol in Table 5.11 as a vector in hamming space then we can calculate the hamming distances between these vectors. The results are given in Table 5.12. Similar to the Table 5.9 unique hamming distance values (shown in Table 5.13) are all same in different rows and columns of the

Table 5.11: Phase Transitions of Chip Sequences in 2450 MHz PHY of IEEE 802.15.4

Data Symbol	Data Bits	Phase Transitions
S_0	0000	+ + - - - - - - + + + - + + + + + - + - + + + - - + + - + + - -
S_1	1000	+ - - + + + - - - - - - + + + - + + + + + - + - + + + - - + + -
S_2	0100	+ + - + + - - + + + - - - - - - + + + - + + + + + - + - + + + -
S_3	1100	+ + - - + + - + + - - + + + - - - - - - + + + - + + + + + - + -
S_4	0010	- + - + + + - - + + - + + + - - + + + - - - - - - + + + - + + +
S_5	1010	+ + + + - + - + + + - - + + - + + + - - + + + - - - - - - + + +
S_6	0110	+ + + - + + + + + - + + - + + + - - + + - + + + - - + + + - - - - -
S_7	1110	- - - - + + + + - + + + + + - + - + + + + - - + + - + + + - - + + + -
S_8	0001	- - + + + + + + + - - - + - - - - + - + - - - + + + - - + - - + +
S_9	1001	- + + - - - + + + + + + + - - - + - - - - + - + - - - + + + - - +
S_{10}	0101	- - + + - - + + - - - + + + + + + + - - - + - - - - + - + - - - +
S_{11}	1101	- - + + - - + + - - + + - - - + + + + + + + - - - + - - - - + - +
S_{12}	0011	+ - + - - - + + + - - + + - - + + + - - - + + + + + + + - - - + - - -
S_{13}	1011	- - - - + - + - - - + + + - - + + - - + + + - - - + + + + + + + - - -
S_{14}	0111	- - - + - - - - + - + - - - + + + - - + + - - + + + - - - + + + + +
S_{15}	1111	+ + + + - - - + - - - - + - + - - - + + + - - + + - - + + + - - - +

table.

Finally, we can write the upper bound for the probability of P_d as a function of P_c like the following:

$$P_d \leq 2P(13, P_c)+2P(14, P_c)+3P(15, P_c)+3P(16, P_c)+2P(17, P_c)+2P(18, P_c)+P(31, P_c) \tag{5.36}$$

where we have used $P(h_i, P_c)$ in the (5.34) with the coefficients coming from the reoccurrences in the Table 5.13

5.7.2 Packet Reception Rate

In 802.15.4 2.4 GHz PHY every four data bits is mapped to a data symbol as given in Table 5.8. Therefore in order to transmit a byte two data symbols should be transmitted. When a b-byte long packet is transmitted, the probability of receiving the

Table 5.12: Hamming Distances between the Phase Transition Vectors of Data Symbols

Data Symbol	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}
S_0	0	14	15	14	16	13	15	13	31	17	16	17	15	18	16	18
S_1	14	0	13	16	14	15	13	15	17	31	18	15	17	16	18	16
S_2	15	13	0	13	15	14	16	14	16	18	31	18	16	17	15	17
S_3	14	16	13	0	14	15	13	15	17	15	18	31	17	16	18	16
S_4	16	14	15	14	0	13	15	13	15	17	16	17	31	18	16	18
S_5	13	15	14	15	13	0	14	16	18	16	17	16	18	31	17	15
S_6	15	13	16	13	15	14	0	14	16	18	15	18	16	17	31	17
S_7	13	15	14	15	13	16	14	0	18	16	17	16	18	15	17	31
S_8	31	17	16	17	15	18	16	18	0	14	15	14	16	13	15	13
S_9	17	31	18	15	17	16	18	16	14	0	13	16	14	15	13	15
S_{10}	16	18	31	18	16	17	15	17	15	13	0	13	15	14	16	14
S_{11}	17	15	18	31	17	16	18	16	14	16	13	0	14	15	13	15
S_{12}	15	17	16	17	31	18	16	18	16	14	15	14	0	13	15	13
S_{13}	18	16	17	16	18	31	17	15	13	15	14	15	13	0	14	16
S_{14}	16	18	15	18	16	17	31	17	15	13	16	13	15	14	0	14
S_{15}	18	16	17	16	18	15	17	31	13	15	14	15	13	16	14	0

Table 5.13: Reoccurrences of Unique Hamming Distance Values

h_i:	13	14	15	16	17	18	31
reoccurrence:	2	2	3	3	2	2	1

useful packet successfully, P_r, with an interferer which concurrently is transmitting, is straightforward after having P_d:

$$P_r = (1 - P_d)^{2b} \tag{5.37}$$

Upper bounds of P_d in (5.35) and (5.36) will became lower bounds in (5.37). In Fig.5.10, P_r results for the coherent and non-coherent demodulators that we analyzed in Sec.5.3 and Sec.5.4 with the simulations are shown.

Simulation results without noise tightly fit the lower bounds that we have obtained by verifying the mathematical analysis. In addition to this, in the figure there are two more non-coherent simulation results with $SNR_i = 20dB$ (later on we will explain

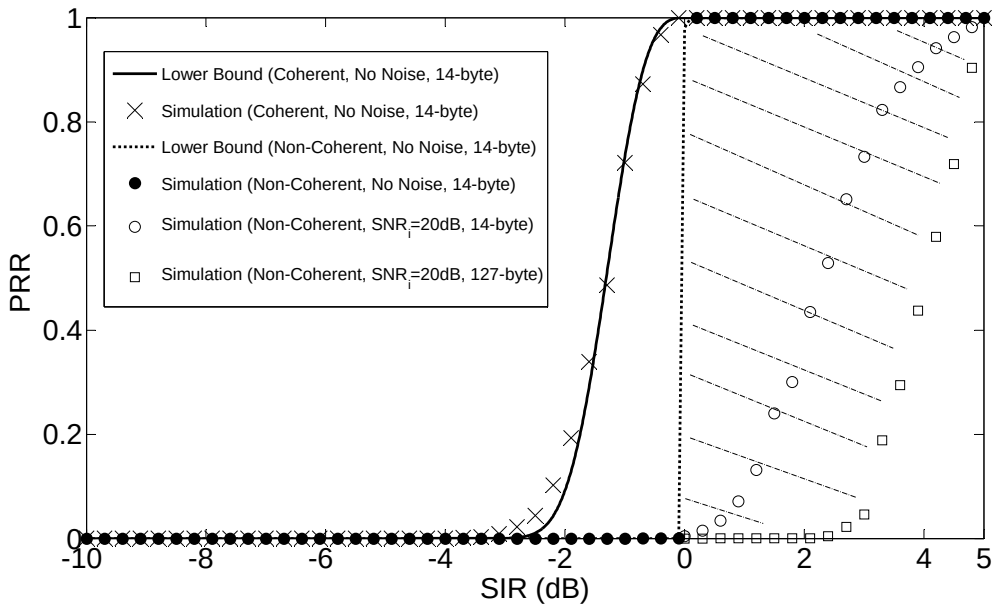

Figure 5.10: Packet Reception Rate

that it is the beginning of the gray region of the 802.15.4 transceivers) in order to demonstrate the effect of SNR on the mathematical model. We defined SNR_i as the ambient SNR while SNR_o as the SNR after considering the imperfections of the receive path of the transceiver. In order to model this non-ideal receive path we used the noise figure definition [70]:

$$n_f = \frac{n_o}{G n_i}$$

where n_i is the noise power at the input of the receiver path, n_o is the noise power at the output, and G is the gain of the receiver. We can find the relation between the SNR at the input and at the output with the help of noise figure as:

$$SNR_o = \frac{SNR_i}{n_f} \tag{5.38}$$

Until here the analysis is generic and can be applied to any kind of transceiver with the given n_f value. By using the receiver sensitivity of the transceiver n_f can be calculated

as[73]:

$$n_f = \frac{S_{min}}{k \cdot T \cdot B \cdot SNR_{min}} \tag{5.39}$$

where $k = 1.38 \cdot 10^{-23}$ is Boltzmann's constant, $T = 290K$ is absolute temperature, S_{min} is the receiver sensitivity, $B = 2MHz$ is the bandwidth in IEEE 802.15.4 2.4GHz PHY and $SNR_{min} \approx 3dB$ is the minimum required signal-to-noise ratio for the non-coherent demodulation of the signal with Additive White Gaussian Noise (AWGN)[28]. In order to compare the simulation results with experiments coming from [48, 76, 94] and Chapter 4 we need the receiver sensitivity of the used devices, which are Chipcon CC2420 and Freescale MC1322 USB dongle. Actually both transceiver has the same receiver sensitivity that is typically $-95dB$ [66, 97]. Therefore for all cited experiments above, n_f should be around $13dB$ (commercial transceivers using IEEE 802.15.4 $2.4GHz$ PHY typically have $10dB$ to $13dB$ noise figures [28, 33]). In order to model the non-ideal receiver and demonstrate a great variety of ambient SNR conditions in the simulations, we have chosen to find the PRR region which starts from the border of the gray region and goes to the without noise condition. This was the result of not having exact SNR values in concurrent transmission measurements that we have found in the literature. Since the intention is to find the solely relation between SIR and PRR, as expected, SNR values during the measurements are not given, but just the Signal-to-Interference-Plus-Noise Ratio (SINR) to PRR curves are reported. In [48] measurements in Fig.1 suggested us that roughly 7dB should be a good estimation for the border of the gray region therefore we have chosen $SNR_i = 20dB$, which is $7dB$ more than $n_f = 13dB$ (noise figure that we calculated by (5.39)), as the border of the gray region. White gaussian noise with the SNR that is calculated by (5.38) is added to the useful signal and PRR is found for 14-byte (experiments in Chapter 4 is conducted with this packet size) and 127-byte (maximum allowable) PSDUs as shown in Fig.5.10. In fact, an interval from $SNR_i = 20dB$ to $SNR_i = \infty$, which should be adequate to represent great variety of different SNR conditions and the other possible imperfections due to the low-cost, low-power transceiver design, results with a considerably narrow region of PRR as shown in Fig.5.10. All the measurements mentioned in [48, 76, 94] and Chapter 5 stay in this region. Thanks to the relatively narrow PRR region, it would not be wrong to conclude that the mathematical analysis captures the essential features of the real-life

performance since measurement points lie on the region that we believe represents all operation conditions. It is also worth to note that $0dB$ is the absolute minimum SIR necessary to start receiving the useful packets.

5.8 Conclusion

In this chapter with the aim of finding mathematical explanations to the measurement results in [48, 76, 94] and Section 4.2, concurrent transmission in 802.15.4 is analyzed. First, we obtained analytical expressions to the chip error rate, P_c, in non-coherent and coherent O-QPSK demodulation in the presence of one interferer transmitting concurrently. We did not go further on the analysis with more than one interferer since it was already confirmed with experimental measurements in [76] and Section 4.2 that the increasing number of interferers does not change the PRR conditioned to the SIR. We also studied two alternative demodulators which are performing in between coherent and non-coherent demodulators. Later in the last section we have transformed chip error rate, P_c, to the data symbol error rate, P_d in 2.4 GHz PHY of IEEE 802.15.4 using an union upper bound, which was tight. Finally, we have used the upper bound to obtain a lower bound to the PRR. All parts of analytical analysis is verified with a versatile simulator written in MATLAB which is able to test different demodulators. In non-coherent case we have found 0dB as the absolute minimum necessary SIR to be able to receive useful packets, moreover we have showed that in the region starting from the border of the gray region to the $SNR = \infty$ which should be adequate to represent all operation conditions of typical IEEE 802.15.4 transceivers, measurements in the literature [48, 76, 94] and in Section 4.2 are in agreement with our analysis.

Chapter 6

IEEE 802.11 Signal Strength Detection

In this chapter details about the sensing part of the Centrale Adriatica Project, which is introduced in Chapter 1, will be explained and a simple method in order to calibrate the ED Scan measurements will be presented. In Centrale Adriatica Project, ZigBee nodes sense the energy levels in each IEEE 802.15.4 channel by using ED Scan requests and sent this information to the sinks in every 10 minutes in order to estimate the signal strength of 802.11 network in the warehouse. A view of the warehouse is given in Fig. 6.1). The warehouse is equipped with a number of 802.11 Access Point (AP) and workers wirelessly communicate with a core network through the APs. The building contains grocery stocks for supplying several supermarkets, hence materials as plastic, metal cans, bottles, liquids, all having different dielectric properties, are continuously fed and withdrawn, so that the reflective/refractive characteristics of the environment change over time. This results in coverage quality being dynamic. In the warehouse 228 ZigBee nodes are deployed in order to sense the energy levels on the channels used by 11 APs working in 802.11b mode as indicated in Fig. 6.2. These APs provides VoIP and data services to the mobile operators working in the warehouse. The scope of this chapter is limited on the sensing the signal strength of 802.11 network. Details about the 802.15.4 network and the application are provided in Section 8.

Figure 6.1: A View of the Warehouse

6.1 Framework and Methodology

IEEE 802.11 coverage is hard to monitor and signals received from AP are variable in strength due to both propagation phenomena and coexistence with other ISM devices [100]. One particular case that deserves attention is when APs are deployed in large scale business locals, such as warehouses and factories. Apart from interference, which can be controlled with smart 802.11 channel assignment, unevenness of signal propagation due to topology, obstacles and presence of objects of different materials, is a fundamental issue to cope with. Monitoring the electromagnetic energy transmitted by the APs is performed by multiple battery-powered ZigBee-compliant devices located in the high rack storage area around 1.5 meters above the ground. The key factor is a novel use of the ED Scan functionality implemented in the IEEE 802.15.4 standard.

In IEEE 802.15.4, first eleven channels (0-10) are reserved for the 868/915 MHz frequency bands while the other sixteen channels (11-26) are utilized in the 2.4 GHz ISM band. The central frequencies in 2.4 GHz are given as:

$$f_c = 2405 + 5(k - 11)$$

where k is the channel number. The spacing between central frequencies are 5MHz and the bandwidth of each channel is 2MHz. On the other hand in 802.11 each channel spans 22MHz, and there are 11 channels with 3 of them are non-overlapping as illustrated in Fig. 6.3. Therefore for each 802.11 channel there are four unique 802.15.4 channels residing in the 22MHz bandwidth. Energy levels in these four channels can be used to estimate the RSS of 802.11.

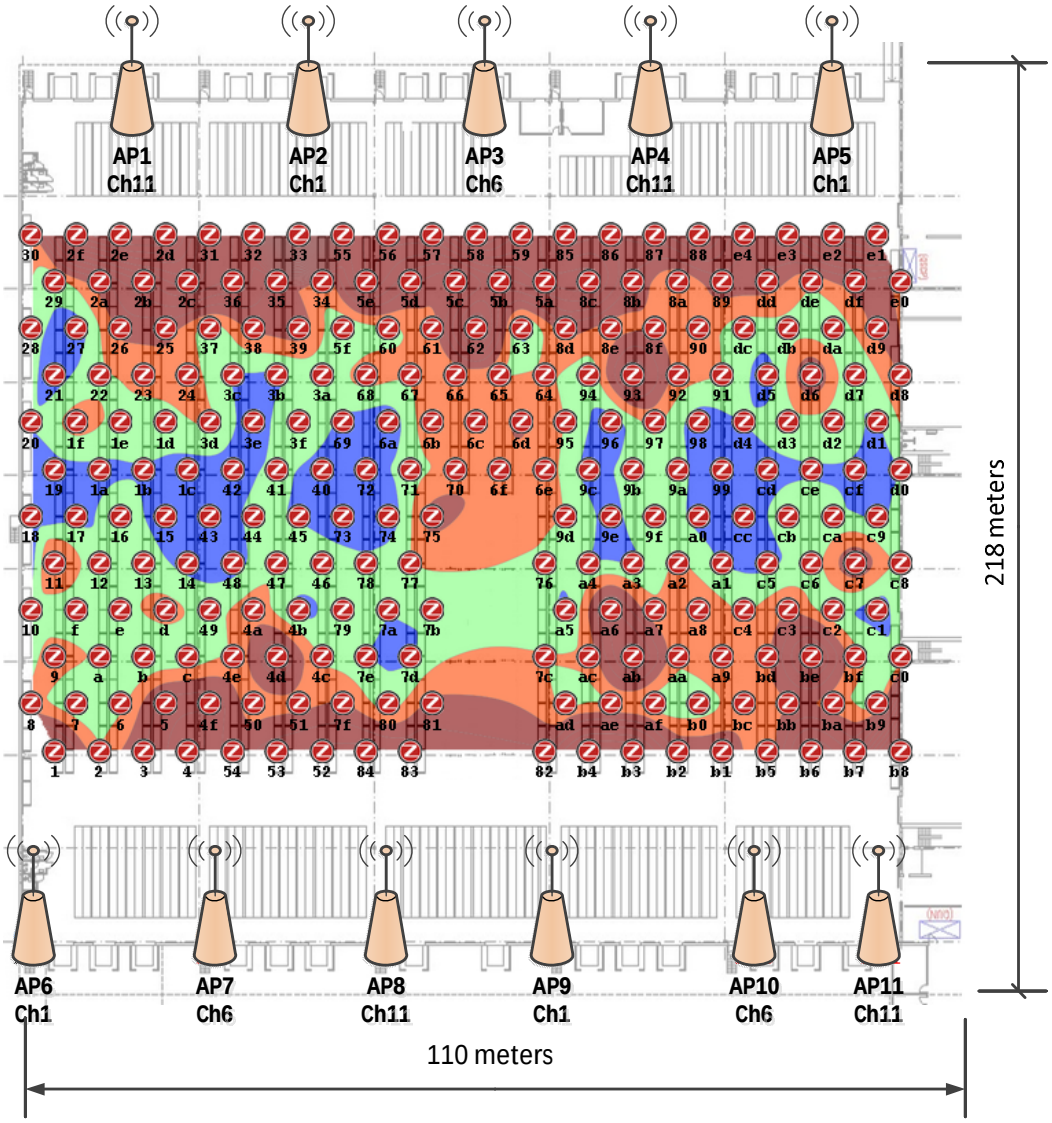

Figure 6.2: Layout of the Warehouse

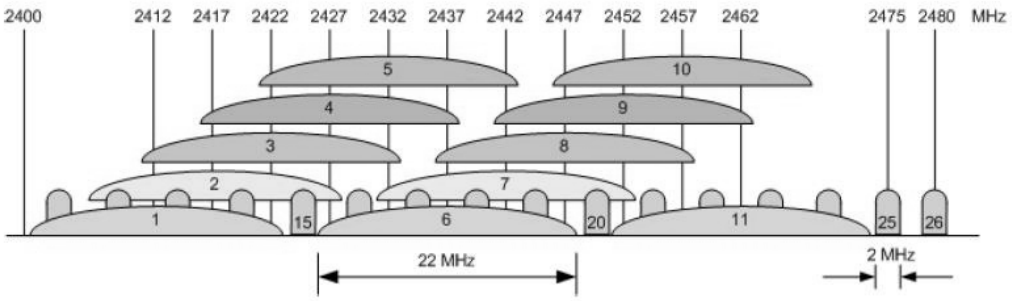

Figure 6.3: Coexistence of 802.15.4 and 802.11 in 2.4 GHz ISM Band

In 802.15.4, an ED scan allows a device to obtain the measure of the peak energy in each requested channel. The ED scan over a specified set of channels is requested using the MLME-SCAN.request primitive with the ScanType parameter set to ED Scan. After receiving the request, for each channel, the MAC Sublayer Management Entity (MLME) first switches to the channel, then performs an ED measurement for

$$aBaseSuperframeDuration(2^n + 1) \tag{6.1}$$

where n is the value of the ScanDuration parameter in the MLME-SCAN.request primitive and aBaseSuperframeDuration is equal to 15,360 ms. The ED measurement is performed by the MLME issuing repetitive PLME-ED.requests to the Physical Layer Management Entity (PLME) and the maximum value is stored by the MLME as the result of the ED Scan as illustrated in Fig.6.4. PLME-ED.request returns an estimate of the received signal power within the bandwidth of the channel. The PLME-ED measurement time is equal to 8 symbol periods (128μs). The result is reported to the MLME as an 8 bit integer ranging from 0 to 255 corresponding to -15dBm and -100dBm values respectively [64]. The linear relationship is given in (3.2).

In 802.11, APs periodically send beacons in order to announce their capabilities and to synchronize the clients in the network. Even though there is no data traffic in the network thanks to the periodic beacons there is always activity in the channels. This can be exploited in order to assess the quality of the signal in a particular position. The default beacon interval is 100ms, for this reason during the MLME-SCAN at least one beacon should be captured, thus the parameter n in (6.1) is set to 4 which results

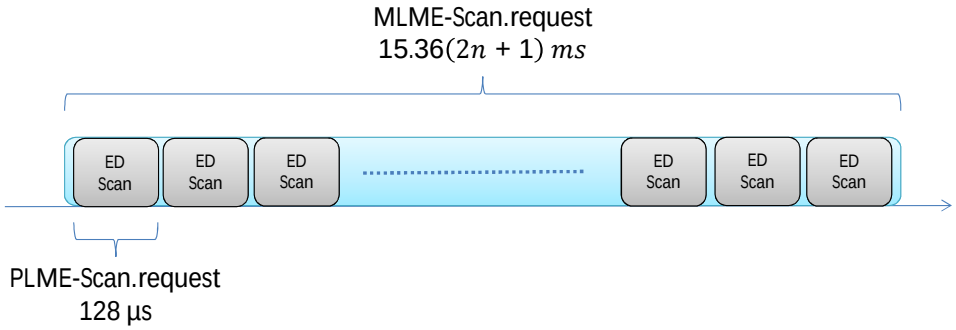

Figure 6.4: ED Scan

261 ms long MLME-Scan. For all 16 channels in 2.4 Ghz band three sequential MLME-SCAN.request are done to estimate the average RSS. In other words, peak energy level in each 802.15.4 channel has been measured three times in roughly 4.2 seconds intervals as illustrated in Fig.6.5. Nodes receiving the synchronization packet from the sink, two seconds after started MLME-Scan measurements. Synchronization and the network layer related issues are explained in Chapter 8.

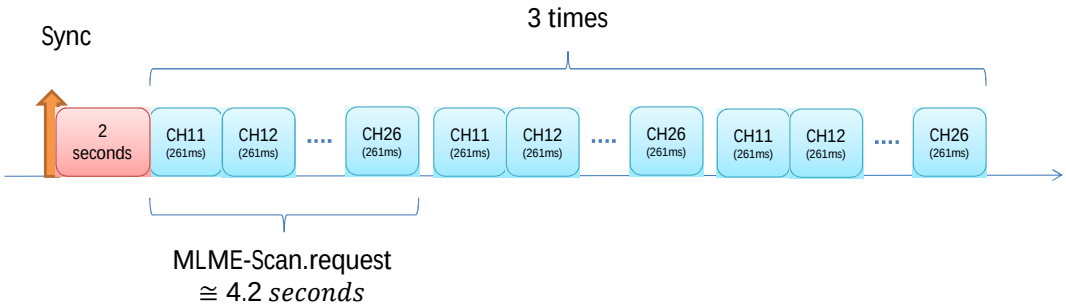

Figure 6.5: ED Scan Timing

6.2 Calibration

In the previous section the idea of using the four 802.15.4 channels that are overlapping with a given 802.11 channel in order to estimate RSS of 802.11 has been discussed but without finding a relationship between the 802.11 spectrum, the RSS value reported in a typical 802.11 client and the ED Scan measurements this is not an easy task. Hence in this section in the light of the measurements conducted in the warehouse by using a spectrum analyzer and a 802.11 client, an estimation to the RSS will be derived.

By using a notebook PC which is equipped with Intel Wireless WiFi Link 5100 network card and Wi-Spy 2.4x Spectrum Analyzer a series of measurement have been conducted in six different positions throughout an aisle in the warehouse. The measurements are done next to the sensors B1, A8, A0, 99, 91, 88 by observing the best RSS value reported with inSSIDer software and the peak power in the spectrum measured by Chanalyzer software in the 802.11 channels 1, 6, 11. For instance, a snapshot of the spectrum analyzer nearby sensor A8 is given in Fig.6.6

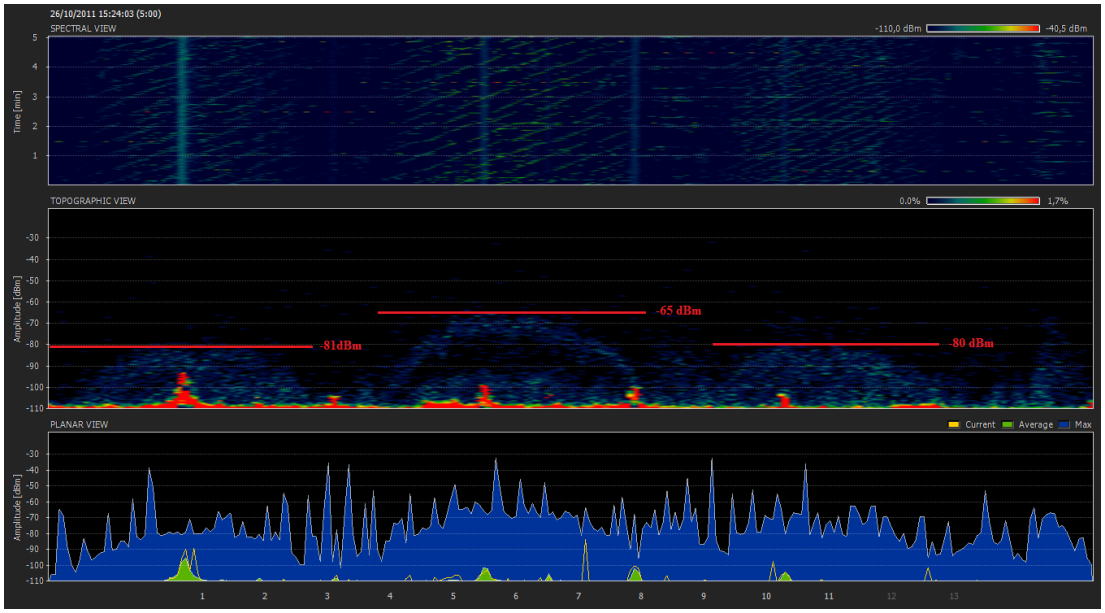

Figure 6.6: Spectrum Measured nearby Sensor A8

Each time the notebook PC is located next to the 802.15.4 node like shown in Fig.6.7 and at least 5 minutes long logs are collected.

