

UNIVERSITY OF PADOVA



FACULTY OF ENGINEERING

Department of Information Engineering  
Ph.D. School on Information Engineering

Information and Communication Science and Technologies  
XXII Cycle

Ph.D. Thesis

---

# **Cross layer analysis of interference effects in wireless systems**

**School Head:** Ch.mo Prof. Matteo Bertocco

**Supervisor:** Ch.mo Prof. Matteo Bertocco

**Ph.D. candidate:** Ing. Giovanni Gamba



*To my Family:  
Lisa and Daniele*



# *Contents*

<b>Contents</b>	<b>v</b>
<b>List of Figures</b>	<b>ix</b>
<b>Acronyms</b>	<b>xiii</b>
<b>Abstract</b>	<b>xvii</b>
<b>Sommario</b>	<b>xix</b>
<b>Introduction</b>	<b>xxi</b>
<b>1 Wireless Sensor Networks</b>	<b>1</b>
1.1 Elements of a WSN . . . . .	1
1.1.1 Fault tolerance . . . . .	2
1.1.2 Scalability . . . . .	2
1.1.3 Costs . . . . .	2
1.1.4 Operating environment . . . . .	2
1.1.5 Network topology . . . . .	2
1.1.6 Hardware constraints . . . . .	3
1.1.7 Radio channel . . . . .	4
1.1.8 Power consumption . . . . .	4
1.2 WSN: scenarios . . . . .	5
1.2.1 Environment . . . . .	5
1.2.2 Medical applications . . . . .	6
1.2.3 Military applications . . . . .	6
1.2.4 Domotics . . . . .	6
1.2.5 Automotive . . . . .	6
1.2.6 Factory automation . . . . .	6
<b>2 An introduction to the IEEE 802.15.4 standard</b>	<b>7</b>
2.1 IEEE 802.15.4 standard for WPAN networks . . . . .	8
2.1.1 WPAN components . . . . .	8
2.2 Overview of the main services . . . . .	8

2.2.1	Energy Detection (ED)	10
2.2.2	Link Quality Indication (LQI)	11
2.2.3	Clear Channel Assessment (CCA)	11
2.2.4	PPDU description	12
2.3	MAC Layer	12
2.3.1	The CSMA/CA algorithm	13
2.3.2	MAC frame	16
2.3.3	Addressing	17
2.4	PHY layer	18
2.4.1	Transmitter functional blocks	18
2.4.2	Receiver performance	22
<b>3</b>	<b>Tmote Sky e TinyOS</b>	<b>25</b>
3.1	Tmote Sky	25
3.1.1	Power supply	27
3.1.2	Microprocessor	27
3.1.3	Sensors	28
3.1.4	Expansion Connectors	29
3.1.5	Radio	30
3.2	Operating System: TinyOS	31
3.2.1	TinyOS: general overview	32
<b>4</b>	<b>Adopted software</b>	<b>33</b>
4.1	Factory Sniffer: Industrial monitoring	33
4.1.1	General description	34
4.1.2	Packet structure	36
4.1.3	GUI and main features	37
4.2	Power Meter	41
4.2.1	Main features	41
4.2.2	GUI and data format	42
<b>5</b>	<b>Wireless Sensor Network (WSN) based industrial control system</b>	<b>43</b>
5.1	High Layer Monitoring protocol	44
5.1.1	Cyclic task	44
5.1.2	Acyclic task	48
5.2	Measurement System and Setup	48
5.3	Cyclic task experiments	49
5.3.1	Without interference	50
5.3.2	With interference	52
5.4	Acyclic task experiments	57
5.4.1	Without interference	57
5.4.2	With interference	59
5.5	Conclusions	61

<b>6</b>	<b>Optimizing a WSN based wireless control system against interference</b>	<b>63</b>
6.1	Standards Overview . . . . .	63
6.1.1	IEEE 802.15.1 standard . . . . .	64
6.1.2	IEEE 802.11 standard . . . . .	65
6.2	Industrial Monitoring protocol . . . . .	65
6.2.1	Cyclic task . . . . .	65
6.2.2	Acyclic task . . . . .	68
6.3	Measurement System and Setup . . . . .	68
6.4	Experimental Results . . . . .	70
6.4.1	Packet Error Rate . . . . .	70
6.4.2	Polling Cycle Time . . . . .	72
6.5	Polling Latency . . . . .	74
6.5.1	Alarm Latency . . . . .	77
6.6	Conclusions . . . . .	79
<b>7</b>	<b>Cross layer CSMA/CA modeling</b>	<b>81</b>
7.1	Theoretical background . . . . .	82
7.1.1	PHY layer interference . . . . .	82
7.1.2	MAC layer interference . . . . .	84
7.1.3	Multi-link network . . . . .	85
7.2	Proposed Model . . . . .	86
7.2.1	SI-chart: fixed threshold . . . . .	87
7.2.2	ITH-chart: fixed signal power . . . . .	88
7.2.3	STH-chart: fixed interference power . . . . .	90
7.3	Numerical results . . . . .	91
7.3.1	Two-dimensional analysis . . . . .	91
7.3.2	Three-dimensional analysis . . . . .	94
7.4	Experimental Validation . . . . .	95
7.4.1	Testbed . . . . .	95
7.4.2	Measurement Results . . . . .	97
7.5	Conclusions . . . . .	100
<b>8</b>	<b>Assessing of interference effects on RSSI</b>	<b>101</b>
8.1	Preliminary notes . . . . .	102
8.1.1	Low-IF receiver architecture . . . . .	102
8.1.2	RSSI architecture . . . . .	104
8.1.3	RSSI common applications . . . . .	105
8.2	Out-of-channel interference . . . . .	106
8.2.1	IEEE 802.15.4 receiver architecture . . . . .	106
8.2.2	Receiver frequency response . . . . .	107
8.2.3	Simulation . . . . .	109
8.2.4	Testbed . . . . .	111
8.2.5	Experimental results . . . . .	113
8.3	Impairments on RSSI . . . . .	115

---

8.3.1	Measurement setup . . . . .	115
8.3.2	Analysis of the RSSI meter frequency response . . . . .	116
8.3.3	Interference effects on RSSI readings . . . . .	118
8.4	Conclusions . . . . .	121
<b>9</b>	<b>RSSI based RF power measurement</b>	<b>123</b>
9.1	Single input approach . . . . .	123
9.1.1	Calibration step . . . . .	124
9.1.2	Correction step . . . . .	126
9.2	Experimental verification . . . . .	127
9.2.1	Adopted testbed . . . . .	127
9.2.2	Experimental results . . . . .	129
9.3	Conclusions . . . . .	132
9.4	Multiple input improvement . . . . .	132
9.4.1	Proposed approach . . . . .	133
9.4.2	Simulation results . . . . .	135
9.4.3	Testbed . . . . .	135
	<b>Conclusions</b>	<b>139</b>
<b>A</b>	<b>Wireless System for ElectroMagnetic Area Notice (WISE MAN)</b>	<b>141</b>
A.1	Proposed Approach . . . . .	142
A.2	Implementation . . . . .	144
	<b>Bibliography</b>	<b>145</b>



## *List of Figures*

1.1	WSN network topologies. . . . .	3
2.1	ZigBee network topologies. . . . .	9
2.2	IEEE 802.15.4 bands and transmission rates. . . . .	9
2.3	A representation of the spectrum used in this standard in the ISM band. . . . .	10
2.4	The PPDU structure in the IEEE 802.15.4 standard. . . . .	12
2.5	The beaconless CSMA/CA algorithm described in the IEEE 802.15.4 standard. . . . .	15
2.6	The superFrame structure in the IEEE 802.15.4 standard. . . . .	16
2.7	The general structure of a MAC frame in the IEEE 802.15.4 standard. . . . .	17
2.8	The structure of a data MAC frame in the IEEE 802.15.4 standard. . . . .	17
2.9	Block diagram of IEEE 802.15.4 modulator. . . . .	18
2.10	O-QPSK constellation scheme in PHY layer of the IEEE 802.15.4 standard. . . . .	19
2.11	In-phase and Quadrature components of a O-QPSK modulator . . . . .	20
2.12	Normalized modulation filter shape. . . . .	20
2.13	Simulation of the modulator output signals. . . . .	21
2.14	Normalized spectrum mask of a half-sine filter. . . . .	22
2.15	BER performance of various standards operating in the 2.4 GHz ISM band. . . . .	23
3.1	Front and Back of the Tmote Sky module. . . . .	26
3.2	Operating conditions. . . . .	27
3.3	Blocks diagram. . . . .	28
3.4	10-pin connector. . . . .	29
3.5	6-pin connector. . . . .	29
3.6	Output power configuration for the CC2420. . . . .	30
3.7	RSSI <i>vs</i> input power [dBm]. . . . .	30
3.8	Functional blocks of a CC2420 transceiver. . . . .	31
4.1	Packets exchange between master and slave inside a polling time. . . . .	34

4.2	Polling and alarm packets inside a polling time. . . . .	35
4.3	Factory Sniffer GUI. . . . .	38
4.4	Power Meter GUI. . . . .	42
5.1	Time diagram of master and slave communication. . . . .	45
5.2	Testbed architecture in the case of ten slaves ( $N = 10$ ). . . . .	49
5.3	Measured $PRTT$ probability density function. . . . .	50
5.4	Measured ECT probability density function. . . . .	51
5.5	Percentage of failed polling, $\Phi_p$ , vs received interference power $P_i$ . . . . .	52
5.6	Percentage of failed pollings, $\Phi_p$ , versus burst duty cycle, 3 slaves. . . . .	53
5.7	Measured $\overline{PRTT}$ probability density function vs $\lambda_B$ and for: (a) $T_B = 100$ ms ; (b) $T_B = 300$ ms. . . . .	54
5.8	Measured $\overline{ECT}$ probability density function vs $\lambda_B$ and for: (a) $T_B = 100$ ms ; (b) $T_B = 300$ ms. . . . .	56
5.9	Measured alarm latency probability density function ( $N = 3$ , mean inter-arrival time = 1000 ms). . . . .	58
5.10	Alarm latency vs number of slaves. . . . .	58
5.11	Percentage of failed pollings, $\Phi_p$ , versus burst duty cycle, 3 slaves. . . . .	59
5.12	Measured alarm latency probability density function vs $\lambda_B$ and for: (a) $T_B = 100$ ms ; (b) $T_B = 300$ ms. . . . .	60
6.1	Frequency channels of (a) IEEE 802.15.4, (b) IEEE 802.11 and (c) IEEE 802.15.1 inside the ISM band. . . . .	64
6.2	Packets exchange between master and slave inside a polling time. . . . .	66
6.3	Testbed architecture. . . . .	69
6.4	Measured PER for all the considered configurations. . . . .	71
6.5	Probability density function for Polling cycle ( $C_p$ ) with no interference. . . . .	73
6.6	Probability density function for Polling cycle ( $C_p$ ) with WiFi interference. . . . .	73
6.7	Probability density function for Polling cycle ( $C_p$ ) with ZigBee interference. . . . .	74
6.8	Probability density function for Polling latency ( $L_p$ ) with no interference. . . . .	75
6.9	Probability density function for Polling latency ( $L_p$ ) with WiFi interference. . . . .	76
6.10	Probability density function for Polling latency ( $L_p$ ) with ZigBee interference. . . . .	76
6.11	Probability density function for Alarm latency ( $L_a$ ) with no interference. . . . .	77
6.12	Probability density function for Alarm latency ( $L_a$ ) with WiFi interference. . . . .	78
6.13	Probability density function for Alarm latency ( $L_a$ ) with ZigBee interference. . . . .	78
7.1	PER versus SINR and $P_i$ : (a) at physical layer, (b) at MAC layer. . . . .	83

7.2	Theoretical and approximated PER for IEEE 802.15.4 systems. . .	84
7.3	SI chart: modeling $PER_k$ in a CSMA/CA-based wireless network.	87
7.4	ITH chart: modeling $PER_k$ in a CSMA/CA-based wireless network.	89
7.5	STH chart: modeling $PER_k$ in a CSMA/CA-based wireless network. . . . .	90
7.6	SI chart. . . . .	92
7.7	ITH chart. . . . .	93
7.8	STH chart. . . . .	94
7.9	Simulation results, 3D representation. . . . .	95
7.10	Adopted testbed. . . . .	96
7.11	Experimental results: $PER$ versus $P_i$ with $P_s = -25$ dBm and different CCA thresholds, $P_{TH}$ . . . . .	97
7.12	Experimental results: $PER$ versus $P_i$ with $P_s = -50$ dBm and different CCA thresholds, $P_{TH}$ . . . . .	98
7.13	Experimental results: $PER$ versus $P_i$ with $P_s = -75$ dBm and different CCA thresholds, $P_{TH}$ . . . . .	98
8.1	Architecture of a low-IF receiver. . . . .	103
8.2	Architecture of a wireless receiver with RSSI stage. . . . .	105
8.3	Simplified architecture of a IEEE 802.15.4 receiver. . . . .	107
8.4	Influence of out-of-channel interference within the WSN bandwidth. . . . .	108
8.5	Out-of-channel interference attenuation due to receiver's filters. .	109
8.6	RSSI estimates (simulated $P_{I_y}$ ) vs interference offset for different interference types. . . . .	110
8.7	RSSI and PER (simulation) vs interference offset for single tone and 10 MHz-bandwidth interference. . . . .	111
8.8	Testbed of the simplified analysis scenario. . . . .	112
8.9	RSSI estimates (measured) vs interference offset and for different interference types. . . . .	114
8.10	RSSI and PER measurement vs interference offset for single tone and 10 MHz-bandwidth interference. . . . .	114
8.11	Estimated RSSI vs frequency offset $\Delta f$ . . . . .	117
8.12	Estimated RSSI vs $P_s$ , for three different frequency offsets: $\Delta f = 0, +2, -2$ MHz. . . . .	118
8.13	RSSI reading error due to interference. . . . .	119
8.14	RSSI reading affected by interference for: (a) $\Delta f = +2$ MHz (b) $\Delta f = -2$ MHz. . . . .	120
9.1	Example of RSSI transcharacteristic: a) ideal, b) non-ideal invertible, c) non-ideal non-invertible. . . . .	124
9.2	Piece-wise-linear transcharacteristic. . . . .	125
9.3	Adopted testbed. . . . .	128
9.4	$\hat{p}$ vs $p$ before and after correction. . . . .	130
9.5	Error trend before and after correction. . . . .	130
9.6	$p$ vs $\hat{p}$ with error bars, before and after the adopted compensation.	131

9.7	Multiple input approach. . . . .	133
9.8	Example of a transcharacteristic: (a) single input approach, (b) multiple input approach. . . . .	136
9.9	Cost function, $\lambda(p)$ (dB scale used) for: (a) single input estimation, (b) multiple input estimation. . . . .	137
9.10	Blocks diagram of a typical digital controlled attenuator. . . . .	138
9.11	Modified testbed for multiple input estimation. . . . .	138
A.1	Logical layout of the proposed system. . . . .	143

## *Acronyms*

<b>ACK</b>	Acknowledgment
<b>ADC</b>	Analog to Digital Converter
<b>AES</b>	Advanced Encryption Standard
<b>AP</b>	Access Point
<b>ARQ</b>	Automatic Retransmission Query
<b>AWG</b>	Arbitrary Waveform Generator
<b>AWGN</b>	Additive White Gaussian Noise
<b>BER</b>	Bit Error Rate
<b>CAP</b>	Contention Access Period
<b>CCA</b>	Clear Channel Assessment
<b>CFP</b>	Contention Free Period
<b>COTS</b>	Commercial Off-The-Shelf
<b>CSMA/CA</b>	Carrier Sense Multiple Access/Collision Avoidance
<b>CSMA</b>	Carrier Sense Multiple Access
<b>DAC</b>	Digital to Analog Converter
<b>DSSS</b>	Direct-Sequence Spread Spectrum
<b>ED</b>	Energy Detection
<b>EIRP</b>	Equivalent Isotropically Radiated Power
<b>FCF</b>	Frame Control Field
<b>FCS</b>	Frame Check Sequence

---

<b>FDMA</b>	Frequency Division Multiple Access
<b>FEC</b>	Forward Error Correction
<b>FFD</b>	Full-Function Device
<b>FHSS</b>	Frequency Hopping Spread Spectrum
<b>FIFO</b>	First In First Out
<b>GPIO</b>	General Purpose Interface Bus
<b>GTS</b>	Guaranteed Time Slots
<b>GUI</b>	Graphical User Interface
<b>IEEE</b>	Institute of Electrical & Electronics Engineers
<b>IP</b>	Internet Protocol
<b>ISM</b>	Industrial, Scientific and Medical
<b>ISO</b>	International Standard Organization
<b>LNA</b>	Low Noise Amplifier
<b>LOS</b>	Line of Sight
<b>LQI</b>	Link Quality Indication
<b>LSB</b>	Least Significant Bit
<b>MAC</b>	Medium Access Control
<b>MHR</b>	MAC Header
<b>MLME</b>	MAC Layer Management Entity
<b>MPDU</b>	MAC PDU
<b>NET</b>	Network Layer
<b>O-QPSK</b>	Offset QPSK
<b>OS</b>	Operating System
<b>OSI</b>	Open System Interchange
<b>PAN</b>	Personal Area Network
<b>PC</b>	Personal Computer

---

<b>PDF</b>	Probability Density Function
<b>PDU</b>	Protocol Data Unit
<b>PER</b>	Packet Error Ratio
<b>PHR</b>	Physical Header
<b>PHY</b>	Physical Layer
<b>PLME</b>	Physical Layer Management Entity
<b>PN</b>	Pseudo-Noise
<b>PPDU</b>	Packet Protocol Data Unit
<b>PSD</b>	Power Spectral Density
<b>PSU</b>	Protocol Service Unit
<b>QPSK</b>	Quadrature Phase Shift-Keying
<b>RFD</b>	Reduced-Function Device
<b>RSSI</b>	Received Signal Strength Indication
<b>SFD</b>	Synchronization Frame Delimiter
<b>SHR</b>	Synchronization Header
<b>SINR</b>	Signal To Interference plus Noise Ratio
<b>SMA</b>	SubMiniature version A
<b>SNR</b>	Signal to Noise Ratio
<b>SN</b>	Sequence Number
<b>SOAC</b>	System On A Chip
<b>TCP</b>	Transport Control Protocol
<b>TCP/IP</b>	Transport Control Protocol/Internet Protocol
<b>TDMA</b>	Time Division Multiple Access
<b>USB</b>	Universal Serial Bus
<b>WLAN</b>	Wireless LAN
<b>WPAN</b>	Wireless PAN
<b>WSN</b>	Wireless Sensor Network





## *Abstract*


**W**IRELESS communication systems are nowadays employed in many fields. The research effort beneath this work is to investigate one of the main issues in a radio communication system: interference. Devices operating in the 2.4 GHz unlicensed Industrial, Scientific and Medical (ISM) band are considered and, in particular, wireless sensors networks compliant to IEEE 802.15.4 standard are used to evaluate performance indices in the presence of interference.

The analysis starts from a real wireless control application and a RF power meter application: a complete perspective involving theoretical formulas, simulations and experimental results is given. A cross layer approach for CSMA/CA based devices, merging interference effects at PHY and MAC layer, is presented. Subsequently effects of interference on channel status evaluation and RSSI circuitry are faced and, finally, the feasibility of a low cost RF power meter is discussed.

The purpose of the work is to understand interference phenomena, using a real life test bed and a complex set of interferers comprising both arbitrary signal generators and real wireless networks. The interaction of interference with a device is manifold: it involves physical layer, hardware and protocols. The final aim of this work is to provide protocol and hardware designers with a set of performance metrics and perspectives useful to improve wireless devices' behavior against interference.



## *Sommario*

GGIORNO i sistemi di comunicazione wireless hanno un impiego assai vasto. Questo lavoro di ricerca si propone di investigare una delle maggiori problematiche per una comunicazione radio: l'interferenza. Vengono considerati alcuni dispositivi operanti nella banda libera ISM a 2.4 GHz e, in particolare, vengono utilizzate reti di sensori wireless aderenti allo standard IEEE 802.15.4 per la valutazione di indici di prestazioni in presenza di interferenza.

L'analisi parte da un'applicazione di controllo wireless e da una applicazione per la misura della potenza a radiofrequenza. Viene di seguito presentato, per i dispositivi basati su CSMA/CA, un approccio cross-layer per un'analisi congiunta degli effetti delle interferenze a livello PHY e a livello MAC. Di seguito vengono affrontati gli effetti dell'interferenza sulla valutazione dell'occupazione di canale e sulla circuiteria di RSSI e, infine, viene discussa la fattibilità di un misuratore di potenza RF a basso costo.

Lo scopo della tesi è comprendere i fenomeni d'interferenza utilizzando un test bed reale e un complesso sistema di interferenze sia provenienti da generatori di segnale sia da reti wireless reali. L'interazione dell'interferenza con un dato dispositivo è molteplice: coinvolge il livello fisico, l'hardware e i protocolli. L'obiettivo finale è fornire ai progettisti dell'hardware e dei protocolli un insieme di metriche di prestazione e prospettive utili a migliorare il comportamento dei sistemi wireless contro l'interferenza.



## *Introduction*

**I**N the design and deployment of modern wireless communication networks, interference is one of the most critical problems to be carefully considered and mitigated. In fact, it can be responsible for severe degradation effects like loss of data packets, reduced throughput, delay, jitter, loss of synchronization, missed alarms, etc. A critical frequency band in which interference phenomena frequently occur is the unregulated and unlicensed 2.4 GHz ISM band (2.4 - 2.4835 GHz). In this band, several wide spread communication standards and applications may operate such as, for instance, IEEE 802.11b/g Wireless LAN (WLAN) [1], IEEE 802.15.4 [2], and IEEE 802.15.1 (Bluetooth) compliant devices, microwave ovens, etc. This clearly increases the possibility of in-band interference among wireless devices when deployed in the same environment.