The measurement results gathered in the Table 6.1 are interesting because each time

Figure 6.7: Spectrum Analyzer Measurement nearby Sensor 88

the RSS value measured by the wireless network card is better than the spectrum peak obtained by the spectrum analyzer. In this case, the spectrum peak can be thought as the lower bound for the RSS, if we assume that wireless adapter of the notebook PC has a typical receiver which can represent a wide range of 802.11 compatible device. The positions of nodes are shown in Fig.6.2.

Table 6.1: 802.11 Spectrum Peak and RSS in Channel 1

Node	Spectrum Peak	RSS	Difference
B1	-62 dBm	-56 dBm	6 dB
A8	-81 dBm	-75 dBm	6 dB
A0	-80 dBm	-80 dBm	0 dB
99	-85 dBm	-80 dBm	5 dB
91	-75 dBm	-72 dBm	3 dB
88	-65 dBm	-58 dBm	7 dB

We will assume that spectrum peak, E_{peak}, measured by the spectrum analyzer is a good representation for the worst case scenario of RSS. Now the question is how to correlate the ED measurements with E_{peak}. In the ideal case energy values measured by the 802.15.4 nodes should be proportional to the spectrum shape of 802.11b as represented in Fig.6.8.

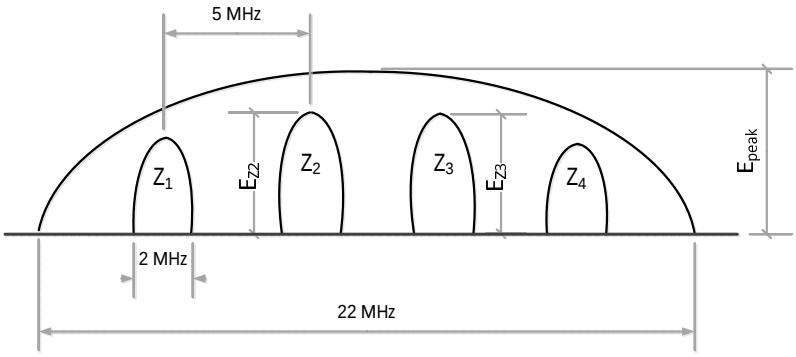

Figure 6.8: 802.11 and 802.15.4 Channel Spacing

In the figure, four 802.15.4 channels that resides in the 802.11 channel are named as Z_1, Z_2, Z_3, Z_4 in order to generalize the scenario independent to the chosen 802.11 channel. Also energy levels measured in these channels are named as E_{Z_1}, E_{Z_2}, E_{Z_3}, E_{Z_4}. Since four 802.15.4 channels are perfectly symmetric located inside the 802.11 channel, in this case E_{Z_1} will be equal to E_{Z_4} and E_{Z_2} will be equal to E_{Z_3}.

For the purpose of deriving an empiric formula the average of E_{Z_2} and E_{Z_3} measured by the nodes in each measurement point, denoted as $E_{Z_{avg}}$, matched with the E_{peak} measured by the spectrum analyzer. A linear relationship is observed and a linear fit is found as shown in Fig. 6.9.

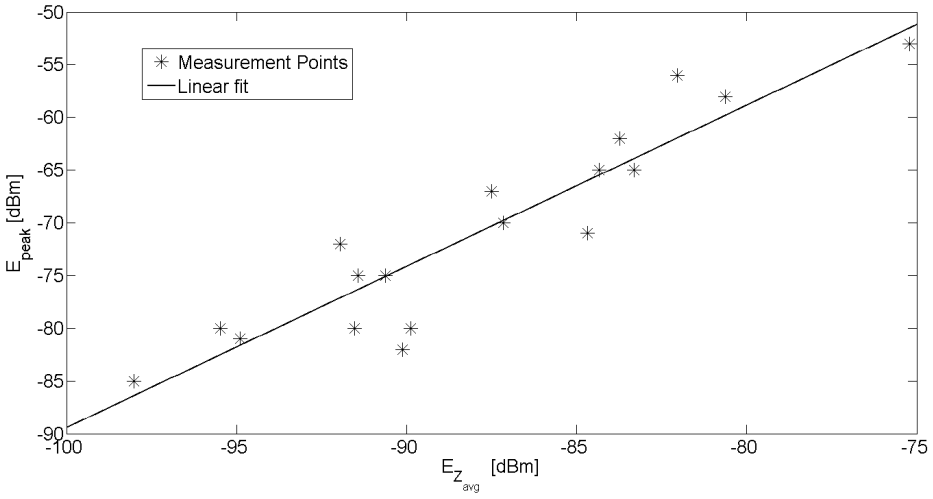

Figure 6.9: Measurements

By using the the linear fit the empiric formula obtained as:

$$\hat{E_{peak}} = E_{Z_{avg}}/2 - 86 \text{ dBm}$$

Therefore RSS should be:

$$RSS_{wifi} \geq \hat{E_{peak}}$$

Lastly in this section there are example measurements from 26 October 2011 to 18 November 2011. A sensor in the middle of an aisle and another one at the end of the same aisle is compared. The one in the middle, Sensor 99, clearly not receives enough signal power (Fig.6.10) on the channel 1 as being far away from the APs operating on channel 1.

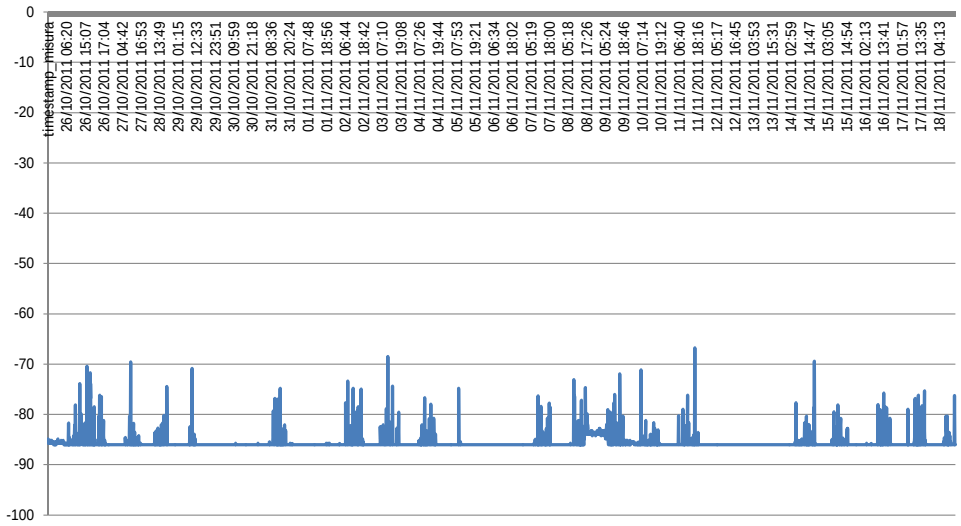

Figure 6.10: Sensor 99 - Channel 1

But because of the mobile operators in channel 1 that are strolling around there is an oscillation during the working hours. Traces of weekends and the holidays clearly seen in the figure. For instance 1st of November is a holiday in Italy. On the other hand the Sensor 88 (Fig.6.11) which is on the line of sight of AP5 working on channel 1

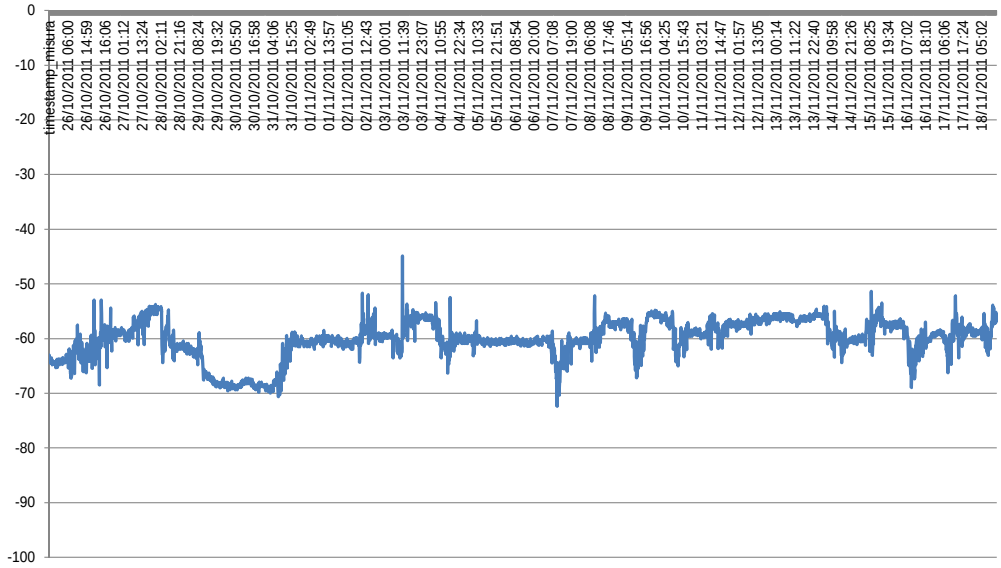

Figure 6.11: Sensor 88 - Channel 1

does not seem to be influenced much from these daily changes because of the working operators in the warehouse.

Part II

ZigBee

Chapter 7

Overview

ZigBee is a short-range wireless technology intended for applications with low cost, low power, and low data rate [36]. ZigBee is built on the top of IEEE 802.15.4 Standard. ZigBee Alliance, which is a non-profit association, defines the NWK layer and the APL over PHY, and MAC layers of the 802.15.4. In such a way, interoperability among devices produced by different manufacturers, implementing the ZigBee protocol stack, is guaranteed. IEEE 802.15.4 standard can be thought as a superset of the ZigBee specifications therefore any ZigBee application is, in fact, an 802.15.4 application. Probably the most significant contribution of ZigBee is giving mesh network capabilities to 802.15.4 applications. Mesh networking allows self-forming and self-healing through the reconfiguration of blocked route paths, with the aim of forming new routes from a node to another until the data reaches the destination. The ZigBee stack architecture is composed of a set of blocks that are called layers. Each layer performs a specific set of services for the layer above. The ZigBee stack architecture is given in detail in Fig. 7.1.

7.1 ZigBee Basics

IEEE 802.15.4 [93] defines two different operational modes, namely beacon-enabled and non beacon-enabled modes, which correspond to two different channel access mechanisms. Even though beacon enabled mode of operation is defined in ZigBee specifications, beacon-enable mode has never been the preferred way in the today's stack implementations. For instance Freescale MAC stack [61] does not even support beacon-

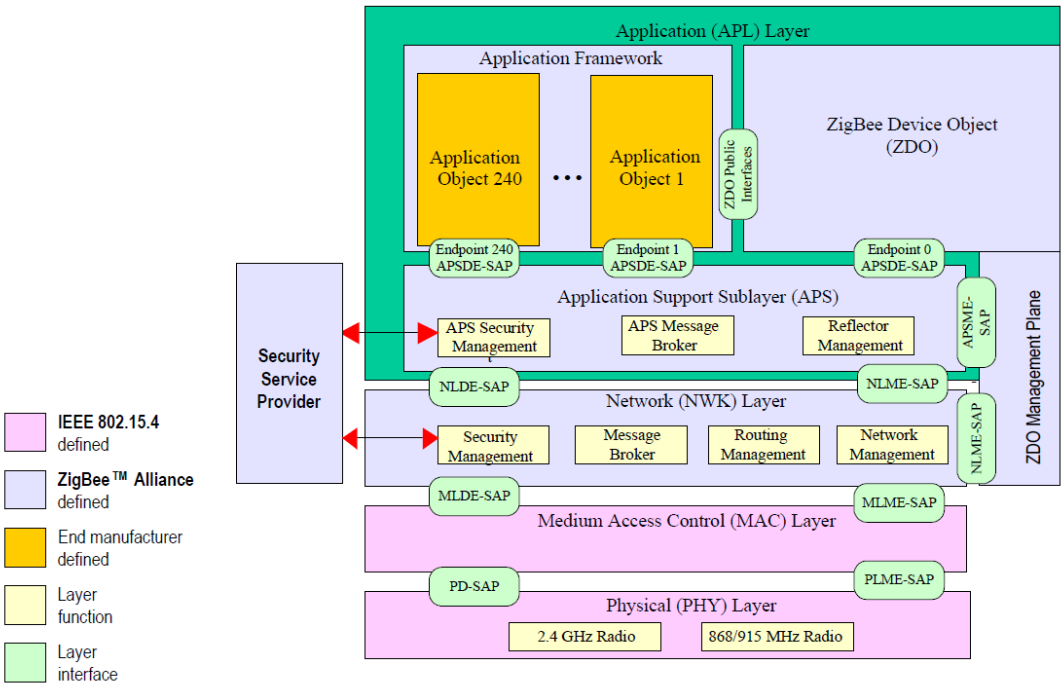

Figure 7.1: ZigBee Stack Architecture

enabled mode while TIMAC [23] from Texas Instruments is supporting it on the MAC layer, but it does not use it on the ZigBee implementation [25]. To benefit from the mesh capabilities of ZigBee, non beacon-enable mode is used. In non beacon-enabled mode, nodes wishing to transfer data use unslotted CSMA-CA described in Section 2.4.2. An acknowledgement mechanism is performed by each node: after the transmission of the packet, node waits for the acknowledgement and if it does not receive it, up to three retransmissions can be performed.

The ZigBee 1.0 specification is first appeared in December 2004 and it is referred as ZigBee 2004. In December 2006, the ZigBee 2006 specification was released, which was followed in October 2007 by the ZigBee 2007 and ZigBee PRO specifications. ZigBee 2007 and ZigBee PRO contain new features, such as group addressing, message fragmentation and interference detection/avoidance by frequency agility. Moreover, the PRO version allows centralized data collection (many-to-one routing), high-security mode, and network scalability in order to support thousands of nodes. A comparison of the different stacks is given in Table 7.1

Table 7.1: ZigBee Stack Comparison

Functionality	2004	2006	2007	PRO
Frequency Agility			✓	✓
Stochastic Addressing				✓
Group Addressing		✓	✓	✓
Many-to-One				✓
Source Routing				✓
High Security Mode				✓
Fragmentation of Packets			✓	✓
Standardized Commissioning		✓	✓	✓
Improved Reliability in Mesh				✓
Cluster Library Support		✓	✓	✓

7.1.1 Device Types and Roles

The IEEE 802.15.4 standard defines two types of devices: FFD and RFD. The FFD contains the complete set of MAC services and can operate either as coordinator, or as simple network device. The RFD contains a reduced set of MAC services and can operate only as a simple device. RFDs are intended to be used in very simple roles in the network and they can simply interact with only one FFD to which they are associated, whereas a FFD can perform routing functionalities and communicate with its child RFDs or other FFDs. The PAN coordinator should always be a FFD. With the NWK layer in ZigBee any device in the network can communicate with each other even though there is no direct radio link. ZigBee supports three types of devices: ZigBee Router (ZR), which is able to perform all the tasks described in 802.15.4, including routing; ZC, which is the principal controller of the PAN, and ZigBee End Device (ZED), which is not able to route the packets coming from other devices. In the ZigBee mesh topology more than one path can connect each couple of devices. In case of link failures or environment changes, the source device can find an alternative path on demand. The ZC is responsible for forming the network and it collaborates with the ZRs for discovering and maintaining the routes [36]. ZigBee uses slightly different terminology in contrast to IEEE 802.15.4; ZC is the corresponding PAN coordinator in 802.15.4 while ZRs are the devices that are referred as coordinators in 802.15.4. Other devices that are neither coordinators nor routers are named as ZED. The corresponding name used in 802.15.4 for ZED, is simple

device. Table 7.2 summarizes these relationships.

Table 7.2: Type of Devices in ZigBee

ZigBee Device	Corresponding 802.15.4 Device
ZigBee Coordinator	PAN Coordinator (FFD)
ZigBee Router	Coordinator (FFD)
ZigBee End Device	Simple Device (RFD or FFD)

7.1.2 Topologies

IEEE 802.15.4 networks may be organized in two different topologies: star and P2P. In the first case, the star is formed around a FFD, acting as PAN coordinator, which is the only node allowed to establish links with more than one device. In P2P topology, instead, each device is able to form multiple links to reach other devices and redundant paths between two nodes can be available. Star topology is preferable in case the coverage area is small and low latency is required by the application. P2P topology is preferable in case a large area should be covered and latency is not a critical issue. P2P topology allows the formation of more complex networks and permits any FFD to communicate with any other FFD beyond its transmission range via multiple hops. IEEE 802.15.4 can also support other network topologies, such as mesh and tree-based. These topologies are not described in the IEEE 802.15.4 standard, but they are described in the ZigBee specifications.

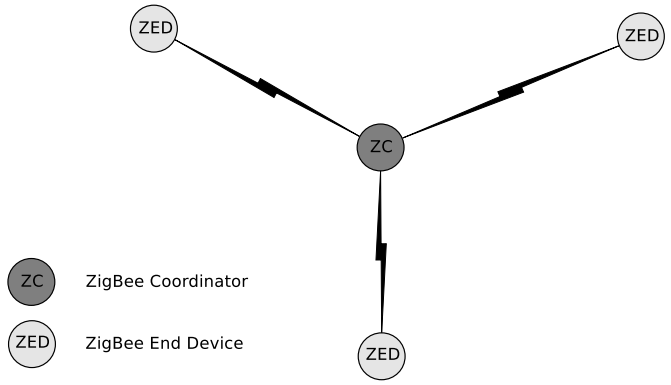

Figure 7.2: A Star Topology in ZigBee

A simple network in star topology can be established with a ZC and a various number

of ZEDs as shown in Fig.7.2. Having ZRs in between ZC and ZEDs, a tree topology as shown in Fig.7.3, can be obtained. In tree topology, ZRs form the branches and relay the messages. ZEDs act as leaves of the tree and they do not participate to the routing. By using tree topologies in ZigBee, coverage of the network can be improved and a special kind of hierarchy among the devices can be defined. Fig.7.3 shows a tree-based topology, where ZRs forward the data coming from ZEDs toward ZC. In this topology, only one path between two devices exists, parents cannot talk with each other.

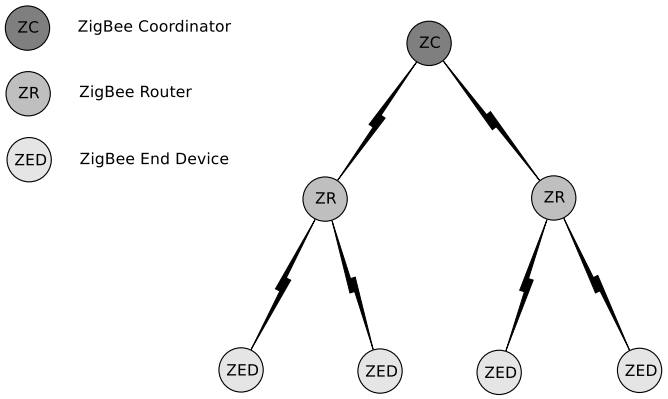

Figure 7.3: A Tree Topology in ZigBee

Furthermore, allowing P2P communication between ZRs in the topology results in a mesh network which provides alternative paths for the communication. In case of a failure in one of these alternative paths message still can reach the destination. The message hops from one node to another until it reaches to the destination. However such multi hopping comes at the expense of potential high message latency and memory necessity to store the routes. In Fig. 7.4 an example mesh network can be seen.

7.1.3 Self-Forming and Self-Healing

ZigBee networks are considered self-forming networks because the first FFD device that is turned on can establish itself as the ZC, then the other devices can join the network by sending association requests without any additional supervision. When the mesh network is formed, generally there is more than one way to relay a message from one device to another, the most optimized way is selected on demand by sending route requests. However, if one of the ZR stops working because of running out of battery or

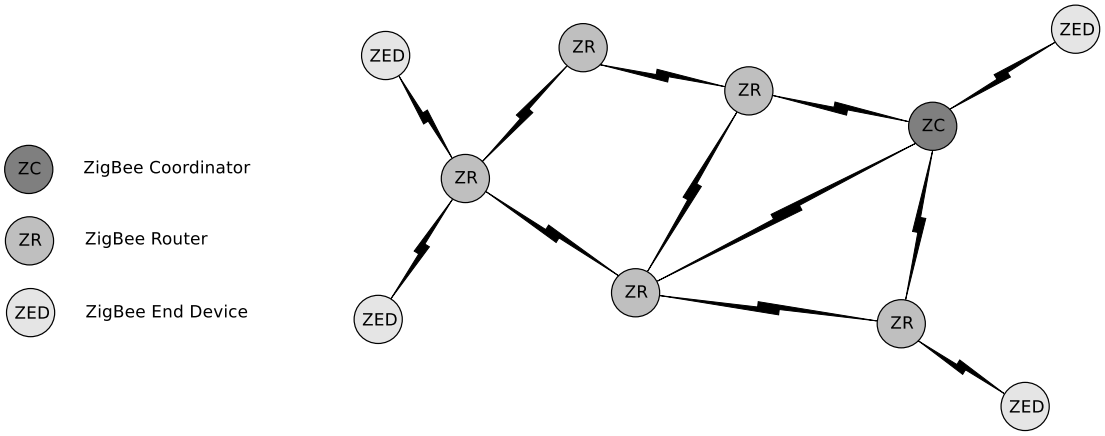

Figure 7.4: A Mesh Topology in ZigBee

because of an obstacle on the route, the network can select another route. This can be considered as the self-healing in ZigBee mesh networking.

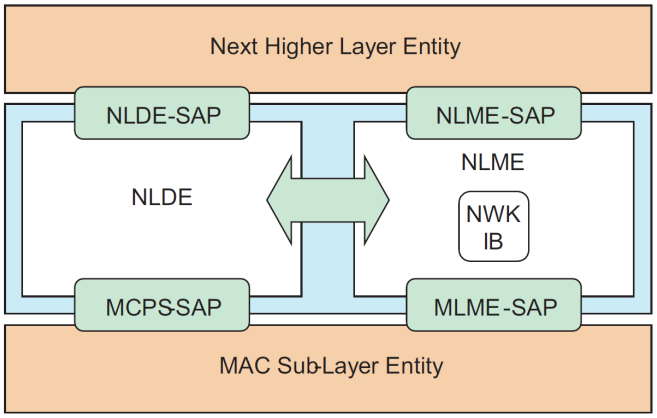

Figure 7.5: ZigBee NWK Layer

7.2 Network Layer

The NWK layer stands between the MAC layer and the APL. It is responsible for the correct operation of the MAC layer and it provides services to the APL mainly on the network formation and routing. To fulfill these tasks, there are two service entities, namely Network Layer Data Entity (NLDE) and Network Layer Management Entity (NLME), as shown in Fig. 7.5. The former one is responsible for the data transmission while the latter one handles the network duties.

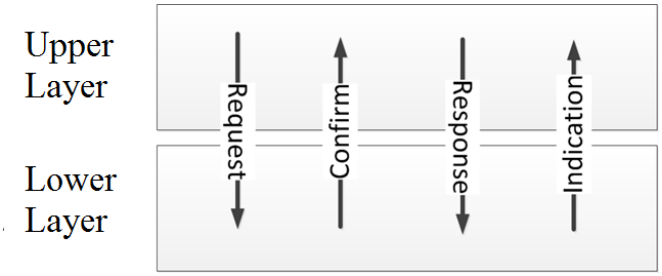

Figure 7.6: Types of Primitives in ZigBee

Specifically, NLDE provides several different services such as: generation of the Network Protocol Data Unit (NPDU), topology-specific routing, security, whereas NLME is used by the APL in order to configure a new device, start/join/rejoin/leave a network, assign addresses, discover the neighbors, discover the routes, control the reception, manage the routing.

Table 7.3: NWK Layer Primitives

Entity	Primitive	Request	Indication	Response	Confirm
NLDE	DATA	✓			✓
NLME	NETWORK-DISCOVERY	✓			✓
NLME	NETWORK-FORMATION	✓			✓
NLME	PERMIT-JOINING	✓			✓
NLME	START-ROUTER	✓			✓
NLME	ED-SCAN	✓			✓
NLME	JOIN	✓	✓		✓
NLME	DIRECT-JOIN	✓			✓
NLME	LEAVE	✓	✓		✓
NLME	RESET	✓			✓
NLME	SYNC	✓			✓
NLME	SYNC-LOSS		✓		
NLME	GET	✓			✓
NLME	SET	✓			✓
NLME	NWK-STATUS		✓		
NLME	ROUTE-DISCOVERY	✓			✓

There are Service Access Points (SAPs) to the direction of both MAC and APL to handle incoming and outgoing primitives as shown in Fig. 7.5. Not only in the NWK

layer but generally speaking in ZigBee all SAPs accept or send four different types of primitives as shown in Fig. 7.6. Commands or data to the lower layer are send by passing requests through SAP handler functions, then lower layer replies these requests by confirms. The messages from the lower layer comes through the indications and upper layer replies them with responses.

The list of all primitives in the NWK layer can be found in Table 7.3. Primitives such as DATA, NETWORK-DISCOVERY, NETWORK-FORMATION, work asynchronously. The upper layer sends the request and the confirm comes when the task is realized. On the other hand, GET and SET primitives work synchronously so the confirm comes immediately after, since these commands just manipulate the NWK Information Base (NIB) attributes rather than performing some tasks. There are also indications such as JOIN, LEAVE, SYNC-LOSS, NWK-STATUS, which do not require any response.

7.2.1 Addressing and Communication Mechanisms

In IEEE 802.15.4, there are two types of addresses, namely short and extended. Extended address is a 64-bit long unique address given to the physical radio, while short address is a 16-bit long address that is obtained through the association to a particular network. In IEEE 802.15.4 both extended and short addresses can be used for the communication. By using short addresses the length of the messages can be reduced. On the other hand, only 16-bit addresses are allowed in ZigBee for the communication inside the network. This 16-bit long address is called NWK address. ZigBee specifications also refer to short address in the MAC layer as the MAC address but it requires NWK address to be equal to MAC address. Therefore NWK and MAC addresses in ZigBee represent the same thing but in different layers, which is unique only inside the network. The notation 'extended IEEE address' for the 'extended address' in 802.15.4 is used to identify the physical radios. Table 7.4 shows the notation. Each radio in a network can have a single IEEE address and a single NWK address.

Table 7.4: Address Types

ZigBee	**802.15.4**
NWK (MAC) address	short address
extended IEEE address	extended address

Since ZigBee uses 16-bit long network addresses in the Network layer, this gives a range of unique addresses from 0x0000 to 0xffff but the range from 0xfff8 to 0xffff is reserved for broadcast messages, therefore in theory a ZigBee network can be formed up to 65528 devices including ZC. The ZigBee network is initialized by ZC which always takes the first address 0x0000. For the other devices that will join the network, distributed and stochastic address assignments are the two specified mechanisms.

Distributed Address Assignment: A ZC or a ZR can act as a parent device, accepting associations from the other devices. A device connected to this parent device is denoted as the child device. Given values for the maximum number of children a parent may have, C_m, the maximum depth in the network, L_m, and the maximum number of routers a parent may have as children, R_m, the address blocks assigned to a router-capable child device can be computed by using the $Cskip(d)$ function, where, d represents the depth in the network:

$$Cskip(d) = \begin{cases} 1 + C_m(L_m - d - 1), & \text{if} R_m = 1 \\ \frac{1+c_m-R_m-C_m R_m^{L_m-d-1}}{1-R_m}, & otherwise \end{cases}$$

Therefore, address assignments to the router-capable child devices are done by using the $Cskip(d)$ as an offset to reserve a address sub-space for the children. For the child end devices, anyway, there is no need for an assignment of a sub-block of address space. Thus n^{th} address assignment A_n is given by the following equation:

$$A_n = A_{parent} + C_{skip}(d)R_m + n$$

where, A_{parent} is the address of the parent device.

Stochastic Address Assignment: In stochastic address allocation mode, the device itself or its parent picks up a random address. Therefore the entire address space is available for all nodes in contrast to distributed address assignment which leads to the possibility of address exhaustion, down the long branches of the tree structure. The device can keep its randomly assigned address as long as there is no address conflict with any other device of the network. In an address conflict, the device picks up a new random address.

Communication mechanisms: In ZigBee, communication mechanisms can be divided into three general categories: broadcast, multicast, and unicast. Broadcast

messages are intended to reach any device that receives the message. However still some groups respect to the types of device can be addressed as shown in Table 7.5.

Table 7.5: Broadcast Addresses

Broadcast Address	Destination Group
0xffff	All devices in PAN
0xfffe	Reserved
0xfffd	Non Sleeping Devices
0xfffc	All Routers and the Coordinator
0xfffb	Low Power Routers
0xfff8 - 0xfffa	Reserved

Multicasting mechanism delivers the message to a specific group of devices that can be flexibly defined by a 16-bit group ID. Unicast is used when the message is intended for a single device. Unless otherwise specified, in ZigBee unicast is the default mode of communication.

7.2.2 Routing

In ZigBee, the default routing protocol used by the NWK layer is Ad-Hoc On-Demand Distance Vector (AODV). Nonetheless for the requirements of different applications three other routing approaches, namely hierarchical, Many-to-One (MTO), and source routing, are also made available by the ZigBee specifications.

7.2.2.1 Hierarchical (Tree) Routing

In this way of routing, first of all network addresses should be distributed in a hierarchical manner by setting nwkAddrAlloc attribute in NIB to zero. The $Cskip(d)$ function is used to distribute the addresses as described in Section 7.2.1. Moreover nwkUseTreeRouting attribute should be set to one in order to benefit from the hierarchically distributed addresses. Simply, in this mode, since nodes are aware of their parent and child device addresses, they relay the messages according to the tree hierarchy without keeping a routing table. Main drawbacks of this type of routing are; it does not allow communication between the peer devices and address space can exhaust rapidly if there are long branches in the tree.

7.2.2.2 AODV Algorithm

In AODV route is established with Route Request (RREQ) and Route Reply (RREP) between two routing capable nodes. The source node broadcasts a RREQ packet and intermediate nodes, immediately after receiving, rebroadcast RREQ to distribute it throughout the network. Once the destination node receives the RREQ, it chooses the path with the lowest cost and sends the RREP. ZEDs cannot perform route discovery, ZC or ZRs perform the route discovery on behalf of the ZED.

In the ZigBee standard the link cost is a function of the probability of successful packet delivery, which is calculated as:

$$C\{l\} = min\{7, round(P_l^{-4})\}, \tag{7.1}$$

where P_l is the probability of packet delivery on the link l. To find the optimal route for a destination, each path is associated with a path cost, which is the sum of the costs of each single link that belongs to the path: $C\{P\} = \sum_i C\{l_i\}$. The route which minimizes the path cost is the optimal route. Path costs are memorized in the route discovery table which contains also the address of the device that is requesting a route and the address of the device that relayed the request to the current device. This address is used for relaying the result of the route discovery back to the source device. To relay a message along the paths, the ZC and ZRs keep routing tables that contain the next-hop to the destinations on the way.

7.2.2.3 Many-to-One

In MTO the device (sink) that wants to establish routes to itself uses the same RREQs described in Section 7.2.2.2, but setting only the destination address field as 0x0000. This RREQ from the sink establishes an entry in the routing tables of all routing capable nodes in the network with the destination of sink. Therefore, all routers can relay packets to the sink without further sending route request messages.

7.2.2.4 Source Routing

In MTO routing nodes can be asked to send route record packets, which are relayed to the sink exactly like the other packets from the same routes, but in each hop by storing the path. Then, when the sink needs to address each node separately it can specify

the route, that is obtained by receiving the route record packet, by including the path inside the packet. This kind of routing is called source routing.

7.3 Application Layer

The ZigBee APL consists of Application Support Sublayer (APS), ZigBee Device Object (ZDO), and Application Framework (AF).

7.3.1 Application Support Sublayer

APS is the interface between the NWK layer and APL which provides services to ZDO and AF. The responsibilities of the APS include: maintaining tables for binding (defined as the ability to match two devices together based on their services and their needs), and forwarding messages between bound devices. To fulfill these tasks, there are two service entities, namely Application Support Sublayer Data Entity (APSDE) and Application Support Sublayer Management Entity (APSME), as shown in Fig. 7.7. The former one is responsible for the data transmission while the latter one handles the management duties.

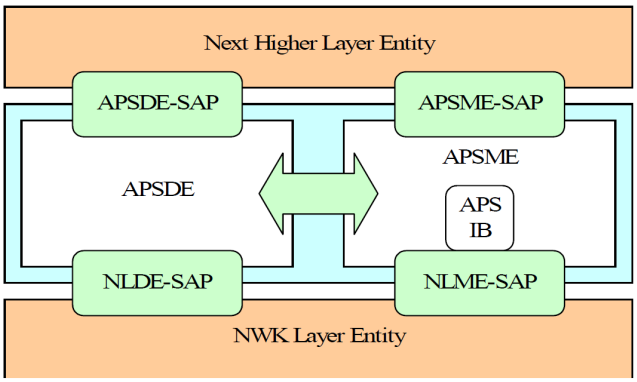

Figure 7.7: Application Support Sublayer

In specific, APSDE provides services such as: generation of the Application Support Sublayer Data Unit (APDU), binding, group address filtering, reliable transport, duplicate rejection, packet fragmentation, whereas APSME, instead, responsible for binding management, Application Support Layer Information Base (AIB) management, security, group management. There are SAPs to the direction of both NWK layer and the

upper layer entities such as ZDO and APS to handle incoming and outgoing primitives as shown in Fig. 7.7. The list of all primitives in APS layer can be found in Table 7.6. DATA primitive works asynchronously; the upper layer sends the request and the confirm comes when the task is realized. All other primitives works synchronously, therefore, confirms comes immediately after. GET and SET primitives manipulate AIB attributes, BIND and UNBIND manipulate binding table, ADD-GROUP, REMOVE-GROUP, REMOVE-ALL-GROUPS manipulate group table.

Table 7.6: APS Primitives

Entity	Primitive	Request	Indication	Response	Confirm
APSDE	DATA	✓			✓
APSME	BIND	✓			✓
APSME	UNBIND	✓			✓
APSME	GET	✓			✓
APSME	SET	✓			✓
APSME	ADD-GROUP	✓			✓
APSME	REMOVE-GROUP	✓			✓
APSME	REMOVE-ALL-GROUPS	✓			✓

Binding: Binding allows Zigbee devices to establish a designated destination endpoint on a remote device from a local endpoint in AF. In this way not only physical devices but also different applications running on these physical devices can be connected seamlessly. Only a device supporting a binding table can initiate this procedure.

Group Addressing: Groups of physical devices can be created by using group addressing table in APS. These groups of devices can then be accessed by using multicast communication described in Section 7.2.1.

7.3.2 ZigBee Device Object (ZDO)

ZDO can be described as the fundamental application that provides self-forming and self-managing capabilities to the ZigBee network. It describes the state machine functionalities of ZC, ZR, and ZED in normal operation and initialization states. Besides, ZDO describes the discovery, network, binding, security, node, and group manager objects. There is a close relationship between ZDO and ZigBee Device Profile (ZDP) because the basic building blocks of the behavior in ZDO is composed by the clusters

(services) in ZDP. ZDP defines capabilities supported by all ZigBee nodes. ZDP, like any other ZigBee profile, is a collection of clusters defining the functionality. ZDP assumes a client/server model like application profiles in AF. Each cluster provides client and server functionalities. Clusters are organized in three groups, namely device and service discovery, binding management, and network management.

7.3.2.1 Device and Service Discovery

The simplest discovery functionality provided by IEEE 802.15.4 Standard is the network discovery. In non beacon enabled mode, nodes can discover the networks on their radio range by actively asking for beacon transmissions, while in beacon enable mode they can passively listen to the periodic beacons coming from the coordinators. In addition to the network discovery provided by 802.15.4, a ZigBee node can discover (using requests) the addresses of the other nodes in the network. A node can ask either the IEEE address or the NWK address of any other device by using the address request service (cluster). ZEDs can only respond to the request with their address while ZRs and ZC can respond to the request with the list of associated devices. This gives the option to the requestor to determine the network topology underlying the queried device. A device making service discovery requests does so via a client role. A device which replies to these requests does so via a server role. Every device in the ZigBee network has the server functionality to respond to the address requests but ZigBee specifications do not impose all devices to have the client functionality.

Service discovery is realized by querying ZigBee descriptors. ZigBee descriptors provide information about the node and Application Object (AO) which resides in AF. There are five principal descriptors, namely simple, node, node power, complex, and user. Simple descriptors keep the information about the AOs, while node, power, complex, and user descriptors keep the general information about the ZigBee node itself.

- **Simple Descriptor:** Each active endpoint (endpoint that contains an AO) has a simple descriptor.
- **Node Descriptor:** Node descriptor indicates the capabilities of the ZigBee node. There should be one and only one node descriptor in each node.

- **Node Power Descriptor:** Node power descriptor provides the current power status of the node. Like the node descriptor there should be one and only one node power descriptor in each node.
- **Complex Descriptor:** Complex descriptor contains extended information about the ZigBee node. It is an optional descriptor.
- **User Descriptor:** User descriptor contains a user-friendly indication (i.e. a text describing the node) to identify the device inside the network. It is an optional descriptor.

More details on service discovery can be found in Section 7.3.4.2.

7.3.2.2 Binding Table Management

Binding table management in ZDO provides the tools to manage the use of binding table in APS, which is also denoted as source binding table. A node can bind itself to a remote node or nodes in the network can store their source binding tables in primary binding table cache located in a non sleeping routing capable device. Any node can store a backup of the source binding table in the primary binding table cache as long as it has storage space left and can recover it later if necessary.

7.3.2.3 Network Management

Network management allows nodes to retrieve management information from remote nodes. For instance, a node can request a remote node to execute an ED, active or passive scan in order to discover networks and channel conditions in the vicinity of the remote node. Similarly neighbor tables, routing tables, binding tables, primary discovery caches can be retrieved from the remote nodes. Also direct join, leave, or permit joining commands can be sent to remote nodes in order to manage the authorization of the nodes in the vicinity of the remote node.

7.3.3 Application Framework

AF is designed to host AOs that can be used by the end manufacturers in order to implement the intended functionality. AF in ZigBee uses the terms AO and ZigBee device interchangeably, like it is done in this chapter. In fact, AO can be thought

in general as the application including all the functionality that runs on the given endpoint. On the other hand, ZigBee device can be defined as the starting template for that application which defines the interfaces and commands. For instance, a light switch is a ZigBee device but a different implementation of the light switch which periodically turns on/off can be described as the application object. Since AF denotes AOs as ZigBee devices in order to avoid ambiguity from now on we will use the following terminology:

- **The ZigBee Node:** The physical device that contains the transceiver;
- **The ZigBee Device:** Application Object (AO) that resides in the Application Framework (AF) (e.g: on/off switch, temperature sensor, etc.). ZigBee devices contain clusters;
- **ZigBee Cluster:** Container of an application functionality provided by Zigbee Alliance (e.g: On/off cluster defines all the functionality to turn on/off a ZigBee device);
- **ZigBee Cluster Library (ZCL):** Collections of clusters organized with respect to the application domains;
- **The ZigBee Profile:** Collection of ZigBee devices for a specific application area (e.g: Smart Energy (SE) Profile , Home Automation (HA) Profile).

The ZigBee devices can be manufacturer specified or profile defined. The Zigbee Alliance publishes profiles to be used by third party manufacturers and assures the interoperability between different manufacturers by firmly defining the ZigBee devices in these profiles. ZigBee devices contain clusters from the ZCL. These clusters always define the server and client-side of the functionality. By using the service discovery functionality provided by ZDO, capabilities of the ZigBee node, ZigBee device/s and the server/client clusters inside the ZigBee device/s can be discovered.

AOs use APSDE-SAP to send and receive data between peer AOs. A peer can be created by binding AOs in different ZigBee nodes. For example an AO in a ZigBee node can be programmed to function as a light switch whereas another AO in another ZigBee node can be programmed as a light actuator to turn on/off a light. By the binding

process, these AOs in different nodes can be peered. These peered AOs in different ZigBee radios then can start interacting.

Inside the AF up to 240 distinct ZigBee devices can be defined by interfacing them to the endpoints indexed from 1 to 240. Endpoints out of this range are reserved for:

- **Endpoint 0:** The data interface to ZDO.
- **Endpoints 241-254:** Reserved.
- **Endpoint 255:** The data interface to broadcast data to all application objects.

7.3.4 ZigBee Cluster Library

Application functionalities of ZigBee are arranged in the clusters defined by ZigBee Alliance. For instance, to turn on/off a device all the necessary functionality can be found in the On/Off Cluster. ZigBee Alliance provides these clusters in a library, namely ZCL [35]. ZCL allows common clusters to be re-used across different applications and allows multiple Original Equipment Manufacturer (OEM) vendors to develop interoperable products. ZCL is organized in different functional domains: General, Measurement and Sensing, Lighting, Heating, Ventilation, and Air Conditioning (HVAC), Closures, Security and Safety, and Protocol Interfaces. For instance "on/off light switch" and "on/off light" devices are defined in "on/off" cluster in the "lighting devices" domain. These "on/off light switch" and "on/off light" devices can be in different ZigBee radios but a logical link can always be created between these two devices by binding them. A device might not be a physical entity, but it can be seen as a component that provides different functionalities to a ZigBee node. As an example, the range extender device is not a separated physical entity within a ZigBee node, but it is a functionality that gives the node it belongs to range extending capabilities.

7.3.4.1 Client / Server Model

A client/server model is used in the cluster definitions. Every cluster definition has a server and client interface. Typically, these interfaces stand in different nodes, where the server entity stores the attributes of the cluster and the client entity affects or manipulates those attributes by sending commands to the server entity. ZigBee profiles do not define a client or server device, simply both are referred to as devices. However

ZigBee devices can be classified as server or client devices since they generally contain only client or only server clusters. Fig. 7.8 represents the client/server model.

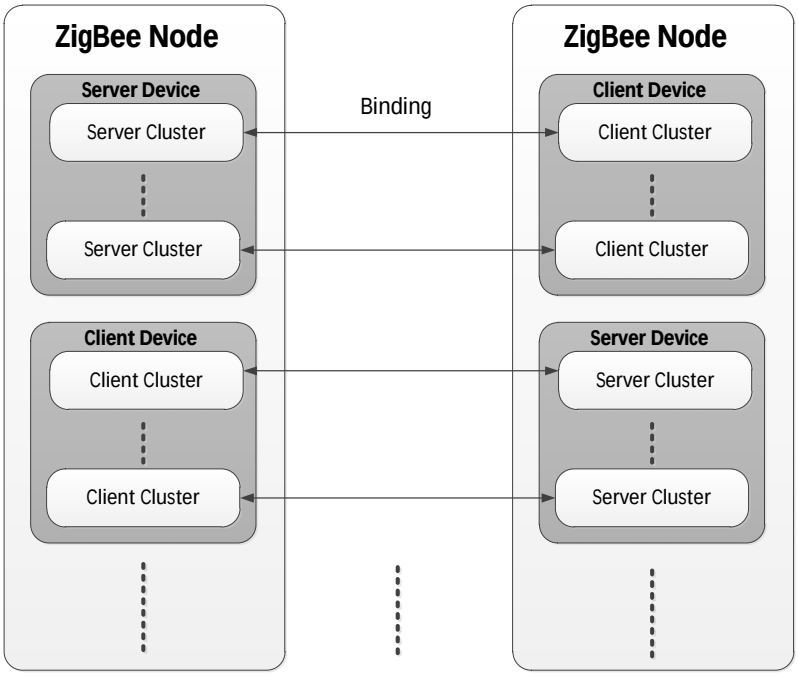

Figure 7.8: Simplified Client/Server Model

The corresponding clusters can be matched by the binding process but first of all discoveries of client and server clusters in different ZigBee nodes should be performed. In the binding process, two devices can only be matched if both devices are in the same cluster, but one is an input (server) device and the other is an output (client) device. In order to find out possible matching inside the ZigBee network, service discovery can be accomplished by issuing an active endpoint query on a given device or by using a match simple descriptor query. This kind of discovery is called service discovery. In order to perform service discovery formerly the wireless links between ZigBee nodes should be established. The next section gives details about service discovery which has been briefly described in Sec. 7.3.2.1.

7.3.4.2 Service Discovery

Service discovery gives the ability to a ZigBee node to determine services offered by the other nodes inside the network. Services can be network related, like security (trust

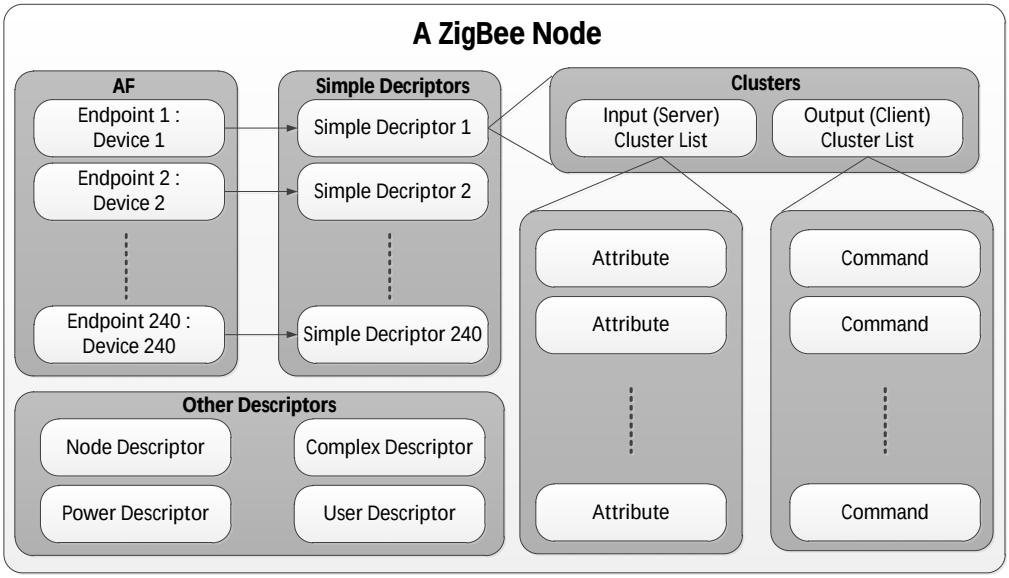

Figure 7.9: Simplified Relationship between ZigBee Devices and Descriptors

center), relaying capabilities of the node, etc., or they can be appliance related, like the application functionality (ZigBee device/s that the node contains) of the node in the ZigBee network. In the AF every application has an unique endpoint (i.e. active endpoint) and for each active endpoint there is a descriptor called simple descriptor. Simple descriptor contains attributes and commands from ZCL. Active endpoint query permits an enquiring device to determine the active endpoints of a remote node. After that, simple descriptor queries can be sent to these active endpoints to obtain the simple descriptors which contain the information about that active endpoint. Also a ZigBee node can send match simple descriptor query to remote nodes with the purpose of finding a match itself. The response includes the peer-able clusters, if there is. Moreover, general information about the node such as frequency band, functionality in the network, current power level, etc. can also be obtained by using queries. Service discovery messages inside the network can be unicast or broadcast addressed. In other words, discovery messages can be broadcast to all or can be sent directly to a specific node. There are also node, power, complex, and user descriptors to make available the general information about the ZigBee node. In Fig.7.9 a simplified model of the relation between descriptors and ZigBee devices can be seen. Service and address discovery

information can be cached in primary and backup discovery caches in a routing capable non sleeping node (i.e. ZC, ZR). This node may be used to store the descriptors of other nodes. A ZED, for instance, which periodically goes sleeping in order to preserve energy, can store its descriptors in the discovery caches. In the case when the device offering a particular service is not accessible at the time the discovery operation takes place, discovery caches can reply the service discovery queries on behalf of the device. ZC and ZRs can act as discovery cache since they are always on.

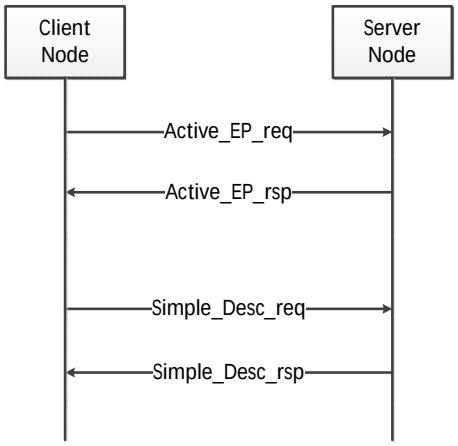

Figure 7.10: Service Discovery Example

As an example, in Fig. 7.10, a node can query the active endpoints in a remote node by sending an Active_EP_req. After receiving the query, the remote node replies with Active_EP_rsp, indicating the active endpoints. Then, the node sends Simple_Desc_req to the active endpoints to receive the descriptors that belong to those active endpoints. The node, power, complex, user descriptors can also be obtained by using the respective requests. Detailed information about the requests for the service discovery can be found in [36].

7.3.4.3 General Commands

ZCL commands to the remote ZigBee devices are sent through passing messages to APSDE-SAP that is shown in Fig.7.1. These messages are sent to the APS as data requests, as shown in Figure 7.6. The APSDE-SAP include the request, confirm, re-

sponse and indication primitives for data transfer. Bidirectional data transfer between APS and AO is mainly for sending and receiving ZCL commands.