The analysis and mitigation of interference problems is typically a complex task, to be carried out at design or deployment level. To this aim, an in-depth knowledge of interference phenomena effects on some network performance parameters is needed. The analysis is even more difficult in the case of wireless networks employing the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol, such as for instance the WLANs and IEEE 802.15.4 WSNs. In these cases interference can impair the system degrading the signal quality or preventing transmission at all.

Such impairments are even more problematic in a wireless control system, where almost all data are time-critical and data loss must be very limited in order to keep the whole system under control.

A very weak point in modern CSMA/CA based devices is the Received Signal Strength Indication (RSSI) circuitry, responsible of measuring the in channel power and hence evaluating if the channel is good for transmission or not. In-channel interference may be deleterious to this chip because it can completely paralyze transmission. Furthermore, even interference situated in adjacent channels can impair the system, since it can saturate the RSSI circuitry leading to a corrupted measure.

Such a sub-system can be greatly improved becoming a low cost measurement chip (with sufficient accuracy and precision). In the following a brief outline of the chapters of this thesis is provided.

**Chapter 1** This Chapter is a general purpose introduction on WSN. It depicts elements and scenarios useful for a beginner reader who has never worked on such a subject.

**Chapter 2** In this Chapter the IEEE 802.15.4 standard is sketched, underlining only few important keys useful to understand the rest of the thesis. In this Chapter many acronyms are explained.

**Chapter 3** The hardware platforms and the used software are described. This chapter is not mandatory for the comprehension of the thesis, but completes the background.

**Chapter 4** This is the starting point for the body of the thesis. The software created and tested is presented, with a special care to the description of the metrics and outputs useful for the analysis described in the following.

**Chapter 5** An in dept analysis of a wireless control system, consisting of theoretical and practical results is here discussed. This system is used as a basis even for other following chapters.

**Chapter 6** The system described in Chapter 5 is used to optimize the choice of several CSMA/CA parameters to defeat real interfering sources.

**Chapter 7** A thorough model for the joint description of Physical Layer (PHY) and Medium Access Control (MAC) layer interference is presented. Both simulations and experimental results are reported.

**Chapter 8** In this Chapter RSSI circuitry is introduced and effects of interference on it is analyzed through an extensive set of experiments.

**Chapter 9** The feasibility of an integrated low cost RF power meter is here discussed. A single input estimation is described and a multiple input approach is sketched.

**Appendix** A vision of a possible future implementation of concepts illustrated in Chapters 8 and 9 is presented.

# Chapter 1

## *Wireless Sensor Networks*

**T**HE concept of “WSN” has emerged about ten years ago and, after becoming an hot research topic in many universities all over the world, now is an actual application field. Many companies produce WSN applications and devices and the industry is trying to exploit all the means this technology can provide.

This kind of network is constituted by different small devices able to communicate with low data rate in the range of tenths of meter, or, in other words, they can be classified as Personal Area Network (PAN).

### 1.1 Elements of a WSN

The word “sensor” means a a device able to output a voltage or current dependent on the input physical quantity (i.e. light, temperature and so on). Thanks to the development of digital signal processing, analog signal can be digitized and elaborated so that the modern concept of sensor must be viewed in a wider sense. WSN are made up of smart sensor, where the real sensor is only a little part of the device. These devices are often called *nodes* (meaning “small particles” in old english). The protocol stack is typically much more lightweight than the one of a standard network, even if in the last few month a modified version of a TCP/IP stack tailored for WSN, called *6LowPAN*, has been developed [3]. The planning of a WSN should take into account various technological aspects that can help the designer to find the most suitable technology to fulfill the project. Some of these aspects are: fault tolerance, scalability, costs, operating environment, network topology, hardware constraints, transmission medium and finally power consumption. In the following paragraph a detailed view of each one is proposed.

### 1.1.1 Fault tolerance

In electronics devices failures can always occur, especially in an hostile environment such as desert or battle fields. The WSN connectivity should be maintained also with a small number of node failures though an auto-configuration of the remaining nodes. Different environment require different level of fault tolerance. For example, in medical or industrial applications an higher tolerance is needed respect to a simple environmental monitoring.

### 1.1.2 Scalability

An high number of nodes coexisting in a given area and transmitting into the same channel could cause interference to each other. On the other hand having a big number of sensors can improve the overall network performance through cooperative algorithms. The network scalability is a very important parameter to take into account in a design stage, because it affects both protocol behavior and hardware.

### 1.1.3 Costs

A typical WSN implies the use of many nodes: only if the cost per node is low enough these kind of networks can be effectively deployed. The cost of a single node is ruled by the sensors and transducers, while the transceiver and the microprocessor, now integrated in a System On A Chip (SOAC) solution, are getting cheaper.

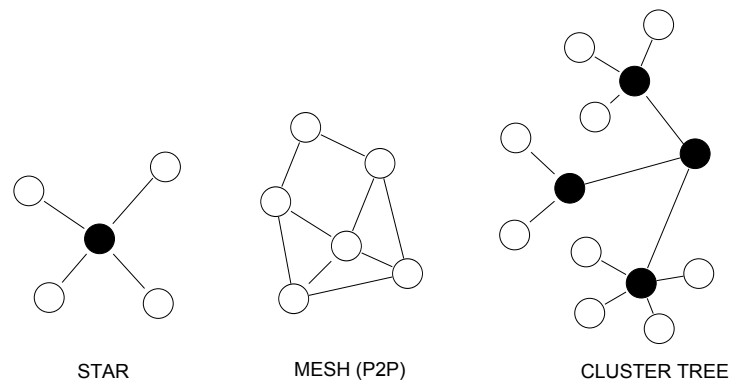
### 1.1.4 Operating environment

The design of a node must consider the operating environment in which it will operate. Obviously a military application needs components and packages far more resistant than a node tailored for domotics.

### 1.1.5 Network topology

Network topology can be viewed from two perspective: physical position of the nodes and logical communication topology. Concerning the nodes physical placement it is worth noting that a wireless communication system allows a much more flexible layout than a cabled system. This layout can be varied by the motion of the nodes or, in other cases, due to the failure of several nodes. It is more useful to define the topology from a logical communication point of view. Network topologies can be roughly classified into three schemes: the *star*, the *mesh* or *peer-to-peer* and the *tree* topologies.





**Figure 1.1:** WSN network topologies.

In Figure 1.1 a sketch of the above mentioned schemes is provided.

**Star** There is a central node defined as controller. All the communications between nodes must be relayed by this central node. This topology can be managed with simple protocols and is mainly employed when the central node is not energy-constrained, being connected to a fix power supply. The controller is often a gateway to connect the WSN to a wired backbone.

**Mesh (P2P)** In this kind of network there is not a central coordinator and each node can act as a relay for other nodes. The routing protocol assumes a great importance in such a topology, because it assures the network connectivity through a multi-hop approach.

**Tree** In the tree topology the nodes are grouped into *cluster* which are ruled by a central coordinator. This is an hybrid solution that works with simpler routing algorithms and less communication paths than the mesh topology.

### 1.1.6 Hardware constraints

Every wireless sensor node has these sub-systems:

**Sensor system** All the sensors and conditioning hardware, comprising the Analog to Digital Converter (ADC) used to interface the analog input with a microcontroller and a digital transmission system.

**Control system** It comprises the communication buses, the memories (typically Flash) and the microprocessor whose target is the overall system management.

**Communication system** The communication can exploit radio waves or infrared optical carriers. The infrared system needs the Line of Sight (LOS) while a radio system is much more flexible and efficient also from an energetic point of view.

**Power supply system** Normal batteries are the most common power supply for sensor nodes, but also other *energy scavenging* systems, such as solar panels, are important to collect energy from the environment. Due to the limits of energy in battery-powered systems, a great importance is to be paid to low-power microprocessors and transmitting protocols.

### 1.1.7 Radio channel

The most suitable operating bands are the ISM bands. These bands are unlicensed ones; the 2.4 GHz band is free almost all over the world. This band allows a good degree of integration in the radio subsystem and leaves a great freedom in the design of the communication protocol. The main constraints are in the maximum Equivalent Isotropically Radiated Power (EIRP) that must be below 20 dBm.

### 1.1.8 Power consumption

As above described, energy is limited and hence is one of the most important constraints. To this aim it is mandatory the use of power aware protocols whose main metric is the available energy. The target of a wireless sensor is the acquisition, the elaboration and finally the transmission of an information.

The acquisition power consumption is mainly dependent on the type of sensor: for example gas sensor have to be warmed-up and this is a further power dissipation.

The elaboration phase is quite power saving, because only a little pre-elaboration is done on-board, while the main part is done after the data has been collected by a central processing unit.

The communication phase is the most power-consuming, both in the transmitting and in the receiving stages.

## 1.2 WSN: scenarios

Several appealing application fields can exploit the WSN best features:

**Monitoring** This kind of network is employed to track a certain physical quantity that is sampled with a fixed frequency. It can be a very energy-consuming task, especially if the sampling period is small.

**Event detection** The network is employed to detect several alarm behaviors. This is typically an asynchronous task, with low energy footprint, being the nodes in an idle state for long times.

**Object classification** The goal of these networks is the object recognition in a pre-determined data set. This could be a very computing-intensive task.

**Object tracking** this kind of network is essentially a smart surveillance system. The network should recognize an object and localize it.

The most interesting application scenarios are the following.

### 1.2.1 Environment

Environmental monitoring is one of the most appealing application field for WSN. These networks can help creating a distributed monitoring system for environmental pollution (e.g near a chemical plant) or can be a good way to control volcanoes or other dangerous natural phenomena.

Agriculture and botanic research can benefit of WSN, because natural climate and animal issues can be caught in real time. Environmental monitoring of desert soils or woods can exploit self configuring networks, even deployed from an airplane in a pseudo-random fashion. In this case the most important problems to be faced are the self configuring network protocols (e.g dynamic routing and energy saving ones) and energy scavenging mechanisms. Data traffic (throughput) is not a key issue because only sporadic and low-bandwidth transmission are expected, being natural phenomena slow enough to fit with hourly or daily sampling.

The most suitable network topology for these applications is the Ad-Hoc topology, that is a non-infrastructure network. The network connectivity should be meshed, allowing a multi-hop communication an a certain degree of redundancy.

### 1.2.2 Medical applications

Biomedical monitoring is one of the most critical application. In this field WSN can help monitoring physiological parameters in a flexible and non-invasive manner. By now only the first *personal health-care* devices, integrated and wirelessly connected are available, but a future vision is a mesh of micro sensor that can monitor the body from inside.

### 1.2.3 Military applications

Embedded and wearable sensor could improve technical military garment, being able to detect a chemical or biological hazard or detecting a soldier's illness or wound. Robust network protocols and technologies (such as spread spectrum modulation) have been typically developed for the stringent military needs. The redundancy of a mesh topology can be a viable solution to guarantee the connectivity even in an hostile and disturbed battle field.

### 1.2.4 Domotics

A commercial scenario, nowadays already developed, is domotics. The interconnection of different appliances can improve the overall home automation and energy saving. A trivial yet useful application is the wireless switch: without holes in the walls a new lamp can be installed. The main drawback is the cost, that should be very low to let this technology rise up.