Table 7.7: ZCL General Commands

Command ID	Description
0x00	Read Attributes
0x01	Read Attributes Response
0x02	Write Attributes
0x03	Write Attributes Undivided
0x04	Write Attributes Response
0x05	Write Attributes No Response
0x06	Configure Reporting
0x07	Configure Reporting Response
0x08	Read Reporting Configuration
0x09	Read Reporting Configuration Response
0x0a	Report Attributes
0x0b	Default Response
0x0c	Discover Attributes
0x0d	Discover Attributes Response
0x0e	Read Attributes Structured
0x0f	Write Attributes Structured
0x10	Write Attributes Structured Response
0x11-0xff	Reserved

In detail, data request primitives transfer commands between peer AOs, which are typically in different nodes. Confirm primitives report the results of the requests. Response primitives return errors, acknowledgements, or some other useful information. Finally, indication primitives notify the transfer of data to the destination. In order to manipulate or access a remote device attribute/s (e.g.: state of a light, temperature ...) general command frames shown in Table 7.7 are used: • *Read Attributes:* is generated by a device that wants to obtain the values of attributes located in a remote device, • *Read Attributes Response:* is generated in response to a read attributes command by the remote device, • *Configure Reporting:* is used to configure the reporting mechanism for the attributes on a remote device, • *Report Attributes:* is generated by the remote device to report the values of its attributes to the others.

Client clusters might contain specific commands to send orders to the corresponding server cluster. In addition to these specific commands, ZCL defines general commands that are profile independent in order to monitor or manipulate any kind of attribute. By using ZCL general command frames, client clusters can access server attributes.

7.3.5 Application Profiles

For different application domains there are several profiles (e.g. Smart Energy (SE) [104], Home Automation (HA) [103], Health Care (HC), etc.). Profiles provide ZigBee devices (e.g. on/off switch, temperature sensor, etc.) that can be implemented in endpoints. Basically, each cluster specification in ZCL defines functionality, attributes and possible commands, whereas the device definitions in the application profiles gathers the related clusters in order to form a ZigBee device which resides in an endpoint.

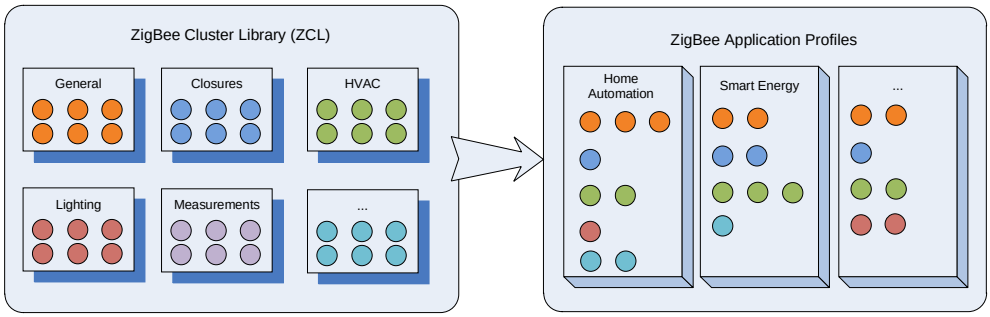

Figure 7.11: The Relationship between ZCL and Application Profiles

ZigBee Profiles define the devices according to their application domain. For instance, HA profile defines the devices like on/off switch, occupancy sensor, thermostat, etc., in order to be used in the buildings application domain. Additionally, ZigBee gives the flexibility to define custom devices by third party manufacturers. The behavior of these devices is raised by assembling manufacturer defined or ZCL clusters. For example, inside the thermostat device defined in HA profile there are implementations of thermostat, groups, and scenes clusters from ZCL. These clusters define attributes and commands in order to manipulate the device functionality or monitor the physical environment.

ZigBee devices might not need to be mapped to physical entities connected to a ZigBee node, but they can be the components that provide additional functionalities

to the ZigBee nodes. For example, the range extender device defined in both SE and HA profiles is not a separated physical entity but it is just a functionality that gives solely range extending capability to the node it belongs.

Chapter 8

Mesh Routing in a Two Dimensional Grid: Centrale Adriatica Project

In Centrale Adriatica Project, 228 ZigBee nodes and 5 sinks (ZCs) are deployed as shown in Fig. 8.1 in order to monitor the WiFi coverage in a warehouse. The details about sensing WiFi signal strength using ZigBee nodes are described in Chapter 6. Here in this chapter implementation of the ZigBee network and some measurements on the network performance is given. The deployment of the nodes is a fine planar grid, thus the measurements in this chapter have a great potential to be a reference measurement to the analytical routing models or simulations in planar sensor deployment studies. Currently in the WSN research community, there are not so many this large-scale test-beds. Some of them around 200 nodes or more can be listed as: MoteLab [101], Kansei [37], KanseiGenie [95], WISEBED [47], Senslab [19], TWIST [69], and w-iLAB.t [43] .

8.1 System Description

In the measurement system, 228 nodes are clustered into 5 sub-networks with 48 nodes in each (except the second one which has 36 nodes). The non overlapping ZigBee channels (i.e. 15, 20, 25, 26), considering the WiFi channels (i.e. 1, 6, 11) employed in the APs, are used in order to mitigate the interference (see Fig. 6.3). As shown in Fig. 8.2 each ZC in the network is connected to a SBC in order to interface them to

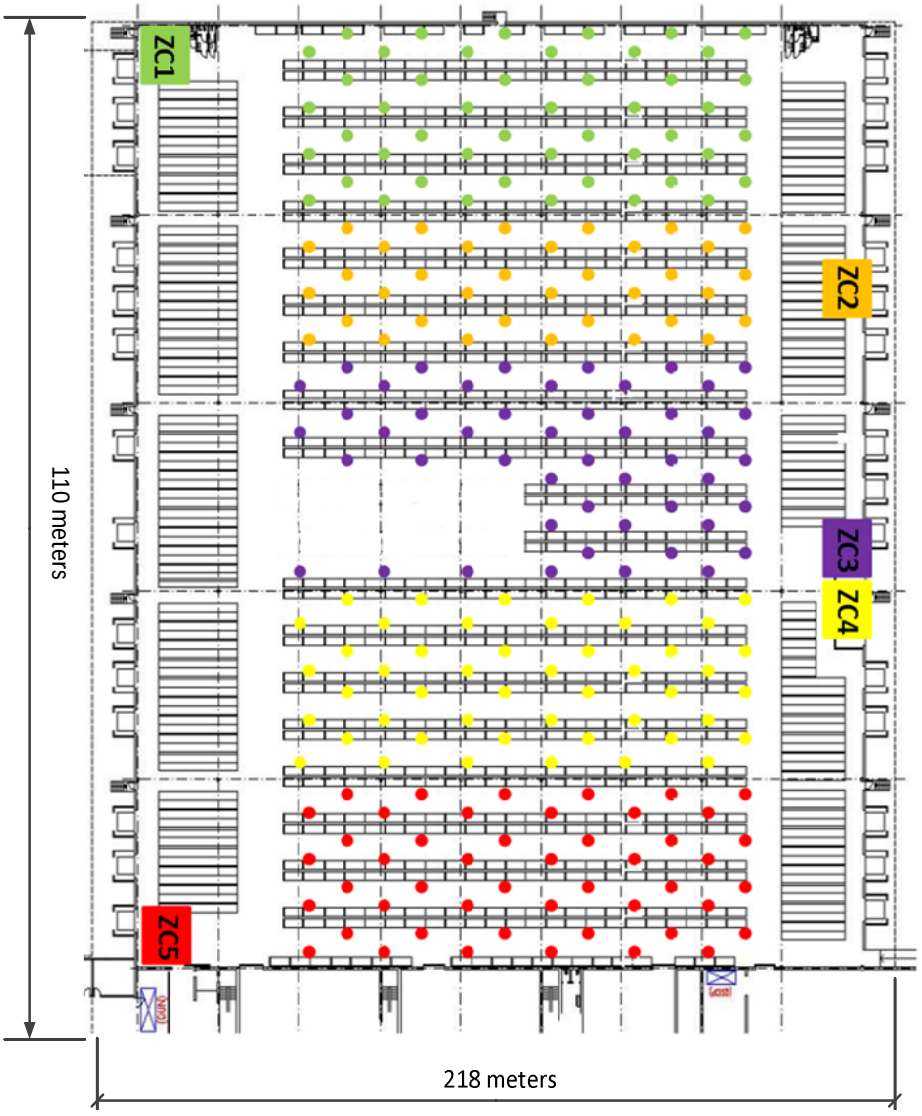

Figure 8.1: ZigBee Networks

the Ethernet network. For the ZigBee network Freescale MC1322x Platform[62] and for the SBCs ALIX 3D3 [55] are used.

In the sub-networks, ZCs are responsible for the synchronization of the nodes and nodes send their measurement results to the ZCs in a multi-hop mesh topology. These results are immediately sent to the SBCs via Universal Asynchronous Receiver/Transmitter (UART) in order to be registered in the database. The data on the database is accessible through a web user interface. SBCs are also capable of receiving commands from the web user interface in order to change the network and measurement parameters. Furthermore, SBCs are responsible for the global synchronization. By using Network Time Protocol (NTP) they periodically update their internal clocks which are used to send the synchronization messages to the nodes and to timestamp the received measurement packets from the nodes. ZC, SBC, and a node are shown in the Figures 8.3 and 8.4.

8.2 Measurement Cycle

In Centrale Adriatica Project, nodes, in normal operation mode, measure the energy levels in WiFi channels at every 10 minutes. At the rest of the time they sleep as long as possible to save energy. But as occasion may require, more frequent measurements can be done up to one minute since nodes wake up every minute to receive the update messages coming from ZCs. The update message can force the nodes to do a measurement or simply sends them to sleeping mode. Update messages contain commands and the reference time. The nodes calculate the next update message instance and wake up 2 seconds before as shown in Fig.8.5, to update their internal clocks and the other parameters.

8.2.1 Synchronization

Because of the relaxed timing requirement of the application the resolution of the local clocks of the nodes are set to one second, therefore ± 1 second synchronization error respect to the clocks of ZCs can occur. This results with the maximum time difference between two nodes in a sub-network as 2 seconds which is the reason why there are at least 2 seconds guard intervals before go sleep instance and after the wake up, update and route request instances. In order to synchronize the nodes in each sub-network a 103-byte long *Write Attributes* command (see Table 7.7) has been broadcast. This

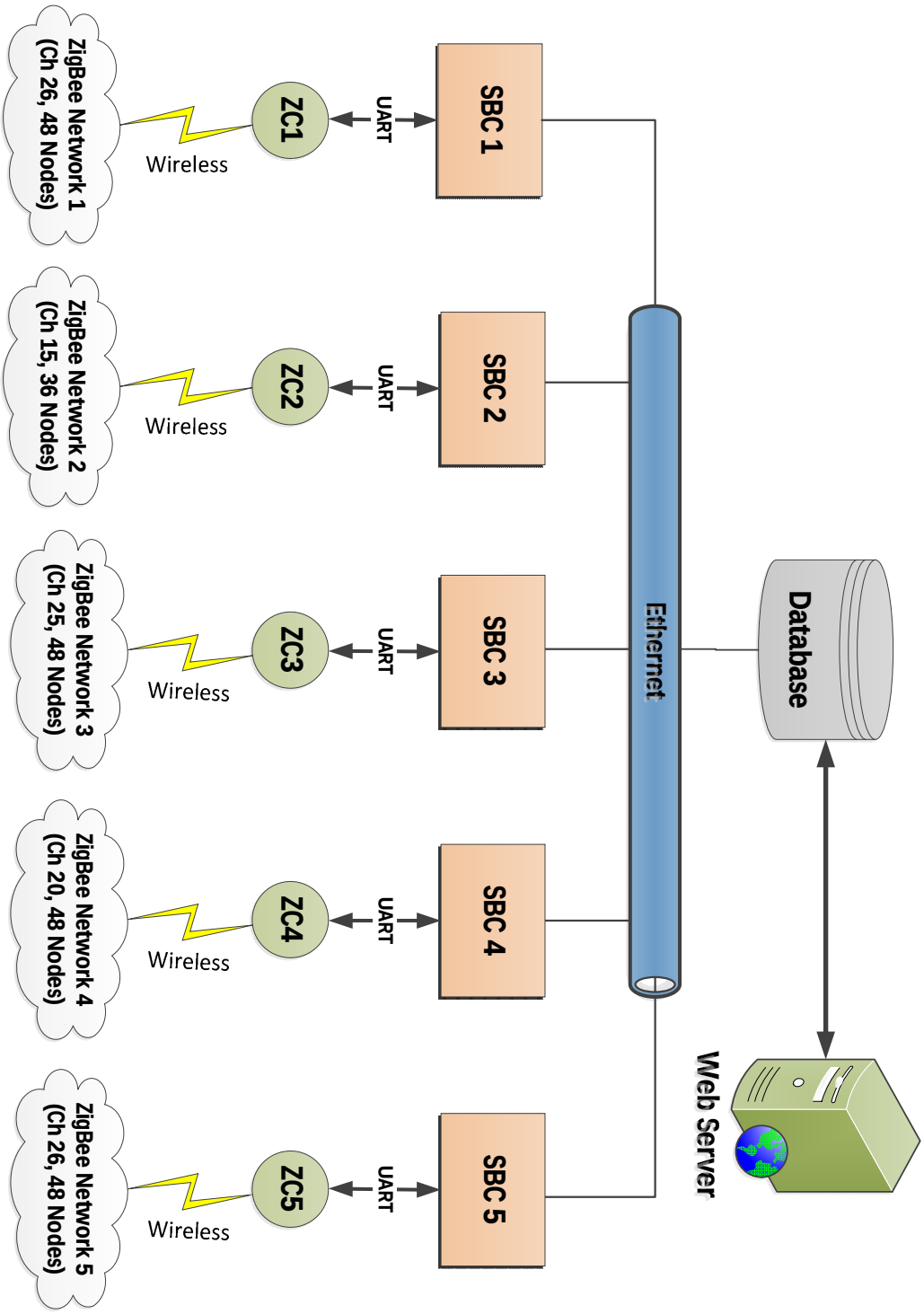

Figure 8.2: Diagram of the System

Figure 8.3: ZC and SBC

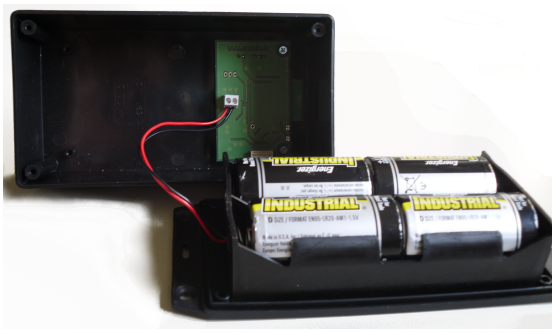

Figure 8.4: Node

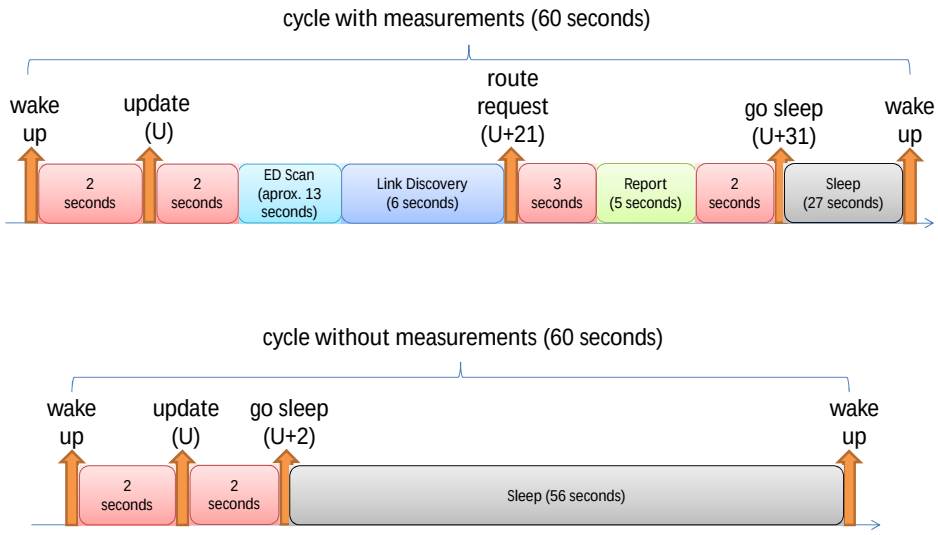

Figure 8.5: Cycle Timing

command over-writes the several attributes on the nodes (e.g. time, measurement parameters,..). Respect to the topology a broadcast message can reach to a node as a direct message from the ZC or it may be relayed by the other nodes. In order to gain some insights on how the command is delivered, the RSS value of the *Write Attributes* command is also stored in the nodes to be packed with the other measurements of the node and send to the ZC. The average RSS values of each received command frame from 26th of October 2011 to 15th of December 2011 (more than 7000 cycles) are shown in Fig. 8.6.

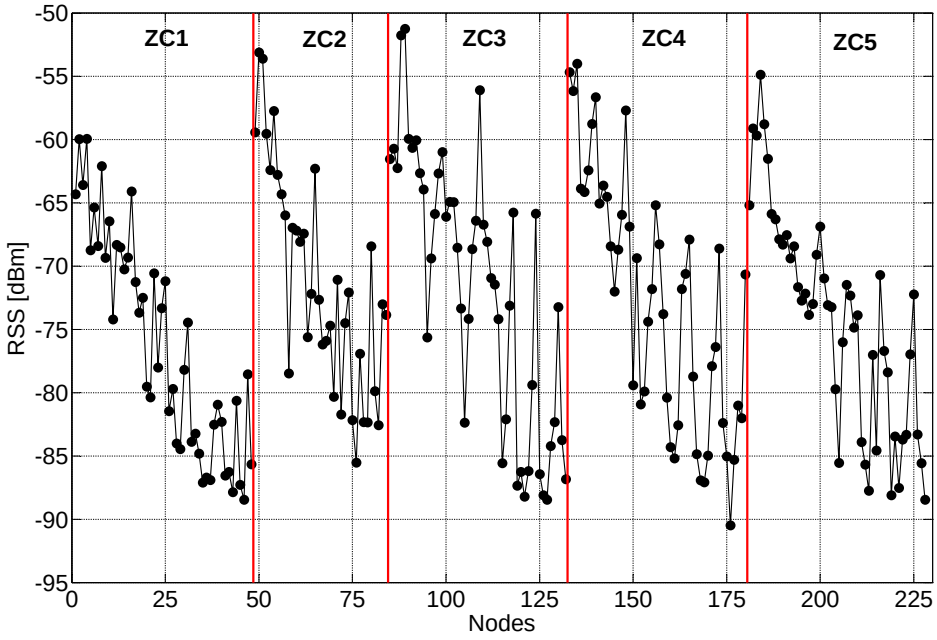

Figure 8.6: Average RSS of *Write Attributes* Command Broadcast

In ZigBee, passive acknowledgement scheme is used for broadcasts. The acknowledgement scheme operates by keeping track of the neighbors that have successfully relayed the broadcast transmission. Routers maintain Broadcast Transaction Records (BTRs) of the broadcast transactions occurring within the radio range. When a device first receives a broadcast it creates a new BTR and indicates to the application layer that a new broadcast frame has been received. But on the other hand when it receives the retransmissions of the broadcast it just keeps track of the passive acknowledgements without informing the application. Therefore the RSS value of a broadcast in the

application layer represents the first received packet but not the retransmissions. In Fig.8.6 RSS values respect to the distances to the ZCs (see Fig. 8.8) are in agreement thus, it can be stated that almost all the nodes were receiving the first transmission of the broadcast at the same time. There are Power Amplifiers (PAs) in the ZCs and the transmit power is 20dBm.

8.2.2 Link and Path Discovery

In the Section 3.2 an empirical relationship between the RSS and PER for 20-byte long packets was found as shown in Table 3.1. The ZigBee Specifications requires the link cost to be calculated as in (7.1). On the other hand the Freescale implementation simply uses 3 different cost values: 1, 3, and 7. The mapping is as shown in Fig.8.7. 0 means there is no connection. In the rest of the chapter the link cost values are the function of RSS as shown in the figure and the path cost is the sum of all link costs through a path: $C\{P\} = \sum_i C\{l_i\}$

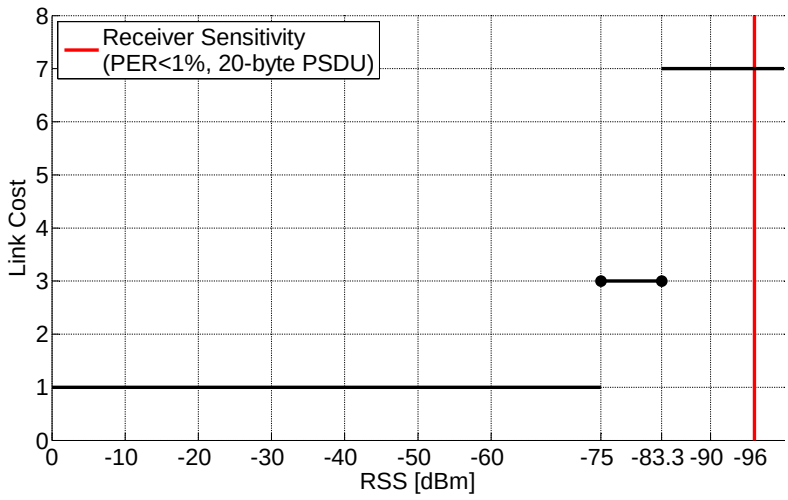

Figure 8.7: Received Signal Strength to Link Cost Map in BeeStack

In the routing of the packets from nodes to the ZCs, MTO with symmetric links is used (details on routing is found Section 7.2.2). Each time a node receives a packet it updates the incoming link cost value for the source of the packet with the calculated new link cost by using the RSS of the packet as in Fig 8.7. In this way nodes populate the

neighbor tables with the information about their neighbors. ZigBee routers periodically transmit link status messages to the one hop distant neighbors in order to indicate the entries in their neighbor tables. Therefore nodes listening to these messages can learn their outgoing link costs to the sender of the link status message. Initially outgoing cost is set to 0. At least a link status message should be received in order to obtain the outgoing cost. Since the symmetric links are used MTO route requests coming from ZCs, only consider the high-quality link among the incoming and outgoing links in order to ensure the reliable communication.

For instance, the link that has the incoming cost 7 and the outgoing cost 1 is considered as a link with the link cost of 7. Moreover in the most extreme case if one of the costs is 0 because of not receiving a packet from the corresponding node yet, result will be no connection between these two nodes even though the physical link between them is in good quality. This last example clearly shows how important to exchange link status messages between the nodes before sending a route request. As the result, in the measurement cycle there is a 6 seconds interval namely Link Discovery in order to exchange link status messages to form the links. The default periodicity of the link status messages is 15 seconds but in order to quicken the link discovery the link status message periodicity is set to 2 seconds. In theory 4 seconds should be enough to exchange link status packets in a non synchronized way at the 2 seconds link status periodicity but link discovery interval is set to 6 seconds to give more room to the nodes that has not yet been able to access the channel because of unexpected packet loses. Also worth to mention here that to transmit a link status message unslotted CSMA-CA is used but there are no retransmissions for the link status messages.

For the sake of simplicity in the implementation ZC never addresses a single node, it always broadcasts the update messages, as described in Section 8.2, in order to overwrite the parameters in all nodes together. Therefore there was no need to use the source routing (see 7.2.2.4) but route record packets are used to reveal the formed routes to reach the ZC. All the relays to the direction of ZCs that are formed in at least 1500 cycles in each sub-networks are given in Fig.8.8. The thicker lines means more relays over that link. The thickness of the lines are normalized over the most used relay.

Normally ZigBee does not store the path cost to a destination in the routing table entry, principal parameters are the next hop and the final destination but the temporary

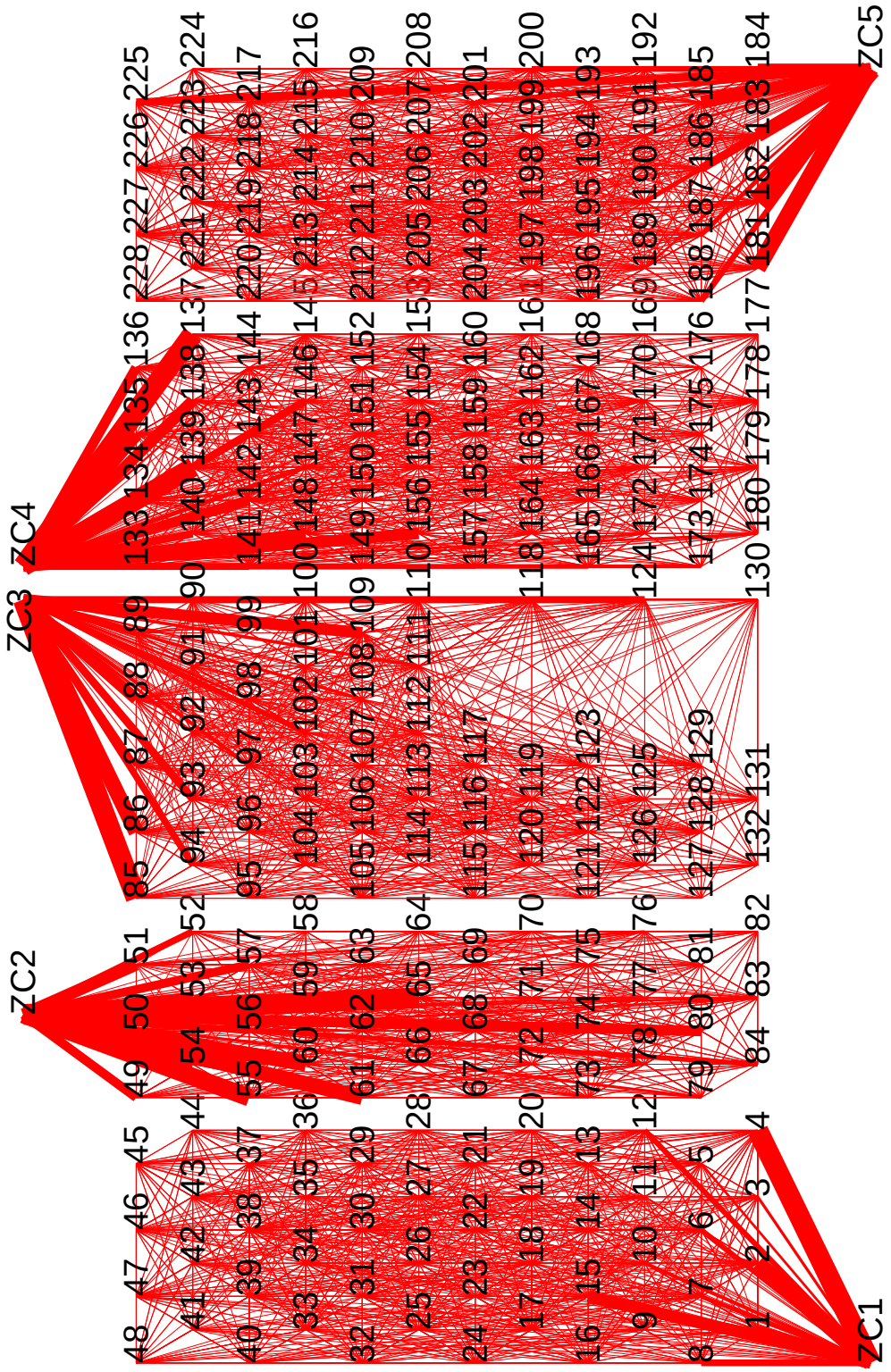

Figure 8.8: Relays

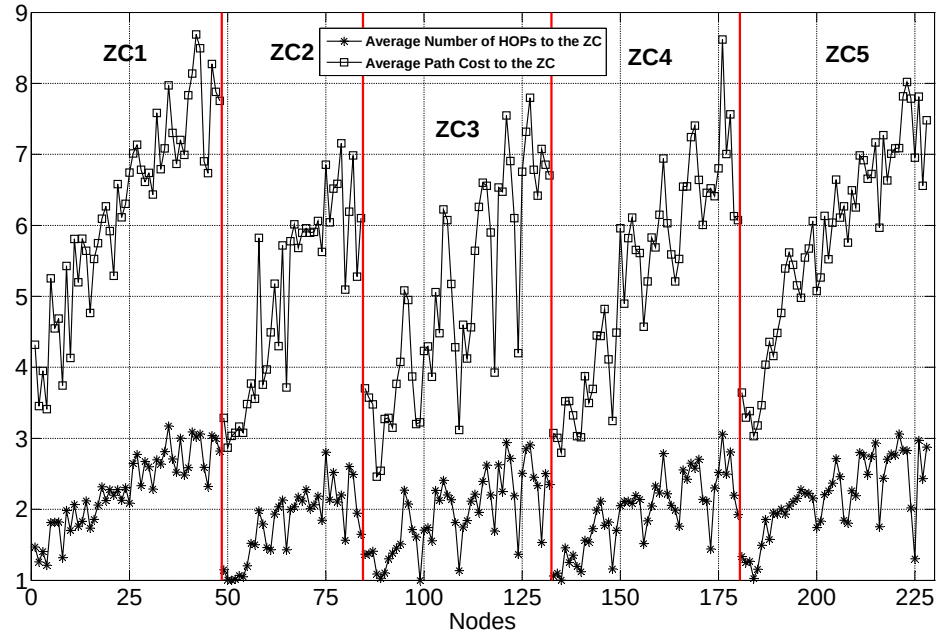

Figure 8.9: Average Path Cost and Number of Hops

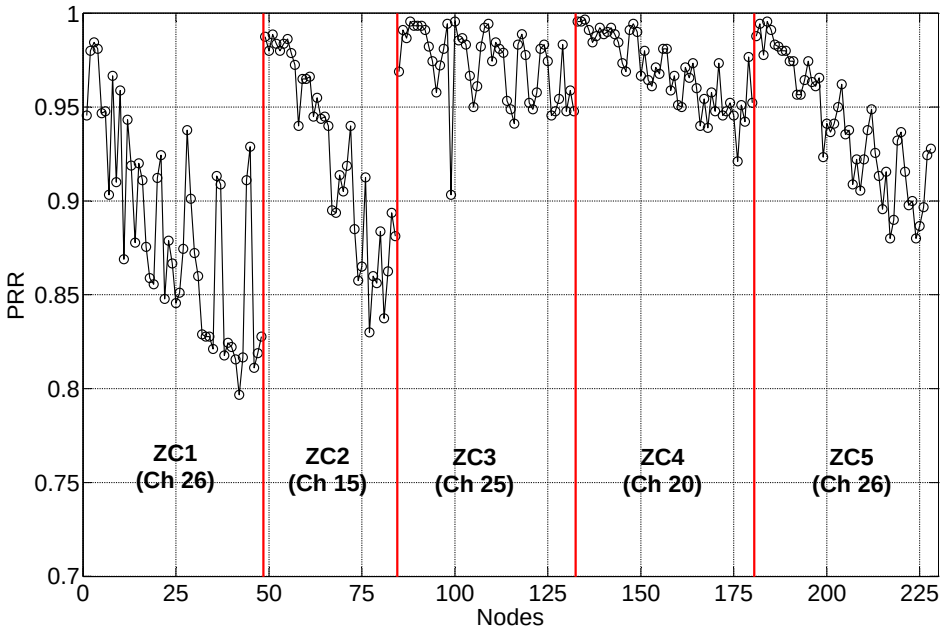

Figure 8.10: Packet Reception Rate

route discovery table keeps the path cost until it is overwritten with a new value in the next route discovery. Hence, just after the route discovery the path cost to the ZCs can be obtained by reading the route discovery table. Average number of hops and the path costs in the routes to reach the ZCs are shown in Fig.8.9.

8.2.3 Reporting

In the Report interval of the measurement cycle, all the nodes are reported their measurement results in 5 seconds by choosing a random instance. The measurements are packed into a 70-byte long *Report Attributes* ZCL command. The PRR for all nodes are shown in Fig.8.10. In almost all nodes PRR is more than 80%. The positions of nodes have a significant impact on the PRR; being away from the ZC decreases PRR. The PRR in the first network is considerably less than the others because of the fact that ZC1 had to be positioned in the only available rack cabinet which is behind a concrete wall as shown in Fig. 8.1. There was no clear line of sight.

8.3 Conclusion

In this chapter we have described the ZigBee network that is used in Centrale Adriatica Project as an example of mesh routing in ZigBee. Because of the fine grid large-scale deployment the measurements in this chapter have a great potential to be a reference measurement to the analytical routing models or simulations in planar sensor deployment studies. Actually when there are more than one node synchronously queried or many nodes in a large-scale deployment are competing for the channel access, the probability to have concurrent (parallel) transmissions is relatively high, therefore one of the first steps on the analysis of a large-scale deployments, can be to have a better understanding of the concurrent transmissions in the PHY as we discussed in Chapter 5).

8. MESH ROUTING IN A TWO DIMENSIONAL GRID: CENTRALE ADRIATICA PROJECT

Chapter 9

Query Strategies in Application Layer: eDIANA Project

In this chapter firstly, the application scenario of eDIANA Project will be presented, then the implementation of this scenario using a ZigBee network will be detailed. eDIANA application can be considered as a typical monitoring scenario since sensor nodes periodically send the information that they collect from the environment to a sink, denoted as CDC. Therefore having a typical monitoring scenario and the hardware/software implementation suggested us to measure the performance of different query strategies in the application layer. These measurement results and the conclusions are reported in Section 9.6. In Europe buildings are responsible for 40% of total energy consumption [51], which is more than the demand of industry or transportation. The total energy consumption has been rising since 1990 and the tendency shows that it will continue increasing if strong actions are not taken. The eDIANA Project [9], funded by the European Commission within FP7 through the ARTEMISIA framework, addresses the need of energy efficiency in buildings through innovative solutions based on networked embedded systems. The main goal of eDIANA is to achieve more efficiency in the use of resources, by prioritizing energy as a scarce resource, more flexibility in the provision of resources and better condition awareness for public and for service and infrastructure owners. This is planned to be achieved through the deployment of embedded systems to the building environments. With the collaboration of industry and universities across Europe (see Fig.9.1) eDIANA is a strong application-oriented project which is focused on the design, development and validation of the EDP, which

will integrate intelligent embedded devices, installed in residential and non residential buildings to improve energy efficiency and to optimize the overall energy consumption, production and storage. EDP integrates intelligent embedded devices, installed in residential and non residential buildings to improve energy efficiency and to optimize the overall energy consumption, production and storage. Similar projects can be listed as: E3SoHo[8], REMODECE[17], IntUBE[12], BeyWatch[4], AmI-MoSES[3], AIM[2], and SmartHouse/SmartGrid [20].

Figure 9.1: The eDIANA Project

9.1 eDIANA Scenario

Based on many parameters like user behavior, comfort level, utility prices, seasonal and daily changes on weather conditions EDP will provide instantaneous feedback to the users in order to ensure their partaking in the energy conservation by means of motivations of the use of right appliances at right time. The eDIANA strategies in order to save energy can be classified in five categories; user awareness, heating/cooling control, user's activity and comfort level, energy generation and storage, and general operation. Specific strategies for each of these groups are summarized in Table 9.1.

Table 9.1: Energy Saving Strategies in eDIANA

User Awareness	**Heating/ Cooling Control**	**User's Activity and Comfort Level**	**Energy Generation and Store**	**General Operation**
Monitoring over Internet	Accessing Weather Forecast Results	Adaptation to the User's Behaviour	Energy Generation (Solar, wind, cogeneration, etc.)	Centralized Management and Control
Energy Price Information	Prediction Model for the Indoor Thermal Performance	Reacting According to the User's Current Activity	Energy Demand	Management and Control over Internet
Monitoring Total Consumption in Cell		Adjustable Comfort Level Thresholds or Modes	Stored Energy (In batteries / thermal capacity)	Demand Response
Monitoring Individual Consumptions in Cell			Energy Trading	Automated Performance Scheduling
Visual Indications to motivate user to save energy				Overriding the Automated Performance Scheduling
Benchmarks between similar premises				
Alerts about unusual consumption (a open window, etc.)				
Reporting Normalized to the Weather Conditions.				

9.2 eDIANA Reference Architecture

The eDIANA reference architecture has a hierarchical organization: first level is the Cell which could be a single house, apartment or working unit; and the second level is Macro-Cell, which is a group of Cells. Each Cell is managed by a CDC which gathers information provided by the Cell level devices. The CDC takes decisions inside the Cell according to Cell level monitoring and directives coming from Macro-Cell Concentrator (MCC) which typically involves the energy trading with the utility both being a consumer and a producer. The MCC decides the general energy consumption strategy of the group of Cells attached to it. In Figure 9.2 the relationship between the CDC and the MCC can be seen.

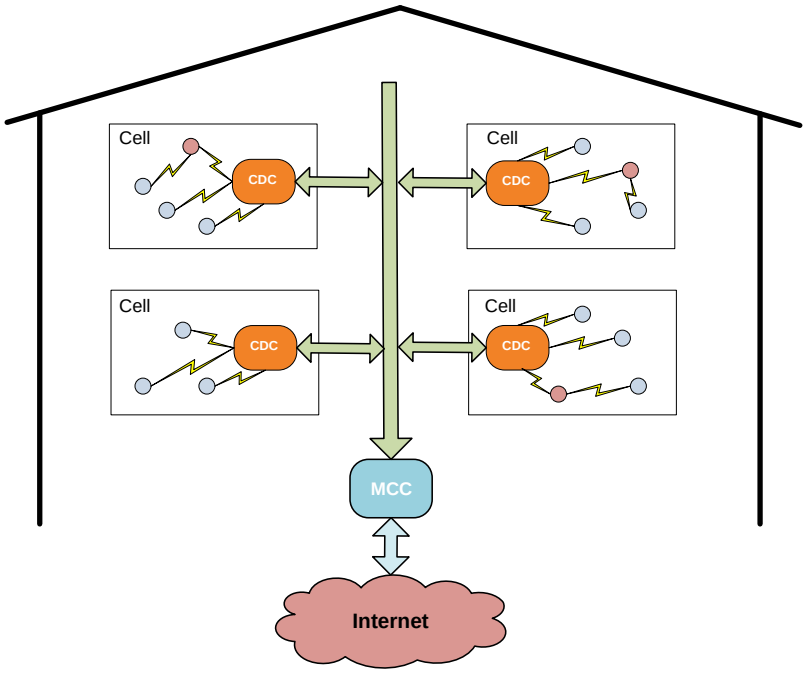

Figure 9.2: The eDIANA Hierarchy

The eDIANA Reference Architecture is an open architecture; it does not demand a unique implementation of its elements so it enables the addition of new components, as well as changing and updating the current elements as long as they are compliant with the architecture. eDIANA Reference Architecture defines several components for

each layer, respect to the functionality that each layer should provide. Cell level integrates all components that interact with the building elements and devices: appliances, lighting, HVAC, etc. eDIANA Reference Architecture defines the following component types inside eDIANA Cell Level (see Fig. 9.3): Cell User Interface (CUI), Cell Monitoring and Metering (CMM), Cell Control and Actuation (CCA), Cell Generation and Storage (CGS).

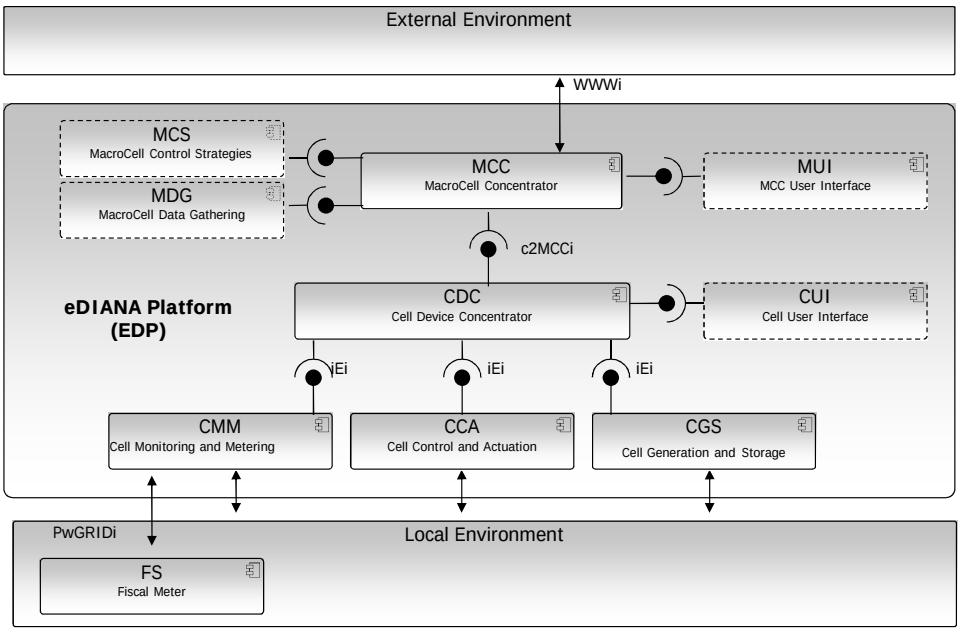

Figure 9.3: eDIANA Reference Architecture

An implementation of eDIANA Reference Architecture can separate these components into different devices or integrate some of them into one. The CUI is the component that provides Cell level information to the users of the platform. CMM devices are the sensors such as temperature, lighting, sun radiation, people presence, energy generation, smart meters of plugged devices; CCA devices are the actuators such as light dimming actuators, blind actuators, smart appliances, etc.; and CGS devices are the energy generation and storage systems. The communication between these components is hierarchical; CMM, CCA and CGS can only communicate with the CDC. The architecture defines a unique interface, Intelligent Embedded Interface (iEi), to accomplish the tasks. The CDC is in charge of the communications between the Cell level

and MacroCell level. This communication is made between the CDC and the MCC, through the Cell to MacroCell Concentrator Interface (c2MCCi). The components that the eDIANA Reference Architecture defines at MacroCell level are: MacroCell Control Strategies (MCS), MacroCell Data Gathering (MDG), MCC User Interface (MUI). These components placed in the MacroCell level interact with the external environment in order to obtain information to elaborate the necessary strategies. These strategies take into account the external environment using the data coming from power grid (through PwGRIDi), the web based resources (through WWWi), and the monitoring information coming from the Cells when producing the recommendations to the CDC. Three interfaces related with MCC are:

- **C2MCCi:** Cell to MacroCell Concentrator interface that specifies all the communication between the Cell and the MacroCell
- **PwGRIDi:** the Power Grid interface specifies the communication between the MCC and the Power Grid
- **WWWi:** the internet interface specifies the communication with external resources.

9.3 ZigBee in eDIANA Scenario

In eDIANA, ZigBee is used for the communications between the Cell level devices. More specifically ZigBee finds place inside the iEi in the eDIANA Reference Architecture.

9.3.1 Intelligent Embedded Interface (iEi)

iEi is a low-cost, low power embedded electronic design that is integrated in the eDIANA Cell level devices. Targeting low cost in the design, requires it to be as simple as possible, and at the same time flexible enough to be accommodated at each Cell level device. The iEi allows the eDIANA Cell level appliances (white goods, consumer electronics, HVAC systems, sensors, lighting devices, energy generation and storage systems, etc.) to be connected to the CDC. This connection allows CDC to read status information from the appliances, monitor their power consumption and control them according to eDIANA energy saving strategies (see Table 9.1). The conceptual model of iEi can be seen in Fig.9.4.

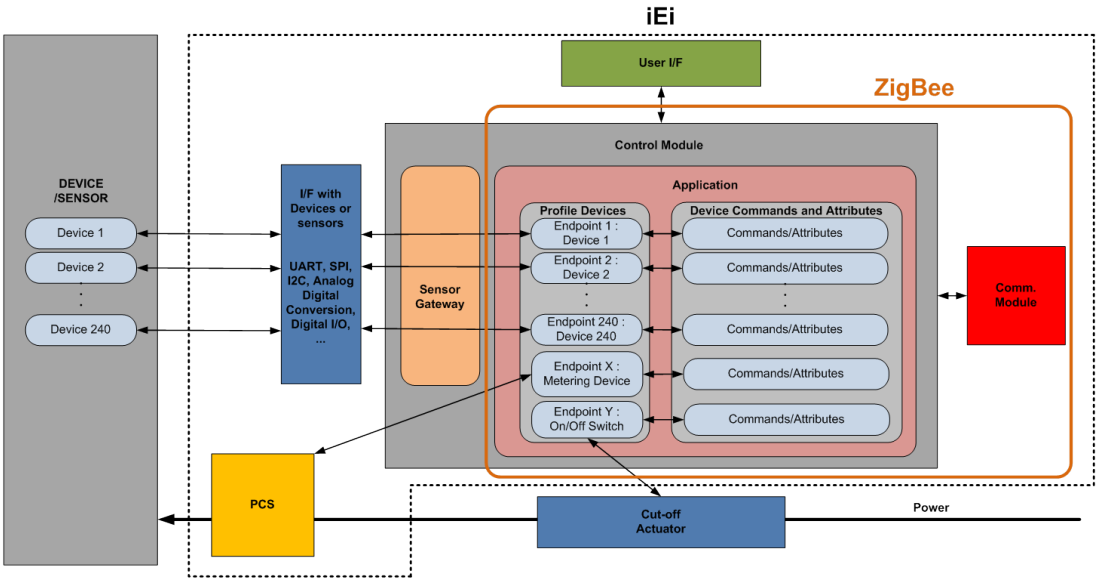

Figure 9.4: Intelligent Embedded Interface (iEi)

- **User I/F:** The iEi has a very basic and easy-to-use user interface, including, for instance, buttons in order to allow the network/device discovery or other configurations, and LED indicators for the configuration confirmation and status information (i.e. communications activity).

- **Sensor Gateway:** This simple gateway connects the physical sensors/devices with the logical ZigBee Devices. Possible interfaces include General Purpose Input Output (GPIO), Analog-to-Digital Converters (ADCs), Inter-Integrated Circuit Interface (I2C), Serial Peripheral Interface (SPI), Ethernet, UART and proprietary interfaces.

- **Control Module:** This is the central processing module of the iEi.

- **Communications Module:** IEEE 802.15.4 compatible transceiver.

- **Power Consumption Sensor (PCS):** Power consumption sensor is integrated to the iEi in order to provide power consumption information to the CDC.

- **Cut-off actuator:** An external cut-off actuator is included in iEi for dummy devices as a mechanism to easily turn on/off them.

Several appliance types can be connected to an iEi, including intelligent appliances (devices with processing capabilities and digital interfaces), dummy appliances (devices with no built-in intelligence that can only be managed in an on-off switch, using an external actuator), the CDC and sensors (both plugged and battery operated). The different building blocks of the iEi can be included/excluded depending on the appliance type.

9.3.2 Classification of Applications

Probably the most important innovation in the energy management put forward by eDIANA can be mentioned as the monitoring the real-time energy consumption of each appliance in order to implement smart algorithms. On the other hand, the implementation of smart algorithms is strictly related with the intelligent control of the actuators. Before going into the details, to establish a framework, in the following given a systematic classification of applications respect to the implementation and the generated traffic. The eDIANA implementation can be classified in two application categories that complete each other:

- **Monitoring Application (MA):** This kind of application gathers the data coming from sensors. The information can be related to the consumed energy or ambient conditions (temperature, humidity, radiance, CO2, etc.).
- **Control Application (CA):** In this kind of application controllers send commands to the controllable devices (intelligent plugs, domestic appliances, brown goods, HVAC, lights, etc.).

9.3.2.1 Monitoring Applications

In eDIANA scenario expected traffic will be MA dominant since ambient conditions and the energy consumed by each appliance will be periodically concentrated in the CDC in the Cell level. In general MAs can be distinguished according to the kind of traffic. In particular two kinds of strategy for a MA can be identified:

- **Query-Based Strategy (QBS):** A query sent by the sink triggers data transmissions.

- **Event-Driven Strategy (EDS):** An event generates the data transmission. The event can be something sensed in the environment (aperiodic traffic) or a clock indicating to the node that a packet must be transmitted at a given instant (periodic traffic).

In ZigBee, the data acquired by a sensor (e.g. occupancy) is stored at the related attribute (e.g. occupancy attribute) of the related cluster (e.g. occupancy sensing cluster) in the hosting device (e.g. occupancy sensor device) that resides in the ZigBee node. The simplified model of this relation between ZigBee node, ZigBee device, cluster, and the attribute can be seen in Fig. 7.9. Attributes can be read or written by the authorized ZigBee devices. For instance, temperature value can be read by the thermostat unit, state of a light can be changed by the light switch. ZCL general commands (see Fig. 7.7) provide the necessary tools in order to access the attributes on remote ZigBee nodes. In eDIANA scenario, MAs can be implemented in three different ways; *i)* Querying the attributes, *ii)* Periodically reporting the attributes, *iii)* Event-driven report of the attributes. In fact, first item, *i*, in the list is a QBS and the other two are EDS.

Querying the Attributes: In eDIANA scenario the CDC can query the Cell level devices when it needs to have information about the environment. Query of an attribute can be done by sending a *Read Attributes* command. Queries can also be sent to ZEDs that may be in sleeping mode. On the behalf of sleeping nodes their parents keep the query until the sleeping node wakes up and sends a data request to its parent.

Periodically Reporting the Attributes: ZigBee nodes can periodically report the data they collected from the environment to the CDC. For this kind of MA, ZCL provides five commands: *Configure Reporting*, *Configure Reporting Response*, *Read Reporting Configuration*, *Read Reporting Configuration Response*, *Report Attributes*. The *Configure Reporting* command is used to configure the reporting mechanism for one or more attributes of a cluster. For example, the CDC can configure a ZED or a ZR to report their attributes periodically and can also set the reporting time interval. When a node receives this command it replies with the *Configure Reporting Response* which informs the status of the configuration. The status can be failure if there is no such an attribute or not possible to report it. Through the *Read Reporting Configuration* command a device can read the configuration parameters of a reporting device. Reporting

devices reply the configuration command by sending a *Read Reporting Configuration Response*. A report mechanism can be set by using the *Configure Reporting* command coming from CDC or out of the box by the manufacturer. The *Report Attributes* command can be used to inform periodically CDC with the results of the measurement done by the sensor.