### 1.2.5 Automotive

In the automotive context wireless sensor can be efficiently used to reduce cabling costs. Other applications are intended for safety, such as navigation aids, but the low reliability of wireless links still keep the WSN away from this applications.

### 1.2.6 Factory automation

Factory automation needs very tight real-time and reliability bounds. Network topology, on the other hand, are typically hierarchical and almost static, limiting the complexity of routing algorithms. Wireless subnets are connected to cabled backbones via gateways, so in this context the coexistence among different wireless networks and wireless-wired coexistence is an important issue. In the case of a wireless control system all these issues must be carefully taken into account.

# Chapter 2

## *An introduction to the IEEE 802.15.4 standard*

**I**EEE 802.15.4 and ZigBee are often confused, while there is a clear difference among them. ZigBee Alliance is an industrial and academic partnership founded about ten years ago, to supply a low-cost, low-power, wireless mesh networking standard to be used in automation and remote control application. The task group 4 of the Institute of Electrical & Electronics Engineers (IEEE) 802.15 committee started to follow the same arguments, releasing a standard, the IEEE 802.15.4, on may 2003. Nowadays the ZigBee specifies a suite of high level communication protocols using small, low-power digital radios based on the IEEE 802.15.4 standard for Wireless PAN (WPAN). ZigBee supplies the higher layers (from Network Layer (NET) up to Application layer) while the IEEE 802.15.4 defines MAC and PHY layers. The main characteristics of this suite are:

- low bit-rate;
- low costs;
- low power consumption.

The target of this type of network is different from other wireless standards already present on the market, like Bluetooth and WiFi. IEEE 802.15.4 (with ZigBee) allows networks covering a large area. Nodes usually are intended to have an autonomy of years. The typical communication range spans from 10 to 75 meters, depending on the environment. Greater distances can be reached using multi-hop techniques, i.e. several nodes serve as “relays” for other nodes, collecting some transmission frames and forwarding them through specific links.

## 2.1 IEEE 802.15.4 standard for WPAN networks

In many fields there is no need for an high throughput (net traffic transmitted on the network, without considering header and retransmissions) or a wide coverage area. Often it is necessary that nodes can communicate each other on a plant, to transmit measurements of some interest. Thus this wireless sensors do not need high bit-rate but a long operating life, little dimensions and low cost, and usually they can not support common protocols stack like Transport Control Protocol/Internet Protocol (TCP/IP), due to the heavy needs in terms of memory and energy (apart from modified version of the stack [3]). The IEEE 802.15.4 standard provides regulations for such devices, and specifies how their MAC and PHY layers are composed.

### 2.1.1 WPAN components

In the standard two types of devices are defined: Full-Function Device (FFD) and Reduced-Function Device (RFD). The former has all the functionalities defined in this standard, the latter implements only a reduced set of them. Every network must have at least one FFD that represents the *coordinator* of all other devices in the network.

FFD devices can communicate with both types of devices, while a RFD cannot talk to a FFD. A FFD node can operate in three different manners:

- PAN coordinator;
- Simple coordinator;
- Normal device.

A RFD node is intended to be used as low rate transmitter. Typically it is a sensor node that sends its little amount of information to another device, for example, for remote control purposes.

The network topologies described in the standard are the ones already described in section 1.1.5. In Figure 2.1 the role of FFD and RFD nodes is sketched.

## 2.2 Overview of the main services

This standard provides two types of primitives for the physical layer: the PHY *data service* and the PHY *management service*. The former enables transmission and reception of Packet Protocol Data Unit (PPDU), while the latter provides an interface to Physical Layer Management Entity (PLME) to enable some control services.

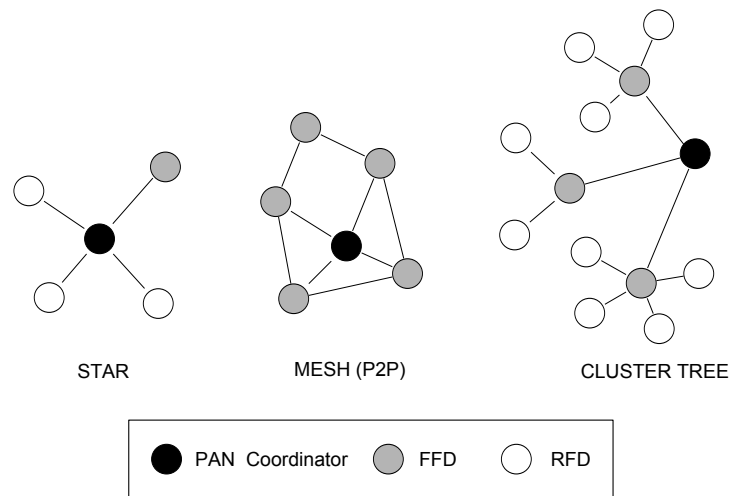


Figure 2.1: ZigBee network topologies.

Some of the principal services offered by the Physical Layer (PHY) are:

- Energy Detection (ED), senses the power on the wireless channel;
- Link Quality Indication (LQI), indicates the quality of the radio channel;
- Clear Channel Assessment (CCA), determines if the channel is free or busy;
- Activation and deactivation of the transceiver;
- Transmission and reception of packets;
- Selection of the transmission channel.

The last point introduces the channel subdivision provided by this standard. It comprises two bands: one for lower bit-rate transmissions (868-915 MHz) and the second exploiting the 2.4 GHz ISM band, providing higher bit-rate. A table with a more detailed description follows:

PHY (MHz)	Frequency band (MHz)	Spreading parameters		Data parameters		
		Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868/915	868–868.6	300	BPSK	20	20	Binary
	902–928	600	BPSK	40	40	Binary
2450	2400–2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

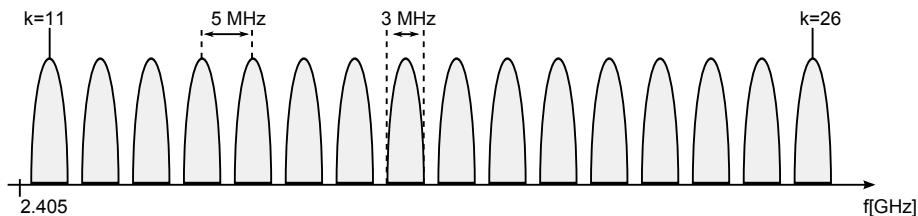
Figure 2.2: IEEE 802.15.4 bands and transmission rates.

A spread spectrum modulation, with the Direct-Sequence Spread Spectrum (DSSS) technique is used: before modulation, each symbol is split into a chip-sequence. It is a Pseudo-Noise (PN) sequence, having spectral properties similar to white noise [4]. In the following only the 2.4 GHz ISM band will be analyzed, where a 32 chip-long spread sequence is employed. This band is the only un-licensed band all over the world.

The standard divides this band into 16 channels ranging from 2400 MHz to 2483.5 MHz, spaced by 5 MHz each other. Every channel supports a fixed bit-rate of 250 Kbps, that leads to 2 Mchip/s. The carrier frequencies have this positioning within the band:

$$F_c = 2405 + 5(k - 11) \text{ MHz} \quad \text{with } k = 11, \dots, 26 \quad (2.1)$$

A sketch of the ISM band subdivision is given in Figure 2.3.



**Figure 2.3:** A representation of the spectrum used in this standard in the ISM band.

It is also specified a tolerance on the frequency of each carrier, and a tolerance on the bit-rate of each channel: both must be at least  $\pm 40$  ppm.

The sensitivity of receiver<sup>1</sup> defined in the standard is at least -92 dBm. An hardware compliant with the standard must correctly receive a signal with a power greater or equal than this threshold.

### 2.2.1 Energy Detection (ED)

This service provides an estimation of the received power in the communication channel. The estimation period lasts 8 symbol periods, returning a 8-bit unsigned integer ranging from 0x00 to 0xFF, where the minimum (0x00) must indicate an energy lower than 10 dBm above the sensitivity. The dynamic range of the ED estimation must be at least 40 dB. In this interval the power level in dBm has to be related to integer values on a linear scale with an accuracy of  $\pm 6$  dBm.

<sup>1</sup>The sensitivity of a receiver is the minimum power level allowing a correct reception of a frame.



Often the ED service is supplied by an RSSI<sup>2</sup> subsystem; this is the reason why the two terms are often used as synonyms.

### 2.2.2 Link Quality Indication (LQI)

The Link Quality Indication measurement is calculated at the reception of a packet. It is related to the Bit Error Rate (BER), calculated on a priori known set of bits, i.e. the Synchronization Frame Delimiter (SFD). The LQI is an estimation of the transmission “quality”, given with 1 byte (unsigned) precision.

### 2.2.3 Clear Channel Assessment (CCA)

The IEEE 802.15.4 standard’s medium access layer is based on the assumption of channel sharing among different nodes.

Thus every station wishing to transmit has to first listen to the channel for a predetermined amount of time to check for any activity on the channel. If the channel is sensed “idle” then the station is cleared to transmit. If the channel is sensed “busy” the station has to defer its transmission. This is the basis of the Carrier Sense Multiple Access (CSMA) technique. This medium access method is the main cause of coexistence problems with other standards operating in the ISM band.

There are three different CCA operating modes, implying performance and implementation differences.

**CCA Mode 1: Energy above threshold** The channel is sensed busy if the in channel power overcomes a chosen threshold (CCA threshold, typically -77 dBm).

**CCA Mode 2: Carrier Sense** The channel is considered busy only if a IEEE 802.15.4 compliant signal (modulation and spreading) is detected, regardless of the power level.

**CCA Mode 3: Carrier Sense and Energy above threshold** this is the combination of Mode 1 and Mode 2. The channel is busy if the signal type is IEEE 802.15.4 and the energy is above the threshold.

---

<sup>2</sup>RSSI is a measurement of the received in-channel power. It is a generic metric implemented in radio receivers. Its meaning is similar to that of ED, and is a technology present on IEEE 802.11 standard compliant devices.

### 2.2.4 PPDU description

Modern computer networks are based on a *ISO/OSI* layer subdivision. The application layer information is split and encapsulated into packets that cross the protocol stack down to the physical transmission medium and receive header information to be properly managed. A packet of layer  $n$  is the Protocol Data Unit (PDU), which contains the PDU of level  $n - 1$ . The PDU of level  $n - 1$ , at the level  $n$ , gets the name of Protocol Service Unit (PSU).

Before being transmitted, a *PHY* layer receives an header, forming the *PPDU*. This will be transmitted by the radio chip after the modulation. In the IEEE 802.15.4 standard the *PPDU* is composed of the parts depicted in the Figure 2.4.

Octets: 4	1	1		variable
Preamble	SFD	Frame length (7 bits)	Reserved (1 bit)	PSDU
SHR		PHR		PHY payload

**Figure 2.4:** The PPDU structure in the IEEE 802.15.4 standard.

Each packet is transmitted and received starting from the Least Significant Bit (LSB). In this standard, the maximum packet length is 127 bytes.

As shown in the Figure, the first part of the *PPDU* is the Synchronization Header (SHR), subdivided in the *preamble* and the *SFD*. This is an essential part of every packet-oriented transmission. In such a communication mode packets are generated in an asynchronous manner, so that the receiver must synchronize to the transmitter. The preamble is used to get bit-level synchronization, while *SFD* determines the start of the frame, and, hence, the frame synchronization. Generally preambles are pseudo-random sequences with a quasi-impulsive autocorrelation, allowing an easy synchronization.

The second section of the *PPDU* is the Physical Header (PHR), that contains information on the packet length. The last part is the *PHY* Payload, that is, the PDU of the upper layer that contains the real informative content of the packet.

## 2.3 MAC Layer

This is, certainly, the most critical layer of the *ISO/OSI* stack. Its characteristics determine the performance and the functionality of the whole standard. The complexity of this level in the IEEE 802.15.4 standard, however, does not allow a full explanation in this work: only some keys will be described in the following.

There are two types of *MAC* services that this standard provides:

**MAC Data Service** enables transmission and reception of the MAC PDU (MPDU) through the PHY layer services;

**MAC Management Service** provides some useful control services through the MAC Layer Management Entity (MLME).

The main topics covered by the MAC defined in this standard are:

- *beacon* management;
- channel access;
- GTS management;
- sending of Acknowledgment (ACK) packets;
- association to a PAN.

In addition to these services, the standard provides also services for authentication and ciphering: a communication can be without security, based on an access list or managed with authentication procedures and Advanced Encryption Standard (AES) cryptography with 32 or 128 bits.

Transmitter can also require ACK packet to ensure a reliable communication. Error correction techniques can be employed, but only Automatic Retransmission Query (ARQ) error correction algorithms fit well with such a low bandwidth system. Forward Error Correction (FEC) algorithms are not supported because they introduce a lot of redundancy in the transmission, decreasing the throughput to unacceptable levels.

### 2.3.1 The CSMA/CA algorithm

The wireless medium is, by definition, a broadcast medium. The channel has to be shared among different devices operating in the same band. It is hence necessary a channel arbitration that allows a multi-user fair access.

In some simple system, deterministic medium access methods like Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) are used. In these methods every node which is allowed to transmit has an assigned “radio resource”, that is a time slot in the former case, while is a dedicated frequency in the latter case.

This approach avoids collisions among transmissions. However, this is not the best choice from the performance point of view. These slots are assigned to a specific node, which can start to use them without alerting other network nodes. Assigned resources not used for a certain period are wasted.

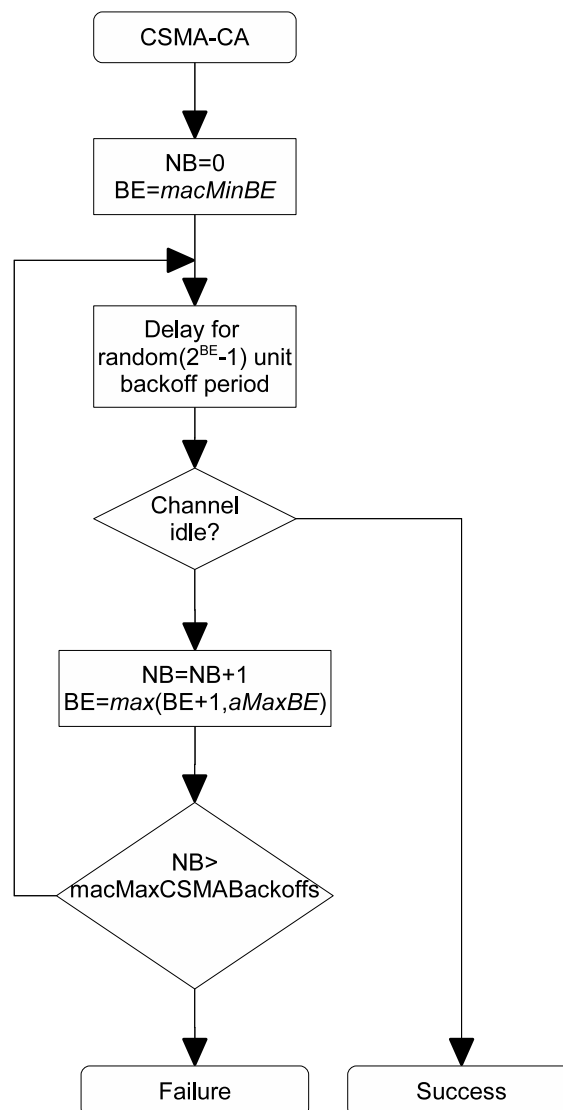
In stochastic methods there are no resources statically assigned to nodes (and hence wasted when not used), but, as a drawback, each device starts a transmission that can collide with other ones. In mean, stochastic medium access algorithms are more flexible than deterministic ones and are useful when there is no central network controller or in dynamic environments. Channel contention is intended to assign to a node in a dynamic way only the resources needed.

The CSMA/CA algorithm is an example of these probabilistic methods. It is also the more used in wireless communication; even IEEE 802.11 adopt it. In this algorithm, substantially, a node that has to transmit senses the channel before transmitting. If this is found “free” then the transmission will occur, else the node will wait for a random interval of time and retry the sensing procedure. This simple method, called *beaconless* in this standard, can be summarized as follows. Assuming that a node has to transmit then it has to go through these steps (see Figure 2.5):

- Wait for a random time  $\tau \in U(0, 2^{BE} - 1) \cdot BP$ , where  $U$  indicates a random variable with uniform distribution,  $BE$  is a term called Backoff Exponent and  $BP$  is a basic period (define in [2]);
- Sense the channel (i.e. perform the CCA);
- If the channel is sensed free, then start the transmission;
- If the channel is sensed busy, then  $BE = \min(BE + 1, aMaxBE)$ , where  $aMaxBE$  is the maximum value allowed for BE. Moreover, increment the counter of the retransmission ( $NB = NB + 1$ );
- If  $NB = \max(macMaxCSMABackoffs)$  then the transmission is failed and this has to be signalled, else repeat the algorithm.

The IEEE 802.15.4 standard provides also a *superframe* structure for transmission, through which the PAN coordinator manages the access to the channel. This structure is used in the *beacon mode*, in which the coordinator use packets named *beacon* to synchronize all devices that are connected to him, to describe the superframe structure and to identify the PAN. A beacon is transmitted on a regular basis between 15.36 ms to 251.65 s.

The superframe structure is represented in the Figure 2.6. It is subdivided in two parts: the first called “active” part and the second is the “inactive” part. The inactive part is used by devices for entering an idle state in which they do not communicate each other and have low power consumption.



**Figure 2.5:** The beaconless CSMA/CA algorithm described in the IEEE 802.15.4 standard.

The active part consists of 16 different time-slot, grouped in the Contention Access Period (CAP) period (9 slots), and the Contention Free Period (CFP) period (7 slots). In the CAP period communications occurs using the CSMA/CA algorithm, even if this is a modified version of the algorithm depicted in 2.6 on the following page to take into account of the time synchronization (the differences and the whole algorithm, named CSMA/CA *slotted* is discussed in [2]). In the CFP the coordinator provides some time-slots, called Guaranteed Time Slots (GTS), to other nodes; in these slots the channel access is free, i.e. a node does not have to use the CSMA/CA algorithm.

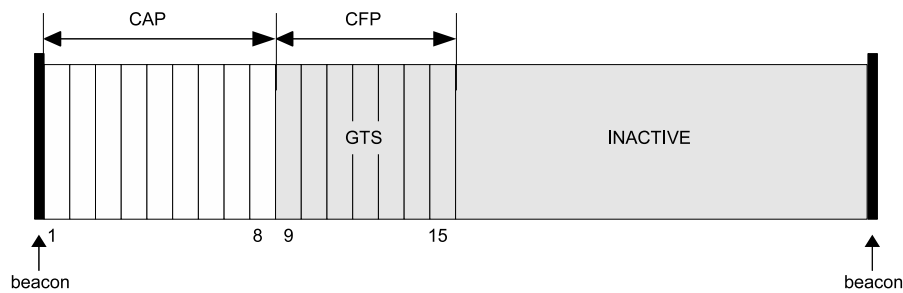


Figure 2.6: The superFrame structure in the IEEE 802.15.4 standard.

### 2.3.2 MAC frame

The IEEE 802.15.4 standard indicates that the maximum length of MAC packets must be 127 bytes. This measure is small compared to other wireless standard in which the medium packet length is about 512 bytes. Therefore, given such a limited number of bytes, the header has to be a small amount of the whole packet, to maintain an acceptable throughput.

The MAC provides four types of datagram:

**Beacon Frame** used only by the coordinator in the beacon mode;

**Data Frame** the datagram used in data transfer;

**Acknowledgement Frame** a small datagram, without neither addresses part nor payload section, to signal a correct reception;

**Command Frame** used to send command to control and set-up nodes.

All of these datagrams share the same packet structure. The first part of a packet is the Frame Control Field (FCF), 16 bit-long, and carries various information about the packet (type, addressing mode, security). Then the 8 bits-long Sequence Number (SN), i.e a progressive number, identifies the packet. After this byte there are the four address fields, with the source and destination PAN identifier and address. The addressing fields can be omitted in the acknowledgment frame and have variable length, depending on the type of address chosen (16 or 64 bits).

After this MAC Header (MHR) there is the payload section, which has a variable length, followed by the Frame Check Sequence (FCS) that ends the datagram. The FCS is a 16 bits-long field, containing a CRC for error correction, calculated on the header and the payload with the polynomial:

$$G(x) = x^{16} + x^{12} + x^5 + 1.$$

To explain in a more detailed way how MAC frames are composed, two diagrams are sketched out. These represent the general frame structure, Figure 2.7, and the Data Frame datagram, Figure 2.8.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Figure 2.7: The general structure of a MAC frame in the IEEE 802.15.4 standard.

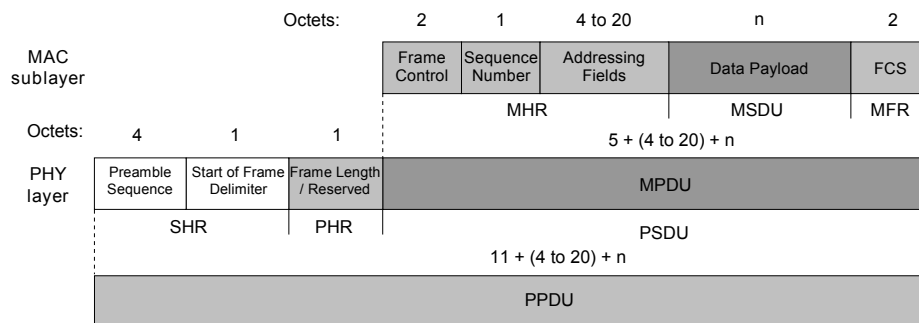


Figure 2.8: The structure of a data MAC frame in the IEEE 802.15.4 standard.

### 2.3.3 Addressing

Every device in a network has an assigned 64 bit IEEE address, thus the maximum length of the address field in the MAC frames is 8 octets (i.e. bytes). With 64 bits it is possible to address  $2^{64} \simeq 1.84 \cdot 10^{19}$  devices, perhaps more than the devices present in a common WSN. A broadcast address is provided and the assigned address is 0xFFFF.