Event-Driven Report of the Attributes: Nodes that detect an event can simply send a *Report Attributes* command.

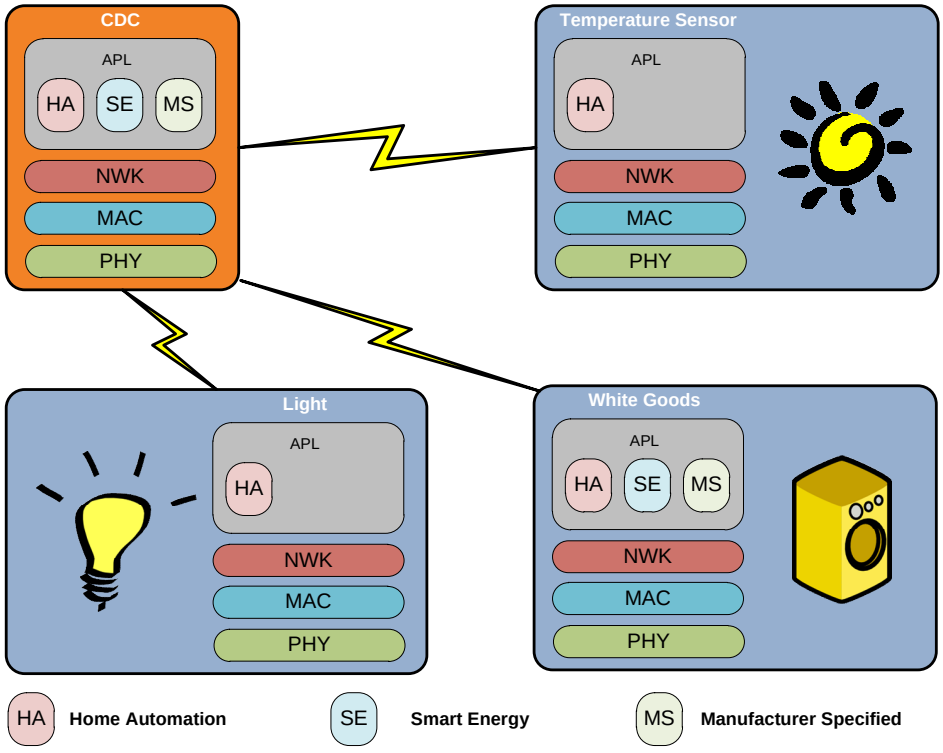

Figure 9.5: Home Automation and Smart Energy Profiles Together

9.3.2.2 Control Applications

On the other hand CAs can be implemented by using the cluster specific commands (e.g. on/off commands in On/Off Cluster) or by using the ZCL general commands (e.g. write attributes).

9.3.3 ZigBee Profiles

Most of the eDIANA Cell level appliances can be implemented by using ZigBee clusters in HA and SE Profiles. Some appliances may require the implementation of more than one profile in different endpoints to function correctly. In particular, all Cell level appliances implement HA Profile except some appliances that require the use of the Simple Metering Cluster, which is in SE Profile. In the case of white goods a Manufacturer Specific (MS) Cluster has been defined to provide devices with the requested functionalities.

Table 9.2: eDIANA Appliances and ZigBee Devices

eDIANA Appliance	**ZigBee Profile**	**ZigBee Application Domain**	**ZigBee Device**
Temperature Sensor	HA	HVAC	Temperature Sensor
Pressure Sensor	HA	HVAC	Pressure Sensor
Smart Electricity Outlet	HA and SE	General SE	Mains Power Outlet and Simple Metering
Light Sensor	HA	Lighting	Light Sensor
White Goods	HA MS	MS	White Goods
Stirling Engine	HA	HVAC	Temperature Sensor, Thermostat, Pump, Heating Cooling Unit
Presence Sensor	HA	General	Occupancy Sensor
Light	HA	Lighting	Dimmable Light, Color, Light, Switch, etc.
Door/Window Sensor	HA	General	Simple Sensor

In Table 9.2 a list of the eDIANA Cell level appliances is shown. Each appliance is mapped with the corresponding ZigBee profile, domain and device. In Fig.9.5, there is an example of Cell implementation, where a CDC, a light, a temperature sensor and a washing-machine are present.

9.4 ZigBee Driver on CDC

The CDC can be simply modeled as a Micro Processor Unit (MPU) and a ZC which are connected through the UART. The ZC interfaces the CDC to the ZigBee network as shown in Fig.9.6.

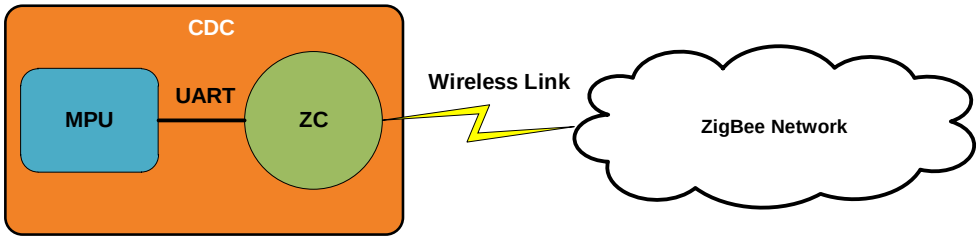

Figure 9.6: Cell Device Concentrator (CDC)

In order to provide necessary control over the ZigBee network a driver has been developed in C. The Freescale BeeStack [91] that is used in the implementation comes with a useful tool called ZigBee Test Client (ZTC). The ZTC is a small application running separately on each layer of the BeeStack. A host processor connecting through ZTC can control the ZC by Application Programming Interface (API) calls. The ZTC allows monitoring of specific BeeStack interfaces and API calls. The ZTC injects or calls specific events and commands into the interfaces between layers. The ZTC architecture and the location of developed ZigBee driver is shown in Fig. 9.7.With the aim of achieving a reliable connection over UART, ZTC defines a frame format ; each frame starts with a frame delimiter (0x02) indicating start of packet, then the opcode group, opcode, and length fields, which are 1-byte long, come. According to the opcode a variable length data payload follows the length field. Finally, at the end of the frame there is a frame checksum calculated as the XOR of all bytes in the opcode group, opcode, length, and data fields. We have written a driver that is able to send, receive and process ZTC frames. All sensor or command related traffic should be handled properly by the ZigBee driver for a successful operation of the EDP. Therefore some requirements have been set;

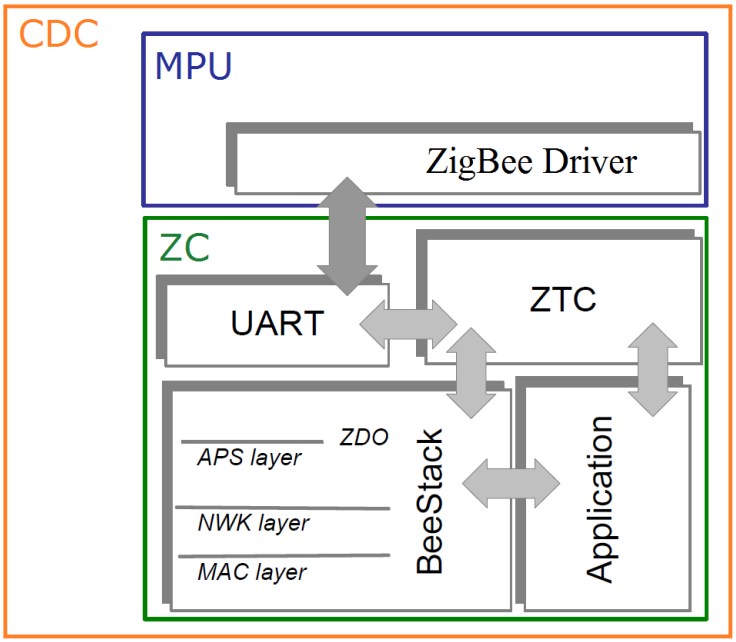

Figure 9.7: ZigBee Driver

- Communication should be asynchronous (receiving or sending messages from/to ZC can occur at random times).
- Some decisions should be taken according to the received message and the state of the network without necessarily informing the CDC.
- Architecture should be scalable so that it can be easily expanded and improved for further implementation of the possible functionalities.

Consequently, to fulfill these requirements five POSIX threads, namely ReadfromUART, ParseComingZTCmessages, StateMachine, WriteonUART, and Display (optional) have been developed to simultaneously carry out the different tasks and to share the resources. The core thread, StateMachine, is a state machine which response incoming and outgoing messages. Flow charts of first four threads can be seen in Fig. 9.8.

ReadfromUART: The ReadfromUART thread receives the raw data coming from the ZC by directly storing them in a buffer. The buffer has been implemented as a circular FIFO buffer by overwriting the oldest entries when the buffer got full.

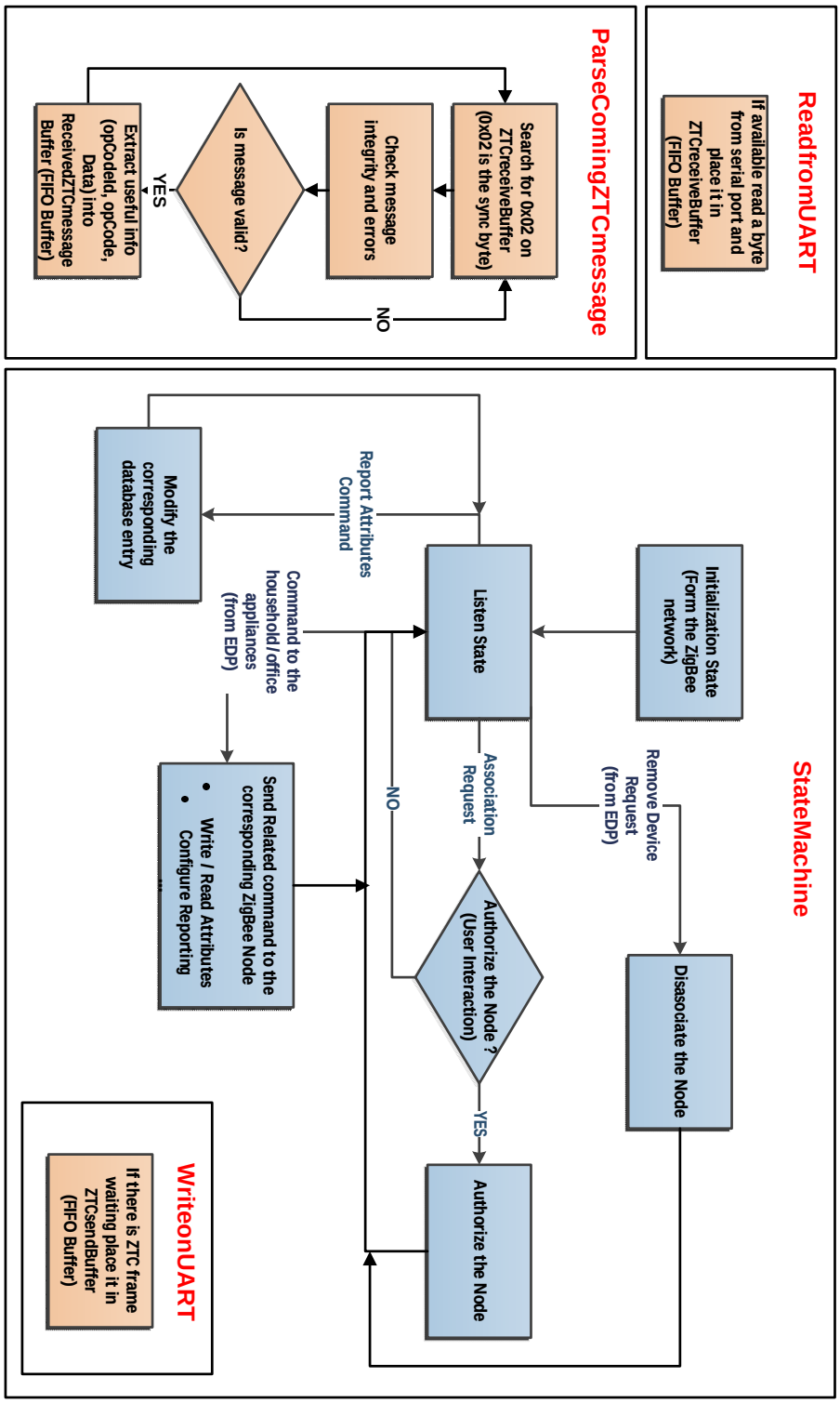

Figure 9.8: POSIX Threads Implementation of ZigBee Driver

ParseComingZTCmessages: The ParseComingZTCmessages thread parses the ZTC frames in the raw data buffer filled by ReadfromUART and processes them to extract the meaningful information. Firstly, the thread searches for a frame delimiter (0x02) in order to find the ZTC frame. Afterwards, it parses the command and checks its integrity controlling the checksum byte. If the format is correct it extracts the information in the ZTC frame, such as the OpCodeId, OpCode and data. Finally, this information is stored in a structured new buffer so that it can be easily accessed by StateMachine thread.

StateMachine: The StateMachine thread is the core of the driver since it implements the state transitions of the system. The thread picks the commands from the buffer that is previously filled by the ParseComingZTCmessages thread, then identifies them respect to their OpCode and OpCodeId. According to the different conditions, the current state of the system changes as shown in Fig.9.8. StateMachine thread also takes decisions about the routine tasks that have to be carried out without necessarily informing the CDC. Furthermore the commands coming from EDP are written to a buffer in order to be processed by WriteonUART thread.

WriteonUART: The WriteonUART thread forms the ZTC frames to be sent to the ZC. Once a command has been properly delivered, the thread picks the next one from the buffer to send it to ZC.

Display (Optional): The Display thread controls the user interface, which permits the interaction with the driver using the keyboard. For instance, "M" shows the main menu with the information regarding all the devices in the network. "P", shows commands and configuration (see Fig.9.9).

9.5 Demonstrator

A ZigBee demonstrator is installed at CSITE building in University of Bologna. The Cell environment of the e-DIANA scenario is reproduced in a down-scaled way (in terms of distances between devices, transmit power, etc.). By decreasing the transmit power, the range of the ZigBee nodes is restricted in order to fit in a small area a full functional test-bed. Since this test-bed is focused on demonstration of the communications in the Cell, there are no real household/office appliances but there are ZigBee nodes that

```
**********************************************
*    Device in Device Initialization State     *
*       Device in Network Discovery State      *
*     Device in Coordinator Starting State     *
*      Device in Coordinator Running State     *
**********************************************
*     NETWORK HAS BEEN PROPERLY ESTABLISHED    *
**********************************************

Do you want to allow 0 0 0 0 0 0 0 1 to join the network? [y/n]
y
y
Nwk Key:  3c d7 1a 58 a2 92 12 12 49 8e 7b 1b b2 cb a6 fd
A new device has been associated
Simple Descriptor Response
Complex Descriptor Response
m
m
*******************************************************************************
************************************ MAIN  MENU ********************************
*******************************************************************************
* Number of Addresses stored: 1        * ProfileId * DeviceId *   Manufacturer   *
*******************************************************************************
* NWK Address 1 : 81 e8                *    1  4   *  3  1   *  42  42  42  42  *
* MAC Address 1 : 0 0 0 0 0 0 0 1      *          *         *                  *
*******************************************************************************
*                         Type P if you want More Options                     *
*******************************************************************************
```

Figure 9.9: A Snapshot of ZigBee Driver User Interface

simply simulate the typical household appliances in order to demonstrate some of the energy saving strategies given in Table 9.1.

In the demonstrator the ZigBee end point configuration, given in Fig.9.10, is used. In the ZC part of end point diagram there are two ZigBee devices namely *Combined Interface* and *Energy Service Portal* while in the iEi part there are a *Generic Device* and a *Metering Device*. The default clusters coming with these devices are shown in Fig.9.10 but the clusters indicated in red in Fig.9.10 are not used because of not applicability in our scenario. Standard security with non-preconfigured network key and HA default trust center link key is used in order to test the security in eDIANA scenarios as well. For the sake of simplicity types of appliances in the demonstration is limited into three, namely On/Off Appliances, Programmable Appliances, Heating or Cooling Capable Appliances. No new cluster is introduced but already defined clusters and attributes in ZigBee are used to test the communications in the Cell.

On/Off Appliances: All simple appliances (Ex: On/Off Light) can be realized as represented in Fig. 9.11.

Programmable Appliances: Programmable appliances like washing machine, dishwasher, etc. can be realized in general as given in Fig. 9.12.

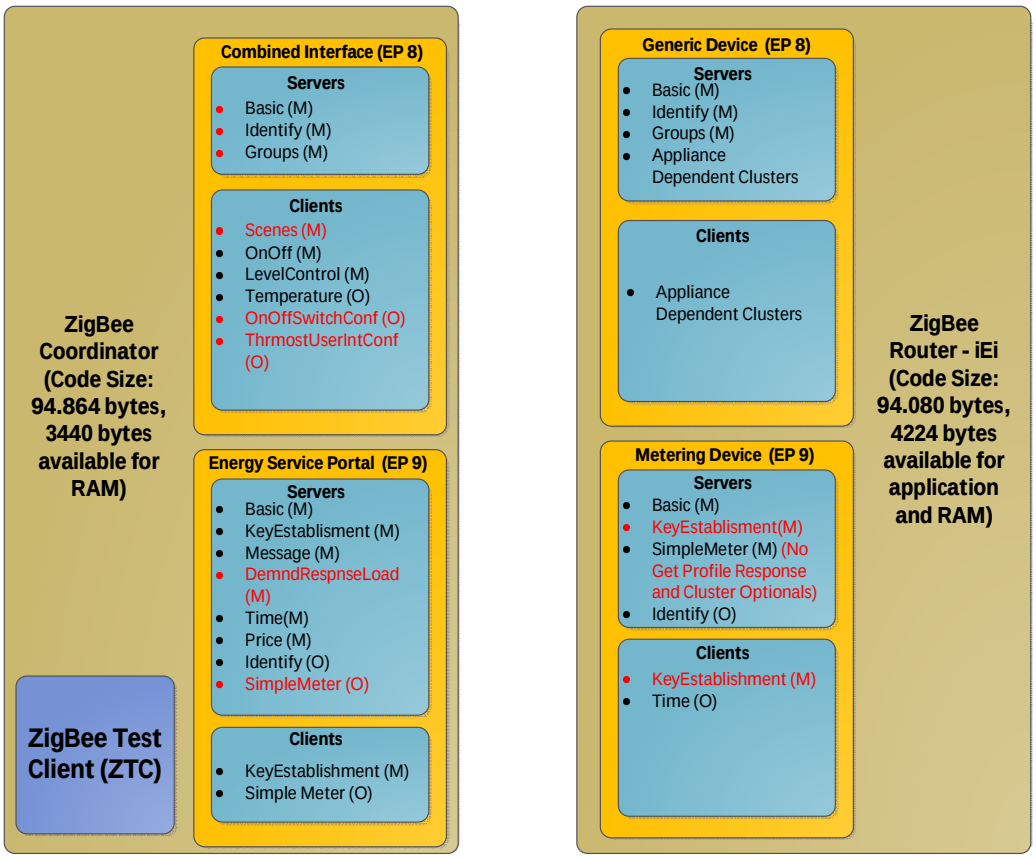

Figure 9.10: End Point Configuration

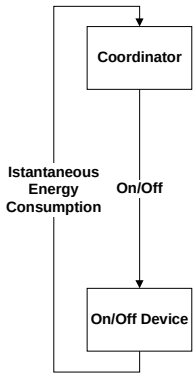

Figure 9.11: Flow Chart of On/Off Appliances

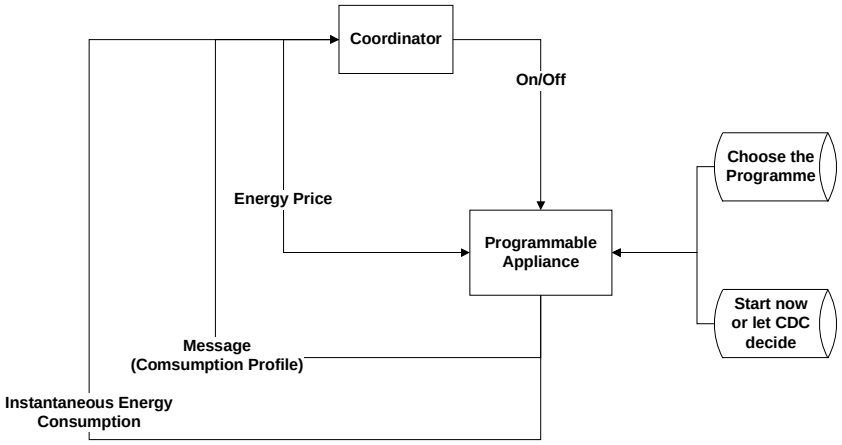

Figure 9.12: Flow Chart of Programmable Appliances

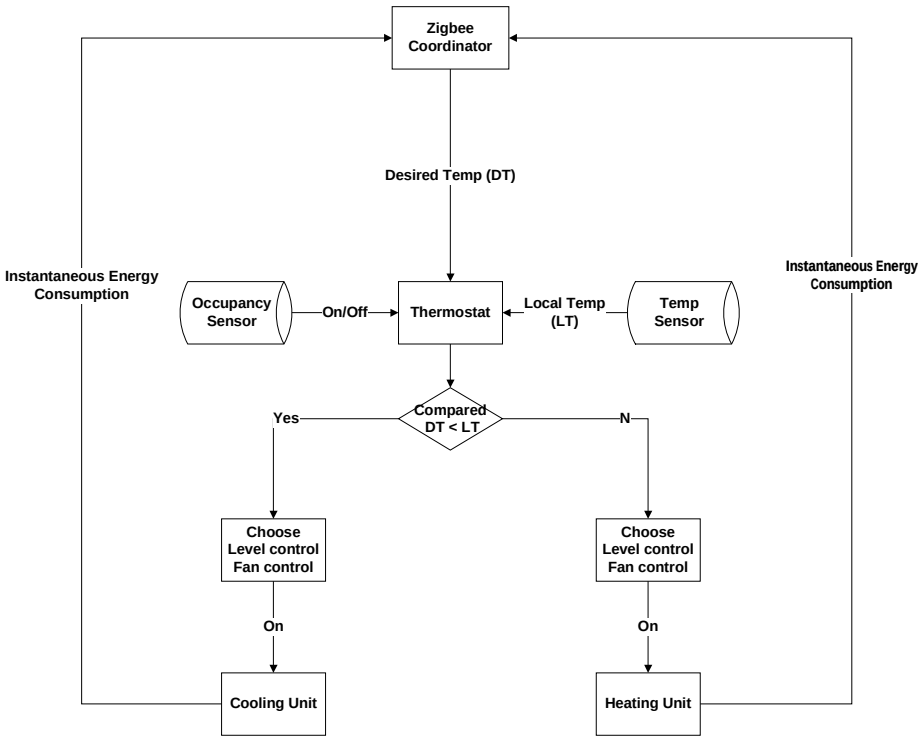

Figure 9.13: Flow Chart of Heating or Cooling Capable Appliances

Heating or Cooling Capable Appliances: Several appliances such as fridge, water heater or HVAC can be realized as a generic heating or cooling device by implementing the flowchart shown in Fig. 9.13.

9.6 Query Measurements

After describing the scenario, implementation and the demonstrator, now in this section we will describe the measurement campaign that we have conducted in the eDIANA compatible demonstrator. ZigBee channel access is based on IEEE 802.15.4 unslotted CSMA-CA [93]. In the literature, considerable effort on modeling CSMA-CA and measurements about PER can be found. A fine collection of references are reported in [89]. However articles on un-slotted CSMA-CA are not numerous respect to the slotted version. Among them an analytical model which gives channel access probabilities and delays for un-slotted CSMA-CA when nodes are perfectly synchronized to the query can be found in [45] on a later work [67], also some experimental results about channel access probability in query based scenarios including the effect of packet capture are provided. Apart from the efforts to model CSMA-CA as a Markov chain, there can be found another study on unslotted CSMA-CA [99], which uses busy cycle of M/G/1 queueing system. So far in the literature studies about star networks are significantly dominant in numbers. In contrast, here in this section we are experimentally focusing on channel access probability and latency on a multi-hop ZigBee network, furthermore reporting relaying time requirement of a typical ZigBee router. A series of experiments have been conducted in order to measure the average latency and PER of a single query when in the rest of the network EDS or QBS (see Section 9.4) is implemented. Here in this section we are measuring the PER and latency in the application scenario of the Section 9.5. With the benefit of mesh topology ZigBee nodes can cover large areas and nodes in the network can substitute down links by finding out new routes. But before letting ZigBee nodes to establish their own routes as in Section 9.6.4, we forced a fixed two-hop topology by restricting the number of children and routers that can associate to the ZC. In the experiment setup, ST SPEAr600 [79] evaluation board and ZC are formed the CDC. We used Freescale MC1322x [63] Platform with BeeStack ZigBee PRO v3.0.7 codebase in the ZigBee nodes. In all the measurements CDC and a ZED, namely ZED_0, were always present but the number of traffic generating nodes was different to

change the measurement conditions. In fixed topology nodes were randomly distributed over a 1x2 meters table while in the mesh topology we distributed them as in Fig.9.21. In the measurement campaign ZED_0 and ZC were always connected to the SPEAr600 via UART to inform SPEAr600 about the send/receive time instances. Every measurement is initiated and timestamped by SPEAr600 therefore measurements reflect the reaction performance of eDIANA application that runs on SPEAr600, however results are strictly related with the network layer performance of the ZigBee nodes. During the measurements transmit power was set to -1 dBm and for all set of measurements we have collected at least 10000 samples. In all experiments ZEDs simulated smart household devices (see Section 9.5) however we did not pay attention to the outcome of the simulation since the main intention was to obtain the network performance by reading the *CurrentSummationDelivered* attribute (attribute that keeps the power consumption value) on ZED_0 in both QBS and EDS. For latency and PER measurements 41-byte long *Read Attributes* commands were sent by the CDC. These commands are reached to ZED_0 at different monitoring conditions and in different topologies. Just after receiving the query ZED_0 is replied by sending a 38-byte long *Read Attributes Response* packet. In this way, we obtained the both way latencies and PERs in QBS and EDS. But before providing these average latency and PER measurements in different traffic conditions, we provide the typical relaying time for a ZR in Section 9.6.1 in order to create a baseline for the rest of the measurements. During the calculations we neglected the time necessary for the penetration of electromagnetic waves and the penetration of the signals on the cables in the UART communications.