A short address mode is available, using only 16 bit. A short address is assigned to a node during the association with a PAN. This address has validity only in that PAN, like a private network address on a LAN, and is lost if the node leaves the PAN. The short address allows to address 65535 devices and this can lead to cover almost all possible WSN networks.

## 2.4 PHY layer

The 2.4 GHz ISM band is subdivided in 16 channels ranging from 2400 MHz to 2483.5 MHz, spaced by 5 MHz each other. Every channel, supports a bit-rate of 250 kbps. The carrier frequencies are the ones described in section 2.2 and in equation (2.1).

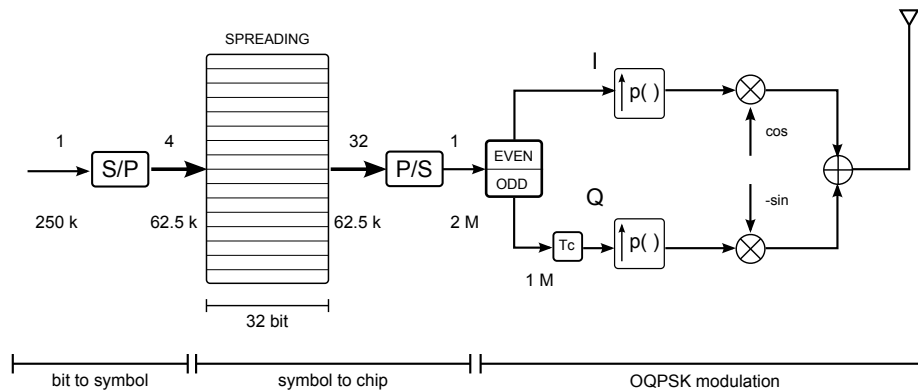


Figure 2.9: Block diagram of IEEE 802.15.4 modulator.

### 2.4.1 Transmitter functional blocks

The transmission circuitry can be divided into three blocks as depicted in Figure 2.9:

1. bits to symbols mapping;
2. symbols to chip (spreading) mapping;
3. Offset QPSK (O-QPSK) modulation.

Every PPDU is processed, starting from the LSB; therefore the first part processed is the preamble. Then bits are grouped in sets of four bits, hence if the bit-rate is 250 kbps (bit period is  $T_b = \frac{1}{250 \text{ kbps}} = 4 \mu\text{s}$ ) then the baud-rate achieved is 62.5 kbps. Then the spreading is performed onto this sequence of symbols: for every group of 4 bits a 32 bit word is selected. This results in a 2 Mchip/s sequence (chip period  $T_c = \frac{1}{2 \text{ Mbps}} = 500 \text{ ns}$ ). The term “chip” describes the bit output by the spreading process. The spreading process is the core of the DSSS technology: the rate of the system and hence the occupied bandwidth is increased by  $M$ , the *spreading factor*. For each bit input to the spreading circuitry  $M$  bit are output. In this case 4 bit are input and 32 are output through a sixteen elements look-up table, statically defined in the standard [2]. Hence a spreading factor equal to 8 is used.



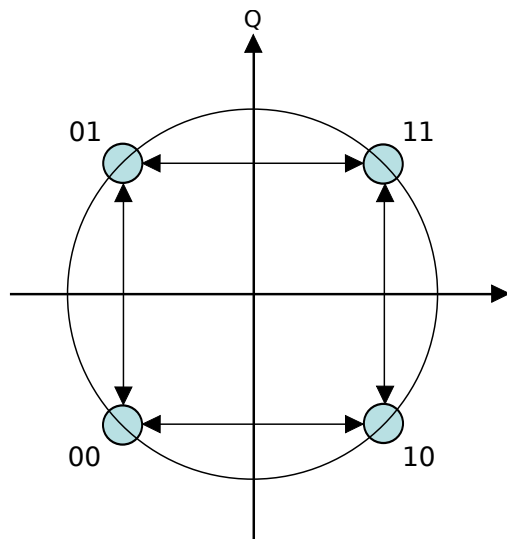
This technology can help the system overcoming impulsive interference [4] and spreads the power spectral density on a larger bandwidth, reducing emission on a specific band.

The chip sequence is then input to the O-QPSK modulator. This kind of modulation is a modified version of the Quadrature Phase Shift-Keying (QPSK) modulator.

QPSK can encode two bits per symbol. The modulator derives from the chip-sequence  $c_0, \dots, c_{2n+1}$  two sequences, called “in phase” (I) and “in quadrature” (Q):

- $C_I = c_0, c_2, \dots, c_{2n}$
- $C_Q = c_1, c_3, \dots, c_{2n+1}$

The symbol rate is half the chip rate, so the symbol period is equal to  $2T_c$ , and for each symbol two chip are transmitted. This pair of chip are then mapped on a constellation as represented in Figure 2.10, and generally symbols are coded using the Gray code to have only one bit changing during a symbol transition to one adjacent.



**Figure 2.10:** O-QPSK constellation scheme in PHY layer of the IEEE 802.15.4 standard.

The phase of the signal can assume four possible values. However this can lead to a jump of  $\pm\pi$  in the phase of the transmitted signal. After the chip to symbol mapping, the digital signal is converted to a real analog voltage through a Digital to Analog Converter (DAC), that is represented as an interpolating low-pass filter in Figure 2.9.

The filter shape is chosen in order to adapt the signal to the wanted bandwidth. Many integrated power amplifier need a constant-power signal, so a sudden change of phase, due to the diagonal cross of the constellation, may introduce undesired power fluctuations.

This problem can be overcome using the Offset QPSK modulation scheme. The sequence in quadrature is delayed of a chip period (see Figure 2.11), or half the symbol period, so the two sequence will never change at the same time (see the modulator in Figure 2.9).

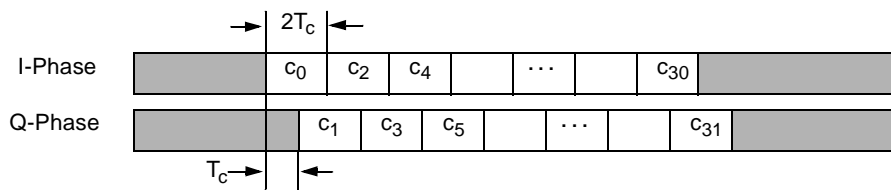


Figure 2.11: In-phase and Quadrature components of a O-QPSK modulator

This means that a transition between two symbols can occur only on the sides of the rectangle, never on diagonals, so only jumps of  $\pm \frac{\pi}{2}$  can arise.

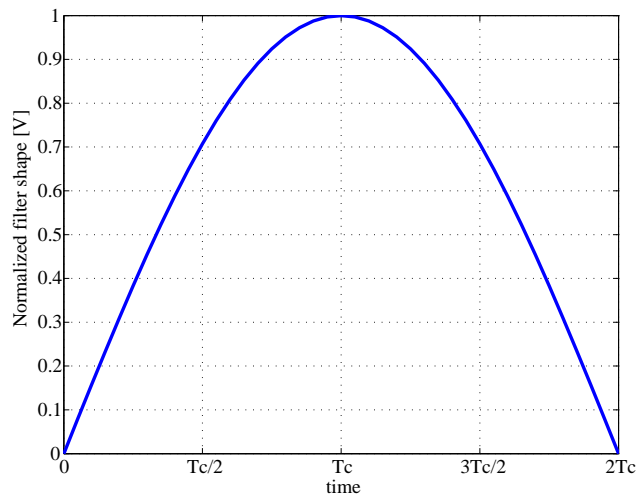


Figure 2.12: Normalized modulation filter shape.

In order to limit the power fluctuations, the filter shape can be chosen so that the signal must move only on the circle. This can be done using a half-sine filter shaping 2.12.

The closed form formula describing this filter is the following:

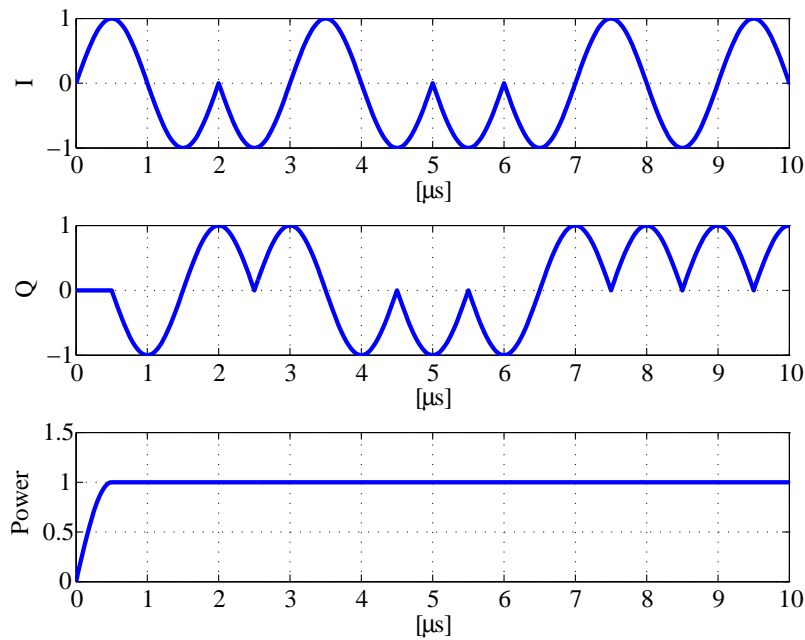
$$s(t) = \begin{cases} \sin\left(\frac{\pi t}{2T_c}\right) & 0 \leq t \leq 2T_c \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

It is straightforward to demonstrate that the power,  $P(t)$ , is always constant (apart from the initial and final transients) through the following equations. A simulative proof is also given in Figure 2.13.

$$I(t) = \sin\left(\frac{\pi t}{2T_c}\right) \quad (2.3)$$

$$Q(t) = \sin\left(\frac{\pi(t - T_c)}{2T_c}\right) = \sin\left(\frac{\pi t}{2T_c} - \frac{\pi}{2}\right) = \cos\left(\frac{\pi t}{2T_c}\right) \quad (2.4)$$

$$P(t) = I^2(t) + Q^2(t) = \sin^2\left(\frac{\pi t}{2T_c}\right) + \cos^2\left(\frac{\pi t}{2T_c}\right) = 1 \quad (2.5)$$



**Figure 2.13:** Simulation of the modulator output signals.

Finally, the O-QPSK modulation with half sine shaping is equivalent to a phase modulation, allowing a very efficient use of integrated transceivers.

The use of an half-sine filter, that is a time limited pulse, has a drawback in the frequency domain. The spectrum occupation is quite large. The closed form Fourier Transform of the filter impulse response can be easily calculated.

$$S(f) = \frac{2T_c}{2j} \left( \text{sinc}(2T_c(f - f_0)) e^{-j2\pi(f-f_0)T_c} - \text{sinc}(2T_c(f + f_0)) e^{-j2\pi(f+f_0)T_c} \right)$$

$$\text{with } f_0 = \frac{1}{4T_c} = 0.5 \text{ MHz} \quad (2.6)$$

Equation (2.6) describes the spectrum<sup>3</sup>. The squared absolute value of  $S(f)$  is the power spectrum mask (baseband) depicted in Figure 2.14. The center frequency, in practical implementation, is the chosen carrier frequency. It is worth noting that the main lobe is bounded in the range  $[-1.5 \text{ MHz}, 1.5 \text{ MHz}]$  and the secondary lobes are attenuated of 23 dB respect to the carrier frequency and centered nearly at  $\pm 2 \text{ MHz}$  respect to the carrier frequency. The other lobes are 1 MHz apart. At the adjacent channel center frequency ( $\Delta f = 5 \text{ MHz}$ ) the lobes are attenuated of 40 dB. In the following the main lobe is used as “bandwidth” of the channel: the conventional bandwidth is therefore 3 MHz large.