9.6.1 Hopping Time over the ZR

In ZigBee, a node accesses the channel by using unslotted CSMA-CA defined in the 802.15.4 Standard [93]. In CSMA-CA with default parameters first backoff phase introduces a uniform delay between 0 to $2240\mu s$. This results in an average delay of $1120\mu s$ if the node is not competing with other nodes for the channel access. After the backoff, node enters the sensing phase for $128\mu s$. Finally, if the channel is found free the packet is sent. The average delay introduced by CSMA/CA can be calculated as $T_c = 1120\mu + 128\mu = 1248\mu$ if there is no other traffic. Packet durations, T_{tx}, for *Read Attributes* request and *Read Attributes Response* are $1248\mu s$ (39 bytes) and $1312\mu s$ (41 bytes) respectively.

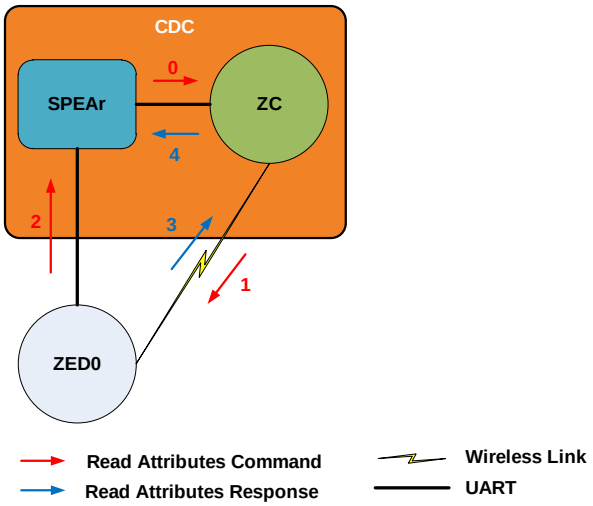

Figure 9.14: Analysis of Hopping Time - One Hop

9.6.1.1 One-Hop Latency

In order to give some insights on the typical hopping time over a ZR, first we measured the one-hop latency (without any traffic) between ZC and ZED_0 as shown in Fig. 9.14. The time that the packet needs from SPEAr, passing through the ZC (Step 0), to the ZED_0 (Step 1) resulting in the request time and the duration of the way back to SPEAr (Step 3 -4) resulting in the response time has been measured. Step 2 indicates that the ZED_0 informs SPEAr that the packet has been received since latency measurement takes place on SPEAr. For the request $T_{1HOP} = 7358\mu s$, for the response $T_{1HOP} = 6179\mu s$ average hopping times are measured.

9.6.1.2 Two-Hop Latency

After the one-hop measurement a ZR has been added to the topology as shown in Fig.9.15. In this new topology the time that the packet needs from SPEAr, passing over ZC (Step 0 - 1) and the ZR, to the ZED_0 (Step 2) resulting in the request time and the duration the way back to SPEAr600 (Step 4 - 6) resulting in the response time has been measured. Step 3 indicates that the ZED_0 informs SPEAr that the packet has been received since latency measurement takes place on SPEAr. For the

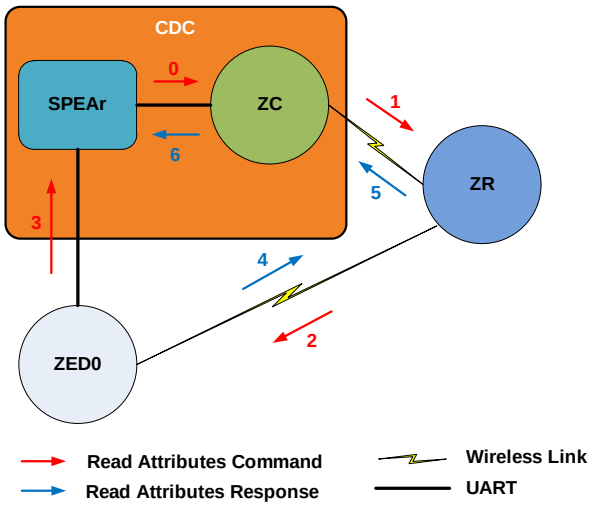

Figure 9.15: Analysis of Hopping Time - Two Hops

request $T_{2HOP} = 12680\mu s$, for the response $T_{2HOP} = 11370\mu s$ average hopping times are measured.

Table 9.3: Hopping Time Measurements

	T_{1HOP}	T_{2HOP}
Read Attributes Request:	$7358\mu s$	$12680\mu s$
Read Attributes Response:	$6179\mu s$	$11370\mu s$

Average time needed by ZR to process the packet is equal to $T_r = (T_{2HOP} - T_{1HOP}) - (T_c + T_{tx})$ which results $2826\mu s$ and $2631\mu s$ for request and response respectively. So in general, it can be stated that average time needed by ZR to process the packet is approximately $3ms$ which also includes transceiver turnaround and acknowledgment durations. We can further analyze the one-hop query case to obtain the total time necessary for the processing on SPEAr, ZC and ZED_0 all together ; $T_{spear} + T_{zc} + T_{zed} = T_{1HOP} - (2T_u + T_c + T_{tx}) = 4514\mu s$ where T_u is the time necessary for UART transmission. In the measurements UART baudrate was 115200bps and we defined a two bytes long protocol for the communications over UART which introduces $174\mu s$ delay.

9.6.2 QBS in Fixed Topology

We synchronously queried the ZR and a number of ZED as shown in Fig. 9.16 by using the same group address in all nodes except ZC. Group addressing in ZigBee preserves the bandwidth by using only once the common links of the routes to the destination group members. In our configuration, firstly ZC unicasts the *Read Attributes* command to ZR (step 1) and ZR replies with *Read Attributes Response* before forwarding the command as a broadcast to all ZEDs (Step 2) including ZED_0. Thus ZEDs in the setup receive the query at the same time and reply it with *Read Attributes Response* (Step 4) in competition with each other to access the channel.

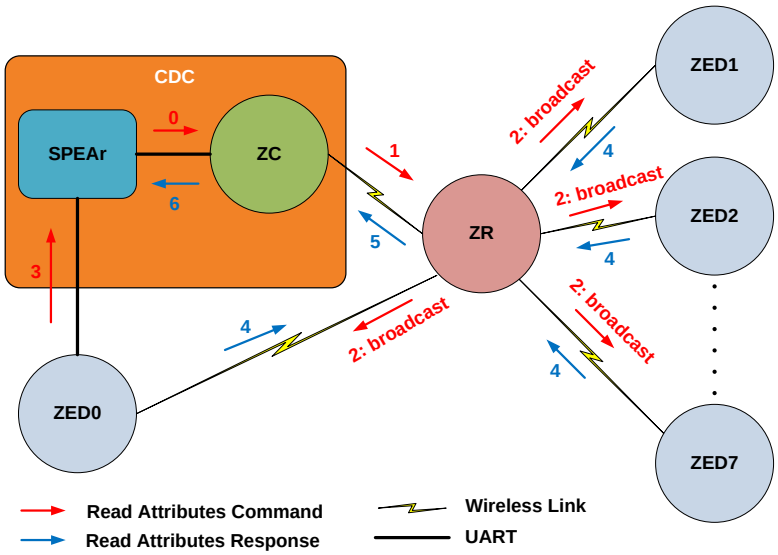

Figure 9.16: Synchronized Query

We just processed the responses coming from the ZED_0 and queries reached to ZED_0 in order to obtain the latency and PER. In the measurements ZEDs were always awake to receive the broadcast message coming from ZR at the same time even though typically ZEDs are utilized as sleeping nodes. In Fig. 9.17 measurement results are given in box-plots.

In the query median value of latency, which is around $34ms$, is independent to the number of ZEDs since query finishes before ZEDs start transmitting any packet. PER is less than 1% in all cases. On the other hand for the responses, the increase on the

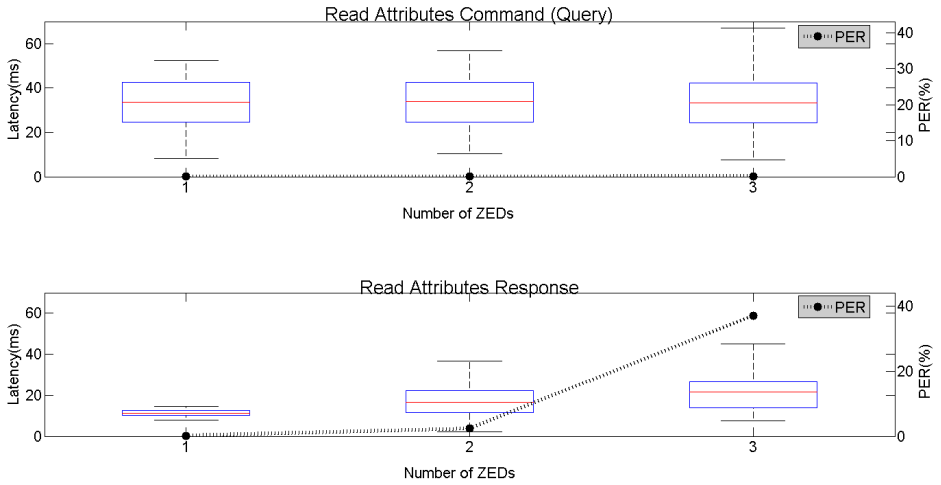

Figure 9.17: Group Query

latency and PER with respect to the number of ZED is clear because of the competition in the channel access. Median values of latencies are around $11ms$, $17ms$, and $22ms$ with respect to the number of ZED. PER is significant when there are three ZEDs. In this setup the ZR also replies to the query as a part of the destination group therefore, query time is considerably higher than the response time even though, both of the routes are two-hop to the destination. Lastly worth to note that if there is only one ZED (which is ZED_0) median value of the response latency is close to T_{2HOP} value in Section 9.6.1. This is an expected result since these two set of experiments are identical in practice except using the group addressing.

9.6.3 EDS in Fixed Topology

In event driven approach periodic traffic is generated by the nodes from ZED_1 to ZED_7 as shown in Fig. 9.18. They periodically reported *CurrentSummationDelivered* attribute to the ZC over the ZR during their polling cycle (ZEDs periodically wakes up and polls the ZR for data). The length of the report was 45-byte.

In these conditions we continuously queried ZED_0 with *Read Attributes* commands similar to the previous experiments but in this case ZED_0 was a sleeping device so it was periodically waking up and polling the ZR for any waiting query. In fact, polling mechanism in ZigBee specifications allows the device to temporarily increase its polling

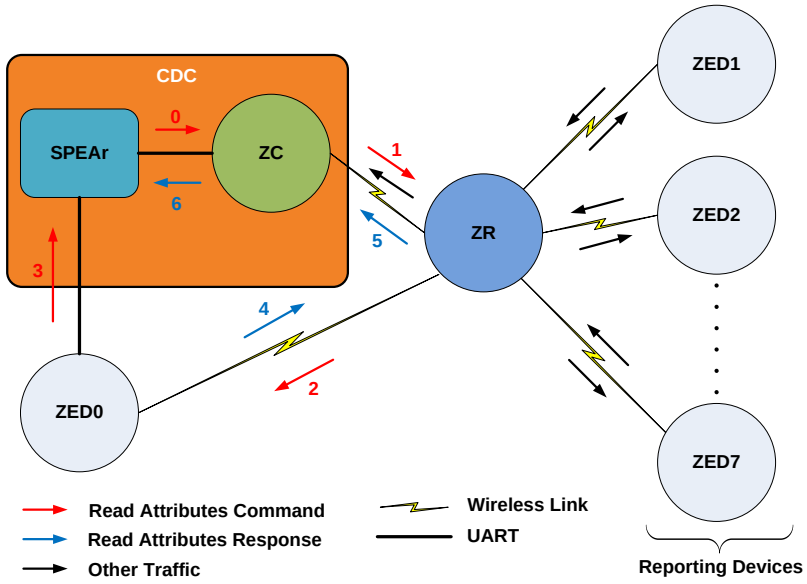

Figure 9.18: Periodic Traffic

rate if there is a continuous data stream by ensuring that it polls its parent at least once every *macTransactionPersistenceTime* seconds. In the ZED_0 *macTransactionPersistenceTime* was $200ms$. This means even though polling interval is set to another value ZED_0 can poll its parent in $200ms$ intervals if there is a continuous data stream like our queries.

In the measurements first we set the reporting interval equal to $3s$ which is also equals to the polling interval for all ZEDs (ZED_0 just polls never reports) and obtained the results in Fig. 9.19. Box-plots for different number of reporting ZEDs (excluding ZED_0) are all similar and PER values in all cases are less than 1%. Query time is around $202ms$ which is highly dependent to the *macTransactionPersistenceTime* value. On the other hand response time is like synchronized query case close to T_{2HOP} value of Section 9.6.1. This means 3 seconds reporting interval is large enough not to impose any limitation on querying the ZED_0. Thus we have increased the frequency of reporting by setting reporting and polling interval to 200ms which is equal to *macTransactionPersistenceTime* and obtained the results shown in Fig.9.20. In the figure median values on the box-plots are similar to Fig. 9.19 both for query and response but the increasing number of reporting ZEDs clearly populates the number of outliers

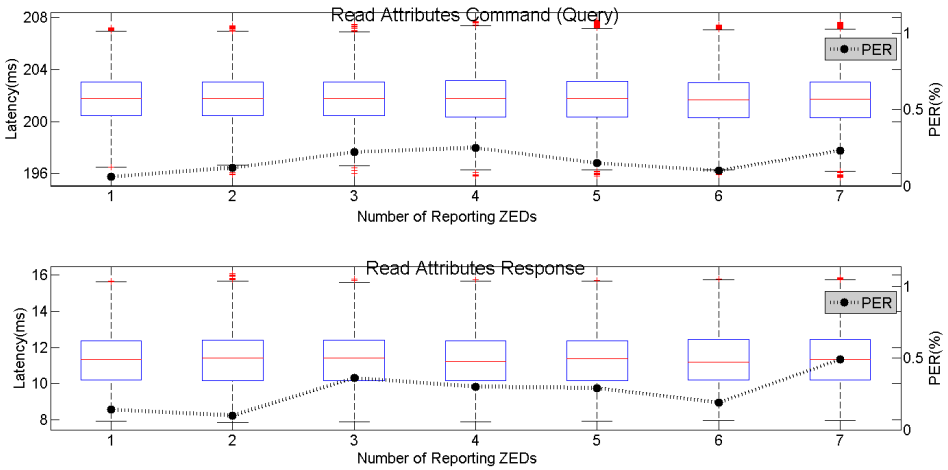

Figure 9.19: Query in Periodic Reporting ($\Delta T = 3s$)

in the box-plots by increasing the variance of latency. PER is less than 1% up to four reporting ZEDs however the beginning of a upward trend, that starts at two reporting ZEDs, can be seen in both query and response.

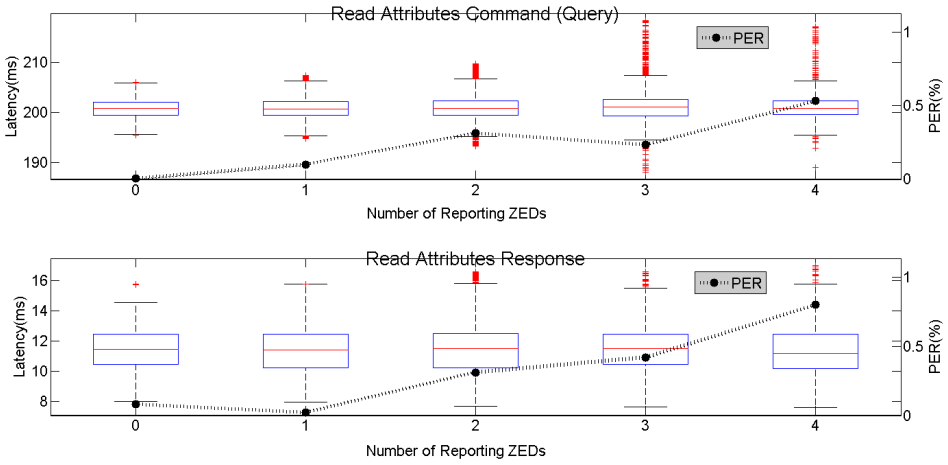

Figure 9.20: Query in Periodic Reporting ($\Delta T = 200ms$)

9.6.4 Mesh Topology

In CSITE Building at University of Bologna, we distributed ZigBee nodes without any restriction on the topology and they formed a mesh network with 1 ZC, 6 ZRs and 8

ZEDs. The locations of nodes and an instance of routes that we have observed during the measurements by using a sniffer is shown in Fig. 9.21. Connectivity in such an office environment dynamically change, therefore there can be unexpected links. For example ZED_0 and ZED_3 are connected to the distant routers rather than the expected ones.

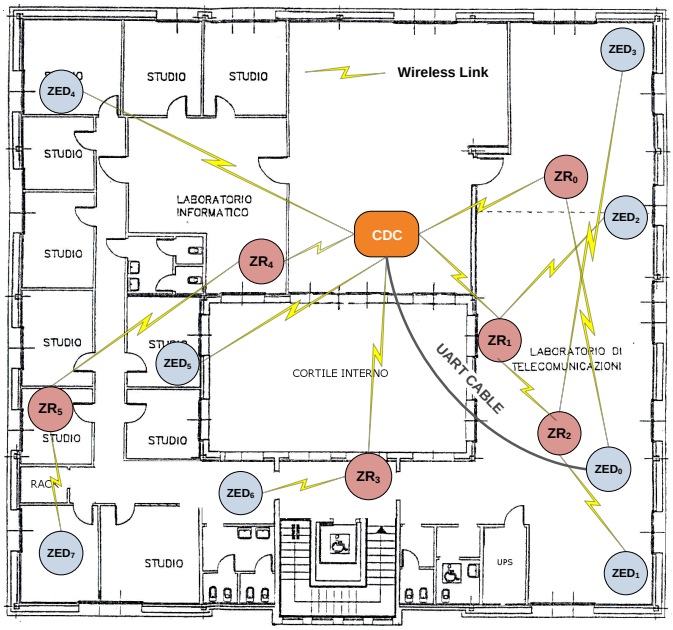

Figure 9.21: An Instance of Mesh Routes

Similar to the previous experiments we have measured the latency in two different settings; firstly we set ZED_1 to ZED_7 as reporting devices and queried sleeping ZED_0, secondly we turned off the reporting and sleeping mechanism in all ZEDs including ZED_0, assigned the same group address to ZR_0, ZR_1, ZR_2, ZED_0, ZED_1, and ZED_2 and synchronously queried all. As previously mentioned latency and PER measurements were done always between ZC and ZED_0 and in mesh case we have never observed more than two hops between these two nodes. The results are shown in Fig.9.22.

In periodic reporting case, median value of the query and response latencies are around $202ms$ and $12ms$ respectively. These results are close to the results in Fig.9.19. In group query, median value of the query is around $34ms$ and the median value of response is around $22ms$ which are close to the results of 3rd column in Fig.9.17. PER in the response of group query is significant.

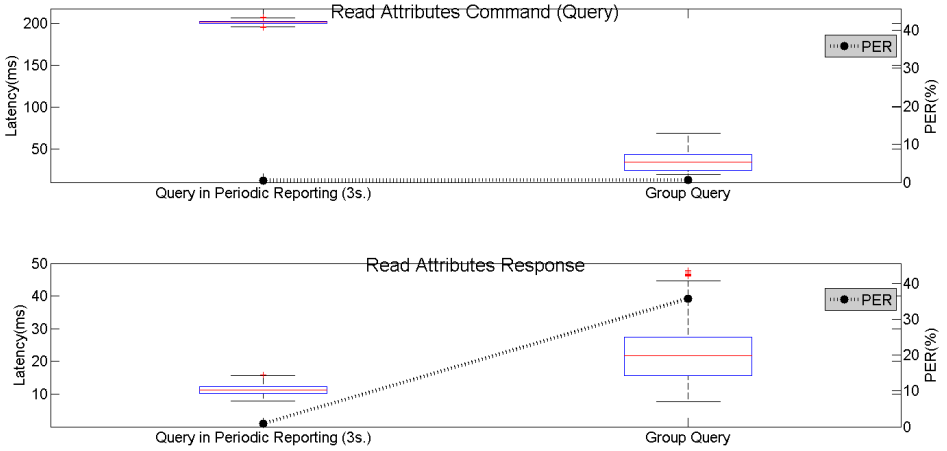

Figure 9.22: Mesh Measurements

9.7 Conclusions

Firstly, in this chapter we have described the eDIANA application scenario and the implementation in order to explain the test-bed that we have used in our measurement campaigns in Section 9.6. In all measurements except the group query with 3 ZEDs we have observed PER less than 1% and response latency was always less than $50ms$. One of the important outcome of the measurements was observing the PER bottleneck in group query with 3 ZEDs in both mesh and tree topologies. For monitoring applications instead of querying a group of three or more nodes, configuring a periodic reporting mechanism is clearly should be the way to be followed. Actually when there are more than one node synchronously queried or many nodes in a large-scale deployment are competing for the channel access, the probability to have concurrent (parallel) transmissions is relatively high, therefore one of the first steps on the analysis of a query scenarios, can be to have a better understanding of the concurrent transmissions in the PHY as we discussed in Chapter 5).

Chapter 10

Conclusion

The thesis study covers all the layers in 802.15.4/ZigBee stack architecture starting from the PHY layer to the APL. Especially Part I constitutes a baseline for the application oriented work in Part II. Whether to use RSS as a metric for the PER of a link or not is a topic discussed in [94], [76], [48]. In our transceiver platform we have showed in Chapter 3 that RSS is a good indicator for the PER, then relying on RSS, we have experimentally obtained the packet capture probability conditioned to SIR in Section 4.2. The measurement methodology used in Section 4.2 drastically differs from the traditional ways used in the literature. It is not a high precision measurement done by using coaxial cables, since we believe this does not really reflect the normal operation conditions, but it is still a novel approach with respect to the rest of the measurements in the literature since we find the instantaneous SIR first, then average it over the statistically significant samples in order to obtain the PER which results with a curve that reflects quite good also the behavior in the measurements given in [48, 76, 94]. Therefore, in the light of these different studies, done with different transceiver platforms, we state that the typical behavior of a non-coherent IEEE 802.15.4 transceiver in concurrent transmission should be like that given in Fig. 4.3. With this in mind in order provide an explanation to this behavior in Chapter 5, we have mathematically analyzed the concurrent transmission in O-QPSK including the pulse shaping and spreading in 2.4 GHz PHY of IEEE 802.15.4 and obtained the 0dB minimum SIR to start receiving packets in non-coherent demodulation. To the best of our knowledge so far, there is no such mathematical analysis in the literature. The threshold agrees with the measurements in Section 4.2 and in [48, 76, 94] even though in the analysis in

Chapter 5 we neglected the noise for the sake of simplicity. Finally at the last chapter of Part I we have discussed a novel approach to measure the RSS in 802.11 by using 802.15.4 compatible nodes.

On the other hand in Part II, in the scope of two industrial projects, there are useful application practices and performance measurements starting from the routing in ZigBee in Chapter 8 where we describe a network with 228 nodes deployed on a planar grid in the high rack storage area of a warehouse for the Centrale Adriatica Project. This deployment provided a nice test-bed to observe the route formations and the statistics on the number of hops, link and path costs in DV routing. The measurements have a potential to be used as a reference measurement on the analytical routing models or simulations in planar sensor deployment studies. Finally in Part II, with the formal approach of a European Commission project starting from the energy saving scenarios and ending up with the application there is a complete project experience of eDIANA Project in Chapter 9. The energy saving strategies given in Table 9.1 is the classified collection of the ideas sourced from the project. The multi-thread ZigBee driver developed for eDIANA in Section 9.4 can be considered as an example of a state machine for interfacing ZigBee to a gateway. Also measurements done with the eDIANA compatible test-bed revealed the bottleneck with synchronized queries in Section 9.6, besides providing useful information on the typical hopping time over the nodes. Actually when there are more than one node synchronously queried or many nodes in a large-scale deployment are competing for the channel access as in Chapter 8 or Chapter 9, the probability to have concurrent transmissions is relatively high, therefore the discussion on the concurrent transmission in Chapter 5 is also essential for the Part II.