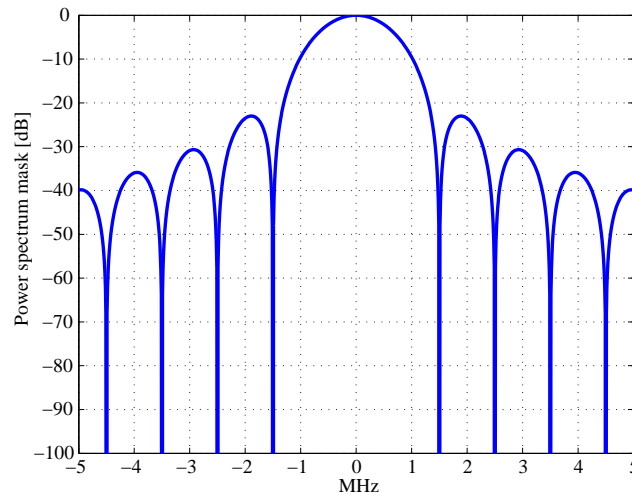


Figure 2.14: Normalized spectrum mask of a half-sine filter.

## 2.4.2 Receiver performance

The O-QPSK modulation joint with a DSSS system has been chosen for its robustness, ease of integration and for the BER performance. Compared to other standards, IEEE 802.15.4 shows very good theoretical performance. In Figure 2.15, different standards operating in the 2.4 GHz ISM band are compared.

<sup>3</sup>The *sinc* function is defined as  $\text{sinc}(a) = \frac{\sin(\pi a)}{\pi a}$

IEEE 802.15.4 is one of the most robust against noise, even if the other ones use higher bandwidth and, hence, suffer more from noise and interference. The use of DSSS circuitry allows the use of decoding algorithms such as Viterbi algorithm [4] that is used for a soft sequence decoding. These algorithms try to receive the transmitted sequence on a per-symbol-basis. The detection of a symbol is based on a maximum likelihood sequence detection on the whole 32-chip-sequence, outperforming single-chip detection [4].

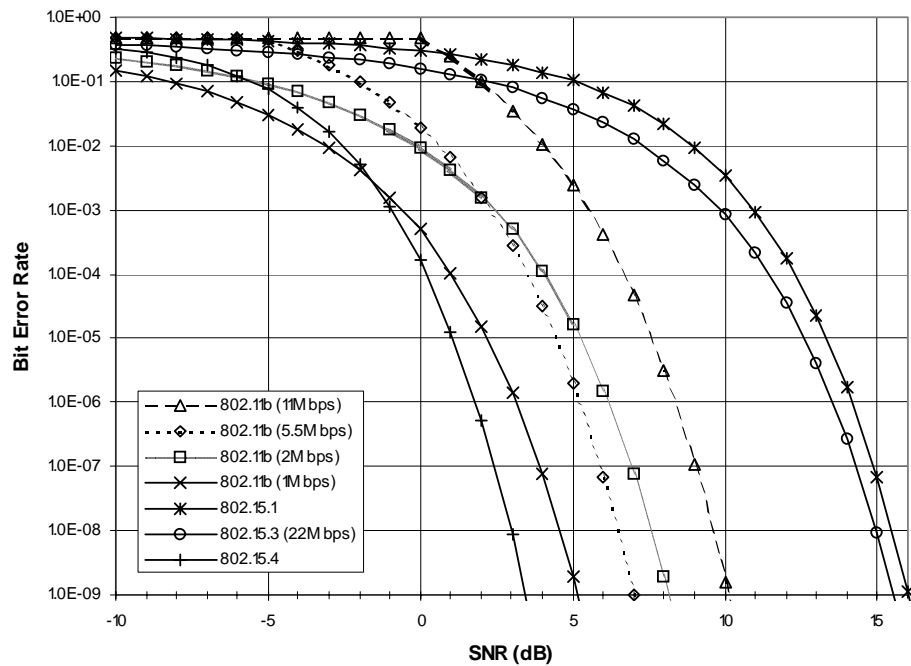


Figure 2.15: BER performance of various standards operating in the 2.4 GHz ISM band [2].



# Chapter 3

## *Tmote Sky e TinyOS*

**M**ANY hardware and software implementations are nowadays available as Commercial Off-The-Shelf (COTS) solutions for WSN. One of the most used is the *Tmote Sky*, a node produced by *Moteiv Corporation*. This platform is at the moment no more available, but improved nodes are available as COTS products. A great amount of transceiver and microprocessors or, even better, SOAC solutions, are nowadays available from the major integrated transceiver vendors (such as Texas Instruments or Freescale Semiconductor).

### 3.1 Tmote Sky

The Tmote Sky platform has been used in all experiments presented in this thesis because it is a well known platform for academic and educational purposes, it is cheap and quite all its function can be used through open source software<sup>1</sup>.

Some noteworthy platform characteristics are described in the following:

- 250 kbps 2.4 GHz IEEE 802.15.4 Chipcon Wireless Transceiver;
- Interoperability with other IEEE 802.15.4 devices;
- 8 MHz Texas Instruments MSP430 microcontroller (10 kB RAM, 48 kB Flash);
- Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller;
- Integrated onboard antenna with 50 m range indoors / 125 m range outdoors;
- Integrated Humidity, Temperature, and Light sensors;

---

<sup>1</sup>Most of the material in this chapter is taken from the Tmote Sky datasheet [5].

- Ultra low current consumption;
- Fast wakeup from sleep ( $< 6\mu s$ );
- Hardware link-layer encryption and authentication;
- Programming and data collection via Universal Serial Bus (USB);
- 16-pin expansion support and optional SubMiniature version A (SMA) antenna connector;
- TinyOS support (see section 3.2): mesh networking and communication implementation.

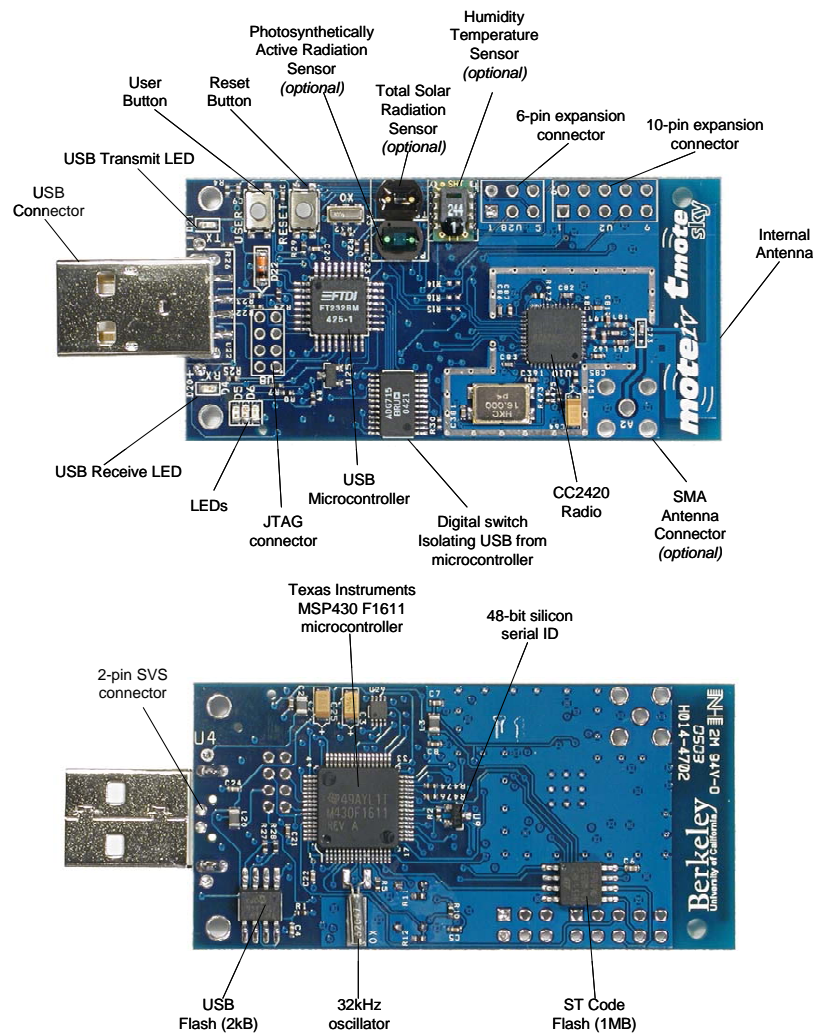


Figure 3.1: Front and Back of the Tmote Sky module.



### 3.1.1 Power supply

Tmote Sky is powered by two AA batteries. The module was designed to fit the two AA battery form factor. AA cells may be used in the operating range of 2.1 to 3.6 V DC, however the voltage must be at least 2.7 V when programming the microcontroller flash or external flash (see Figure 3.2). If the Tmote Sky module is plugged into the USB port for programming or communication, it receives power from the host computer. The mote operating voltage when attached to USB is 3 V. If Tmote will always be attached to a USB port, no battery pack is necessary. The 16-pin expansion connector can provide power to the module. Any of the battery terminal connections may also provide power to the module. At no point should the input voltage exceed 3.6 V: doing so may damage the microcontroller, radio, or other components.

	MIN	NOM	MAX	UNIT
Supply voltage	2.1		3.6	V
Supply voltage during flash memory programming	2.7		3.6	V
Operating free air temperature	-40		85	°C
Current Consumption: MCU on, Radio RX		21.8	23	mA
Current Consumption: MCU on, Radio TX		19.5	21	mA
Current Consumption: MCU on, Radio off		1800	2400	μA
Current Consumption: MCU idle, Radio off		54.5	1200	μA
Current Consumption: MCU standby		5.1	21.0	μA

Figure 3.2: Operating conditions.

### 3.1.2 Microprocessor

In Figure 3.3 there is a sketch of the different chips mounted on a Tmote.

The low power operation of the Tmote Sky module is due to the ultra low power Texas Instruments MSP430 F1611 microcontroller featuring 10 kB of RAM, 48 kB of flash, and 128 B of information storage. This 16-bit RISC processor features extremely low active and sleep current consumption. The MSP430 has an internal digitally controlled oscillator (DCO) that may operate up to 8MHz. The DCO may be turned on from sleep mode in 6 μs, however 292 ns is typical at room temperature. When the DCO is off, the MSP430 operates off an eternal 32768 Hz watch crystal. Although the DCO frequency changes with voltage and temperature, it may be calibrated by using the 32 kHz oscillator. In addition to the DCO, the MSP430 has 8 external ADC ports and 8 internal ADC ports. The ADC internal ports may be used to read the internal thermistor or monitor the battery voltage. A variety of peripherals are available including SPI, UART, digital I/O ports, Watchdog timer, and Timers with capture and compare functionality.