Appendix A

802.15.4 Transceiver Comparison

In the design of WSNs, there is no one-size-fits-all architecture. The applications drive the requirements. However, 802.15.4 Standard has being widely supported by a great number of research institution and universities. It has being envisioned as a strong candidate for WSNs. For instance, Zigbee and 6LowPAN [1] are based on 802.15.4 Standard and TinyOS [24] can send 802.15.4-compliant packets. Many of the research conducted in the field of WSNs and Home Area Networks (HANs) are based on 802.15.4. In the market there are various types of 802.15.4 compatible RF platforms. IEEE 802.15.4 can be implemented in two relatively diverse methods; by using TinyOS based devices or by using IEEE 802.15.4 Standard stack based devices.

A.1 TinyOS Compatible

TinyOS is an event driven open source operating system for embedded sensor networks developed by UC, Berkeley. Because of its open source license it is possible to implement almost any kind of medium access and routing protocol through changes in the source code, as long as the hardware allows. However this may need considerable effort in most of the cases. In current TinyOS, the available MAC options are: B-MAC [87] and S-MAC [18]. Using B-MAC, 802.15.4-compliant packets can be generated in TinyOS. There is also an open-source implementation of IEEE 802.15.4/ZigBee for TinyOS [53]. A recent release of TinyOS supports hardware platforms from Atmel and Texas Instruments. University of California, Berkeley has open source reference designs, called Mica2Dot[15], Mica2[14], and Telos[22]. The motes produced in Berkeley are available

to the general public through a company called Crossbow [7]. Crossbow's product family also consists of improved versions of these motes called Micaz[16] and Iris[13]. All these motes are TinyOS-compatible but the well known TinyOS simulator TOSSIM [75] can only emulate Mica2 and Mica2Dot motes. Support for the other motes is not currently available in the simulator.

Mote Hardware Platform		**IRIS**	**MICAz**	**MICA2**	**MICA2DOT**
Models (as of April 2005)		XM2110	MPR2400	MPR400/410/420	MPR500/510/520
MCU	**Chip**	ATMega1281	ATMega128L		
	Type	7.37 MHz, 8 bit			4 MHz, 8 bit
	Program Memory (kB)	128			
	SRAM (kB)	8	4		
Sensor Board Interface	**Type**	51 pin			18 pin
	10-Bit ADC	7, 0 V to 3 V input			6, 0 V to 3 V input
	UART	2			1
	Other interfaces	DIO, I2C			DIO
RF Transceiver (Radio)	Chip	RF230	CC2420	CC1000	
	Radio Frequency (MHz)	2400		315/433/915	
	Max. Data Rate (kbits/sec)	250		38.4	
	Antenna Connector	MMCX			PCB solder hole
Flash Data Logger Memory	Chip	AT45DB014B			
	Connection Type	SPI			
	Size (kB)	512			
Default power source	Type	AA, 2x			Coin (CR2354)
	Typical capacity (mA-hr)	**2000**			**560**

Figure A.1: Atmel Platforms compatible with TinyOS [52]

A.1.1 Atmel Platforms compatible with TinyOS

In Mica2Dot, Mica2 and Micaz motes, 8-bit Atmel ATMega128L MCU is used, while Iris has 8-bit Atmel ATMega1281 MCU. All motes with Atmel MCU have 128 KByte

program memory and 4KByte SRAM (Iris has 8kByte). Among these motes only Iris has a transceiver from Atmel, namely RF230, which works in the 2.4 GHz ISM band. Other platforms have transceivers from Texas Instruments (formerly Chipcon). Mica2Dot and Mica2 work in sub 1 GHz band, having Texas Instruments CC1000[31] transceiver, whereas Micaz works in the 2.4 GHz ISM band, having Texas Instruments CC2420[97] transceiver. Iris is the latest TinyOS mote from Crossbow that has three times improved radio range and twice the program memory respect to the previous Mica motes. Among the motes having Atmel MCU, next generation Iris is technically superior thanks to its 8 KByte SRAM and improved radio performance. In Fig. A.1, a summary of Atmel platforms compatible with TinyOS can be found.

A.1.2 Texas Instruments Platform compatible with TinyOS: TELOS

Telos mote has 16-bit Texas Instruments MSP430 MCU with 10 KByte RAM and 48 KByte program memory. It has Texas Instruments CC2420 transceiver like Micaz. Compared to all other TinyOS motes mentioned above, Telos platform delivers the lowest power consumption as well as the fastest MCU wake-up time from sleep state. In TinyOS-based solutions, Telos and Iris have better technical specifications but unfortunately device emulation in bit-level is only available for Mica2 and Mica2Dot motes, since TOSSIM and the other emulators such as Atemu[88] and Avrora[98], just support these motes only.

A.2 Other Solutions

On the other side of the medallion, there are international companies having 802.15.4-compatible software stack and devices. Since they are investing significant amount of resources for the development of their stack and 802.15.4 compatible products, they provide high quality software and professional tools to the developers. The general trend among the producers is to make available the object codes of their 802.15.4 stacks to developers free of charge. Ember, Freescale, ST Microelectronics, and Texas Instruments are the only promoter partners (2009) in ZigBee Alliance that are developing 802.15.4 compatible transceivers.

A.2.1 Ember and ST Microelectronics Platform

Ember and ST Microelectronics are strategic partners for ZigBee technology. ST denotes its transceiver as SN250[21] while Ember called it EM250[10]. In practice both transceivers are compatible in terms of features and performance. The most significant specification of these transceivers is the -99 dBm receiver sensitivity. Ember and ST provide EmberZNet[11] ZigBee Stack, developed by Ember to the third party developers. Ember does not provide a standalone 802.15.4 MAC stack for developing custom applications using 802.15.4.

A.2.2 Freescale Platform

Freescale simply calls its stack 802.15.4 MAC, and the object code is free of charge, in other words the use of this stack is free but its source code is not open. in June 2008 Freescale announced its next-generation 802.15.4 platform, namely MC13224V[62]. This new platform has impressive specifications like 32-bit ARM7 Core, 22mA receive, 29mA transmit, 0.85 μA hibernate currents at 3.3V battery operation, -100 dBm receiver sensitivity in NCD mode, 96 KByte RAM and 80 KByte ROM. 80 KByte ROM which contains bootcode.

A.2.3 Texas Instruments Platforms

Texas Instruments entered the ZigBee/802.15.4 market after the acquisition of Chipcon in 2006. Chipcon's CC2420 was the industry's first IEEE 802.15.4 compliant RF transceiver. System on chip solution CC2431[5] from Texas Instruments contains 8051 compliant MCU, CC2420 transceiver and a hardware location engine. Moreover, Texas Instruments has a next generation device, CC2520[6], which is a significantly improved transceiver, having +5dBm transmit power and excellent channel rejection values (49dBm at 5MHz, 54dBm at 10MHz). TIMAC [23] is Texas Instruments' 802.15.4 stack. TIMAC is distributed as object code free of charge.

	ST SN250 (Ember EM250)	TI CC2431	TI CC2520	Freescale MC13224V	Telos B (Crossbow)	IRIS (Crossbow)
MCU / Tranciever:	16 bit XAP2b / Custom	8 bit 32 MHz Intel 8051 / CC2420	No MCU / CC2520	32-bit TDMI ARM7 / Custom	TI 8 MHz 16-bit MSP430F1611/ CC2420	Atmel 8 bit 8.37 MHz Atmega 1281 MCU / AT86RF230 Transceiver
Flash Memory:	128KB	128 KB	N/A	128 KB	48KB	128 KB
Other Memory:	5KB RAM	8 KB RAM	N/A	96 KBSRAM 80 KB ROM	10KB RAM 16KB EEPROM	8KB SRAM 4KB EEPROM
Max Transmit Power:	+5dBm in Boost Mode	+ 0 dBm	+5 dBm	+4 dBm	+0 dBm	+3 dBm
Rx Current: (MCU Active)	35.5 mA (Vdd=3V, MCU 12 MHz)	26.7 mA (Vdd=3V, MCU running at 32 MHz)	18.5 mA (Low Power Mode, Vdd=3V, No MCU)	22mA (Vdd=3.3 V, MCU running at 2 MHz)	24,8mA (MCU Active)	24 mA (Vdd=3V, MCU Active)
Tx Current: (MCU Active)	32.8 mA (0dBm,Vdd=3V, MCU 12 MHz)	26.9 mA (0dBm,Vdd=3V, MCU 32 MHz)	25.8 mA (0dBm,Vdd=3V, No MCU)	29mA (0dBm,Vdd=3.3 V, MCU 2 MHz)		25 mA (+3dBm, Vdd=3V, MCU Active)
Sleep Current:	1uA max (with sleep timer running)	0.3 µA (No clocks. RAM retention)		0.85 µA typical Hibernate (Retain 8 Kbyte SRAM contents)	6.1 µA	8 µA
Rx Sensitivity:	-98 dBm (Boost Mode)	-92 dBm	-98 dBm	-100 dBm (NCD mode)	-94 dBm	-101 dBm
Channel Rejection: (Desired Signal -82 dbm)	35dBm(-5MHz) 35dBm(+5MHz) 40dBm(-10MHz) 40dBm(+10MHz)	30dBm(-5MHz) 45dBm(+5MHz) 53dBm(-10MHz) 54dBm(+10MHz)	49dBm(-5MHz) 49dBm(+5MHz) 54dBm(-10MHz) 54dBm(+10MHz)	38dBm(-5MHz) 38dBm(+5MHz) 57dBm(-10MHz) 57dBm(+10MHz)	38dBm(-5MHz) 47dBm(+5MHz)	34dBm(-5MHz) 36dBm(+5MHz) 52dBm(-10MHz) 53dBm(+10MHz)
Other:		Hardware Location Engine			TinyOS, 6LoWPAN, Fast MCU Wake up Time	TinyOS, 6LoWPAN

Figure A.2: Comparison of the Platforms

References

[1] 6lowpan - ipv6 over low power wpan. http://datatracker.ietf.org/wg/6lowpan/charter/. 167

[2] Aim - a novel architecture for modelling, virtualising and managing the energy consumption of household appliances. http://www.ict-aim.eu/. 138

[3] Ami-moses - ambient-intelligent interactive monitoring system for energy use optimisation in manufacturing smes. http://www.ami-moses.eu/. 138

[4] Beywatch - building energy watcher. http://beywatch.eu/. 138

[5] Cc2431 system-on-chip for 2.4 ghz zigbee / ieee 802.15.4 with location engine. Rev. 2.01 - SWRS034. 170

[6] Cc2520 datasheet 2.4 ghz ieee 802.15.4/zigbee rf transceiver. SWRS068 DECEMBER 2007. 170

[7] Crossbow. www.xbow.com. 168

[8] E3soho - energy efficiency in european social housing. http://www.e3soho.eu/. 138

[9] ediana project. http://www.artemis-ediana.eu/. 4, 137

[10] Em250 single-chip zigbee/802.15.4 solution. July 5, 2006. 170

[11] Emberznet application developers guide. 14 November 2008. 170

[12] Intube - intelligent use of buildingsenergy information. http://www.intube.eu/. 138

[13] Iris wireless measurement system. 6020-0124-01 Rev. 168

[14] Mica2 wireless measurement system. 6020-0042-0. 167

[15] Mica2dot wireless microsensor mote. 6020-0043-0. 167

[16] Micaz wireless measurement system. 6020-0060-04 Rev A. 168

[17] Remodece - residential monitoring to decrease energy use and carbon emissions in europe. http://remodece.isr.uc.pt/. 138

[18] S-mac software: Information and source code. http://www.isi.edu/ilense/software/smac/index.html. 167

[19] Senslab - very large scale open wireless sensor network testbed. http://www.senslab.info/. 125

[20] Smarthouse smartgrid project. http://www.smarthouse-smartgrid.eu/. 138

[21] Sn250 single-chip zigbee/802.15.4 solution. October 2007. 170

[22] Telos-b mote platform. 6020-0094-01 Rev A. 167

[23] Timac: Ieee802.15.4 medium access control (mac) software stack. http://www.ti.com/tool/timac. 102, 170

[24] Tinyos. http://www.tinyos.net/. 167

[25] Z-stack - zigbee protocol stack. http://www.ti.com/tool/z-stack. 102

[26] Sensor applications reference design (sard) users guide, 2006. 43

[27] 13192 evaluation board development kit (13192evb) users guide, 2007. 43

[28] Application note: Pcb design with em250, Apr. 2008. 87

[29] Chipcon cc2420 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver data sheet, 2008. SWRS041B. 80

[30] Mc13192 2.4 ghz low power transceiver for the ieee 802.15.4 standard, 2008. 43

[31] Chipcon cc1000 single chip very low power rf transceiver data sheet, 2009. 42, 55, 169

[32] G.D. Abowd and J.P.G. Sterbenz. Final report on the interagency workshop on research issues for smart environments. *IEEE Personal Communications*, pages 36–40, October 2000. 3

[33] J.T. Adams. An introduction to ieee std 802.15.4. In *Aerospace Conference, 2006 IEEE*, page 8 pp., 0-0 2006. 87

[34] I.F. Akyildiz, Y. Sankarasubramaniam W. Su, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002. 1, 2

[35] ZigBee Alliance. Zigbee cluester library specisfication, May 2008. 117

[36] ZigBee Alliance. Zigbee specification, Jan 2008. 13, 101, 103, 120

[37] A. Arora, E. Ertin, R. Ramnath, M. Nesterenko, and W. Leal. Kansei: a high-fidelity sensing testbed. *Internet Computing, IEEE*, 10(2):35 – 47, march-april 2006. 125

[38] E.J. Baghdady. Fm demodulator time-constant requirements for interference rejection. *Proceedings of the IRE*, 46(2):432 – 440, Feb. 1958. 55

[39] E.J. Baghdady. Theory of stronger-signal capture in fm reception. *Proceedings of the IRE*, 46(1):728 – 738, Apr. 1958. 55

REFERENCES

[40] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. 18:535–547, mar 2000. 43

[41] P. Bonnet, J. Gehrke, and P. Seshadri. Querying the physical world. *IEEE Personal Communications*, pages 10–15, October 2000. 3

[42] F. Borgonovo, M. Zorzi, L. Fratta, V. Trecordi, and G. Bianchi. Capture-division packet access for wireless personal communications. *IEEE Journal on Selected Areas in Communications*, 14(4):609 – 622, May. 1999. 55

[43] Stefan Bouckaert, Wim Vandenberghe, Bart Jooris, Ingrid Moerman, and Piet Demeester. The w-ilab.t testbed. 2010. 125

[44] C. Buratti. A mathematical model for performance of ieee 802.15.4 beacon-enabled mode. In *Proc. of IEEE IWCMC 2009*, Leipzig, Germany, Jun 2009. 42, 43, 46, 47, 48, 51, 52

[45] C. Buratti and R. Verdone. Performance analysis of ieee 802.15.4 non-beacon enabled mode. *IEEE Trans. Veh. Technol*, 58:3480–3493, 2009. 47, 155

[46] A. Cerpa, J. Elson, M. Hamilton, and J. Zhao. Habitat monitoring: application driver for wireless communications technology. In *ACM SIGCOMM'2000*, Costa Rica, April 2001. 3

[47] Ioannis Chatzigiannakis, Stefan Fischer, Christos Koninis, Georgios Mylonas, and Dennis Pfisterer. Wisebed: An open large-scale wireless sensor network testbed. In *Sensor Applications, Experimentation, and Logistics*, volume 29 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 68–87. Springer Berlin Heidelberg, 2010. 125

[48] Yin Chen and Andreas Terzi. On the mechanisms and effect of calibrating rssi measurements for 802.15.4 radios. Coimbra, Spain, Feb 2010. EWSN. 56, 59, 87, 88, 165

[49] Z. Chen, C. Lin, H. Wen, and H. Yin. An analytical model for evaluating ieee 802.15.4 csma/ca protocol in low rate wireless application. In *Proc. IEEE AINAW 2007*, 2007. 43

[50] Harshal S. Chhaya and Sanjay Gupta. Performance of asynchronous data transfer methods of ieee 802.11 mac protocol. *IEEE Personal Communications*, 3(5):8–15, Oct. 1996. 55

[51] European Commision. Directive 2002/91/ec on the energy performance of buildings. *Official Journal of the European Communities*, pages 65–71, Dec. 2002. 4, 137

[52] Crossbow. Mpr-mib users manual, June 2007. xii, 168

[53] A. Cunha, A. Koubaa, R. Severino, and M. Alves. Open-zb: an opensource implementation of the ieee 802.15. 4/zigbee protocol stack on tinyos. IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems, 2007. 167

[54] de Morais Cordeiro, C., Sadok, D., and D.P. Agrawal. Modeling and evaluation of bluetooth mac protocol. pages 518 – 522. Proceedings. Tenth International Conference on Computer Communications and Networks, Oct 2001. 55

[55] PC Engines. Alix.3c3 / alix.3d3 system boards, Apr. 2010. 127

[56] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01)*, volume 4, pages 2033–2036, Boston, MA, May 2001. 2

[57] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *MobiCom'99*, pages 263–270, Washington, USA, 1999. 2

[58] M. Ogawa et al. Fully automated biosignal acquisition in daily routine through 1 month. In *International Conference IEEE-EMBS*, pages 1947–1950, Hong Kong, 1998. 3

[59] Freescale Semiconductor. *1322x-Low Power Node Reference Manual*, rev. 1.2 edition, Sep 2008. 32

[60] Freescale Semiconductor. *1322x USB Dongle Reference Manual*, rev. 1.5 edition, Nov 2010. 32, 33

[61] Freescale Semiconductor. *Freescale 802.15.4 MAC PHY Software Reference Manual*, rev. 2.5 edition, May 2010. 101

[62] Freescale Semiconductor. *MC1322x Datasheet*, rev. 1.3 edition, September 2010. 29, 33, 43, 56, 127, 170

[63] Freescale Semiconductor. *MC1322x Reference Manual*, rev. 1.5 edition, November 2011. 31, 43, 155

[64] Freescale Semiconductor, Inc., Chandler, Arizona, USA. *Freescale 802.15.4 MAC PHY Software Reference Manual*, May 2009. Rev. 2.1. 31, 92

[65] Freescale Semiconductor, Inc., Chandler, Arizona, USA. *Freescale MC1322X Reference Manual*, May 2009. Rev. 1.2. 80

[66] Freescale Semiconductors. *1322x USB Dongle Reference Manual*, 2008. 43, 87

[67] C. Gezer, C. Buratti, and R. Verdone. Capture effect in ieee 802.15.4 networks: Modelling and experimentation. Modena, Italy, May 2010. Symposium on Wireless Pervasive Computing, ISWPC 2010. 155

[68] Priyanka Gupta and Stephen G. Wilson. Ieee 802.15.4 phy analysis: Power spectrum and error performance. Kanpur, India, Dec 2008. INDICON 2008. 81

[69] Vlado Handziski, Andreas Kopke, Andreas Willig, and Adam Wolisz. Twist: A scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. REALMAN06, May. 2006. 125

[70] Simon Haykin. *Communication Systems*. John Wiley Sons, 4th edition, 2001. 61, 69, 86

[71] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *proceedings of the ACM Mobicom'00*, pages 56–67, Boston, MA, 2000. 1

[72] Jae Hyun Kim and Jong Kyu Lee. Capture effects of wireless csma/ca protocols in rayleigh and shadow fading channels. *IEEE Transactions on Vehicular Technology*, 48(4):1277–1286, Jul. 1999. 55

[73] S. Lanzisera and K.S.J. Pister. Theoretical and practical limits to sensitivity in ieee 802.15.4 receivers. In *Electronics, Circuits and Systems, 2007. ICECS 2007. 14th IEEE International Conference on*, pages 1344 – 1347, dec. 2007. 87

[74] K. Leentvaar and J. Flint. The capture effect in fm receivers. *IEEE Transactions on Communications*, 24(5):531 – 539, May. 1976. 55

[75] Philip Levis and Nelson Lee. Tossim: A simulator for tinyos networks, Sep. 2003. 168

[76] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. NC, USA, Nov 2008. ACM SenSys. 56, 59, 87, 88, 165

[77] Ritesh Maheshwari, Shweta Jain, and Samir R. Das. On estimating joint interference for concurrent packet transmissions in low power wireless networks. CA, USA, Sep 2008. ACM MobiCom. 56

[78] Tatsuro Masamura, Shuichi Samejima, Yoshiteru Morihiro, and Hiroaki Fuketa. Differential detection of msk with nonredundant error correction. *IEEE Transactions on Communications*, COM-27(6):912–918, June 1979. 70

[79] ST Microelectronics. Spear600 datasheet, Feb 2010. 155

[80] J. Misic, S. Shafi, and V. B. Misic. Maintaining reliability through activity management in an 802.15.4 sensor cluster. 3:779–788, may 2006. 43

[81] Andreas F. Molisch. *Wireless Communications*. John Wiley and Sons, 2005. 33

[82] C. Namislo. Analysis of mobile radio slotted aloha networks. *IEEE Trans. Veh. Technol.*, VT-83:199–204, Aug 1984. 41, 55

[83] A. Nyandoro, L. Libman, and M. Hassan. Service differentiation using the capture effect in 802.11 wireless lans. *IEEE Transactions on Wireless Communications*, 6(8):2961–2971, Aug. 2007. 55

[84] Y. Onozato, J. Liu, and S. Noguchi. Stability of a slotted aloha system with capture effect. *IEEE Trans. Veh. Technol.*, 38:31–36, Feb 1989. 41, 55

[85] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, New York, NY, USA, 2002. 49, 65

[86] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza. Sensor-based information appliances. *IEEE Instrumentation and Measurement Magazine*, pages 31–35, December 2000. 3

[87] Joseph Polastre, Joseph Hill, and David Culler. Versatile low power media access for wireless sensor networks. Baltimore, Maryland, USA, Nov 2004. ACM SenSys. 167

[88] J. Polley, D. Blazakis, J. McGee, D. Rusk, and J.S. Baras. Atemu: a fine-grained sensor network simulator. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 145 – 152, oct. 2004. 169

[89] S. Pollin, M. Ergen, S.C. Ergen, B. Bougard, L. Van der Pierre, F. Catthoor, I. Moerman, A. Bahai, and P. Varaiya. Performance analysis of slotted carrier sense ieee 802.15.4 medium access layer. 7:3359–3371, sep 2008. 43, 155

[90] Lawrence G. Roberts. Aloha packet system with and without slots and capture. *ACM SIGCOMM Computer Communication Review*, 5(2), Apr. 1975. 55

[91] Freescale Semiconductor. Freescale beestack reference manual, Dec 2008. 148

[92] B. Sibbald. Use computerized systems to cut adverse drug events: report. *CMAJ: Canadian Medical Association Journal*, 164:1878, 2001. 3

[93] IEEE Computer Society. Ieee std 802.15.4-2006, Sep 2006. 13, 30, 44, 57, 101, 155, 156

[94] Dongjin Son, Bhaskar Krishnamachari, and John Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *International Conference on Embedded Networked Sensor Systems (SenSys 06)*, Boulder, Colorado, USA, May 2006. 42, 55, 56, 59, 87, 88, 165

[95] Mukundan Sridharan, Wenjie Zeng, William Leal, Xi Ju, Rajiv Ramnath, Hongwei Zhang, and Anish Arora. *KanseiGenie: Software infrastructure for resource management and programmability of wireless sensor network fabrics*. Springer. 125

[96] Kalyan Pathapati Subbu and Ivan Howitt. Empirical study of ieee 802.15.4 mutual interference issues. In *SoutheastCon, 2007*, Richmond, Virginia, USA, March 2007. 42, 58

[97] Texas Instruments. *CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee Ready RF Transciver*, 2006. 55, 87, 169

[98] B.L. Titzer, D.K. Lee, and J. Palsberg. Avrora: scalable sensor network simulation with precise timing. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 477 – 482, april 2005. 169

[99] T.O.Kim, J.S.Park, H.J.Chong, K.J.Kim, and B.D.Choi. Performance analysis of ieee 802.15.4 non-beacon mode with the unslotted csma/ca. *IEEE Communications Letters*, 12:238–240, Apr. 2008. 155

[100] M. Torrent-Moreno, S. Corroy, F. Schmidt-Eisenlohr, and H. Hartenstein. Ieee 802.11-based one-hop broadcast communications: understanding transmission success and failure under different radio propagation environments. In *MSWiM 06*, page 6877, 2006. 90

[101] Geoffrey Werner-Allen, Patrick Swieskowski, and Matt Welsh. Motelab: A wireless sensor network testbed. 2005. 125

[102] Kamin Whitehouse, Alec Woo, Fred Jiang, Joseph Polastre, and David Culler. Exploiting the capture effect for collision detection and recovery. In *IEEE Workshop on Embedded Networked Sensors (EmNetS-II)*, Sydney, Australia, May 2005. 42, 55

[103] ZigBee Alliance. *ZIGBEE SMART ENERGY PROFILE SPECIFICATION*, revision 15 edition, December 2008. 122

[104] ZigBee Alliance. *ZIGBEE HOME AUTOMATION PUBLIC APPLICATION PROFILE*, revision 26, version 1.1 edition, February 2010. 122