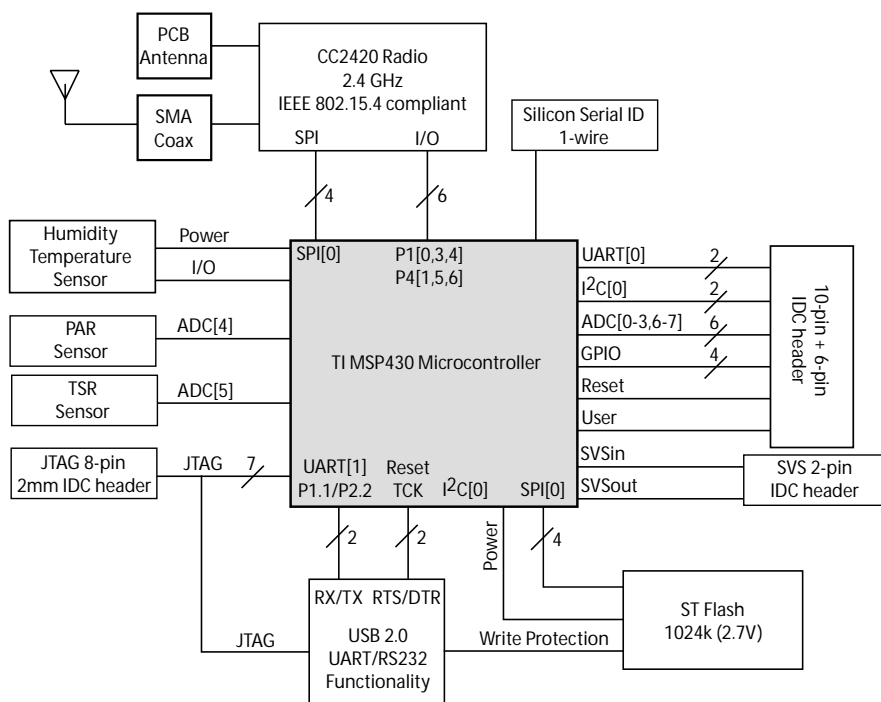


Figure 3.3: Blocks diagram.

The F1611 also includes a 2-port 12-bit DAC module, Supply Voltage Supervisor, and 3-port DMA controller. The features of the MSP430 F1611 are presented in detail in the Texas Instruments MSP430x1xx Family User's Guide available at [6].

### 3.1.3 Sensors

The Tmote Sky mounts a default sensor set and can use external ADC ports to connect other kind of sensors. The default sensors are:

- Humidity/Temperature Sensor;
- Light sensor TSR (Total Solar Radiation), sensing all visible spectrum including infrared;
- Light sensor PAR (Photosynthetically Active Radiation), sensing only artificial lights.

Each sensor must be calibrated for absolute measurements. The ADC provides a 12-bit-long output, and so values ranging from 0 to 4095.

### 3.1.4 Expansion Connectors

The Tmote Sky has two expansion connectors and a pair of on-board jumpers that may be configured so that additional devices (analog sensors, LCD displays, and digital peripherals) may be controlled by the Tmote Sky module. On the far side of the board from the USB connector is a 10-pin IDC header at position U2 and a 6-pin IDC header at U28. The 10-pin connector has the same connections as Tmote Sky and is the primary connector. It provides digital input and output signals as well as analog ones. Peripherals may be connected to the 10-pin connector using an IDC header, an IDC ribbon cable, or by designing a printed circuit board that solders directly on to the IDC header providing a robust connection to the module. An additional 6-pin (U28) header provides access to the exclusive features of Tmote Sky. Two additional ADC inputs are provided that may be reconfigured by software to be two 12-bit DAC outputs. ADC 7 may also act as the input to the supply voltage supervisor. The user interface elements (the reset and user buttons) are exported by the 6-pin header for use in external interfaces and packaging.

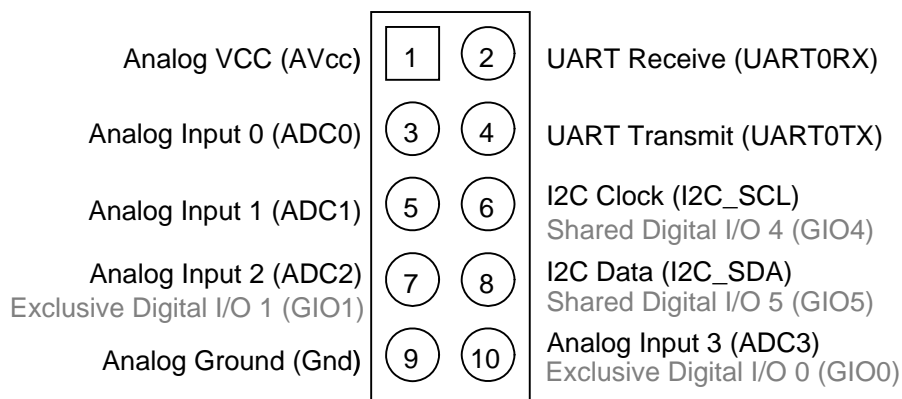


Figure 3.4: 10-pin connector.

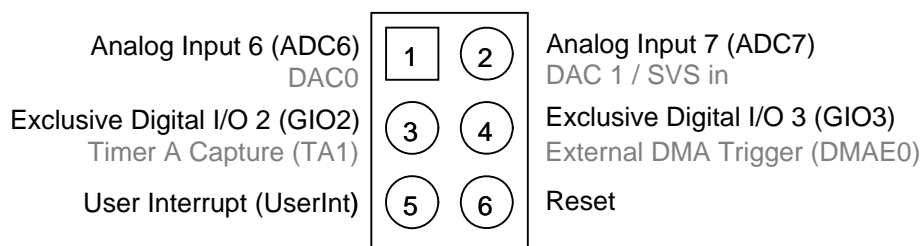


Figure 3.5: 6-pin connector.

### 3.1.5 Radio

Tmote Sky features the TI CC2420 radio for wireless communications. The CC2420 is an IEEE 802.15.4 compliant radio providing the PHY and some MAC functions. With sensitivity exceeding the IEEE 802.15.4 specification and low power operation, the CC2420 provides reliable wireless communication. The CC2420 is highly configurable for many applications with the default radio settings providing IEEE 802.15.4 compliance. The CC2420 is controlled by the TI MSP430 microcontroller through the SPI port and a series of digital I/O lines and interrupts. The radio may be shut off by the microcontroller for low power duty cycled operation. The CC2420 has programmable output power. Common CC2420 register values and their corresponding current consumption and output power are shown in Figure 3.6.

PA_LEVEL	TXCTRL register	Output Power [dBm]	Current Consumption [mA]
31	0xA0FF	0	17.4
27	0xA0FB	-1	16.5
23	0xA0F7	-3	15.2
19	0xA0F3	-5	13.9
15	0xA0EF	-7	12.5
11	0xA0EB	-10	11.2
7	0xA0E7	-15	9.9
3	0xA0E3	-25	8.5

Figure 3.6: Output power configuration for the CC2420 [7].

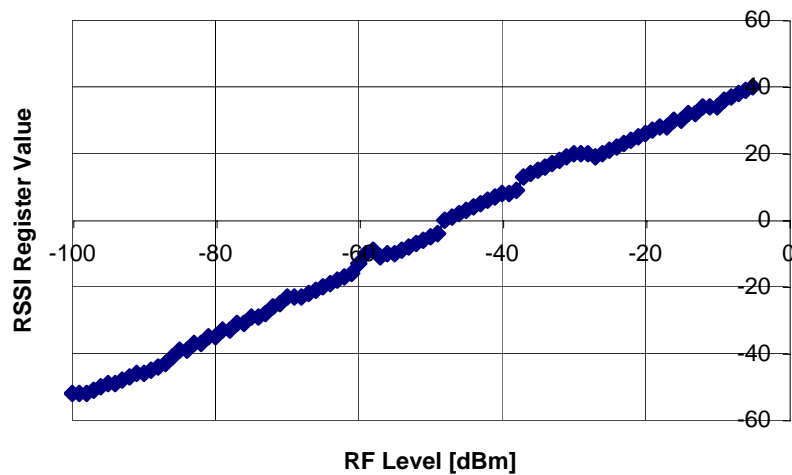


Figure 3.7: RSSI vs input power [dBm].

The channel is selectable both at compile and run time along with the output power. The CC2420 transceiver allows also non-standard operating modes, such as frequency hopping or channel spacing of 1 MHz. The CC2420 provides a digital RSSI that may be read any time. Additionally, on each packet reception, the CC2420 samples the first eight chips, calculates the error rate, and produces a LQI value with each received packet. A mapping from RSSI to the RF level in dBm is shown in Figure 3.7. The functional blocks of the CC2420 transceiver are depicted in Figure 3.8.

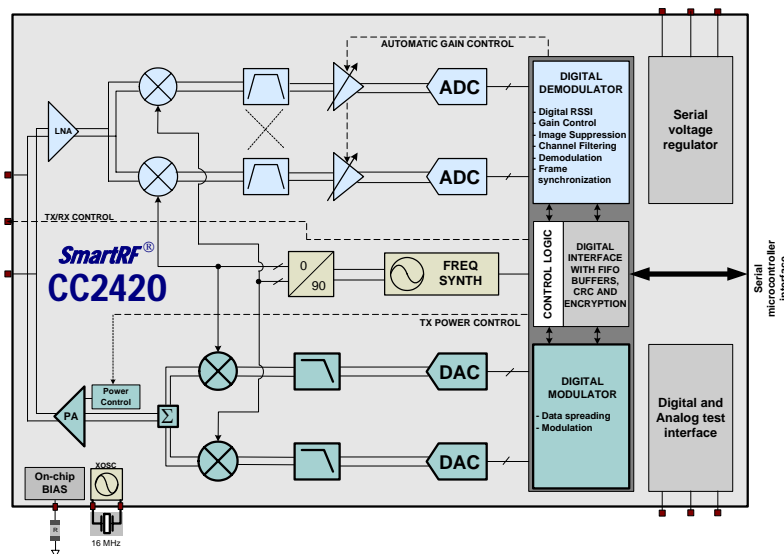


Figure 3.8: Functional blocks of a CC2420 transceiver [7].

## 3.2 Operating System: TinyOS

WSNs require specific software (firmware) solution to manage all the subsystems (radio, sensors). A first approach is a embedded firmware with no operating system. This is more efficient because the firmware is tailored for a specific application on a specific hardware, but increases the time to market and reduces the reusability of the software. A more flexible approach is to use a Operating System (OS) designed for such low-performance microcontrollers. This is a modular approach that can improve the code reuse and is far more simple to debug. Nearly ten years ago the first pioneer studies on such a computational-constrained hardware platforms started and nowadays several OSs are available for WSNs [8].

The main features of such an OS should be:

- **Low computational effort and low memory footprint:** sensors use low-performance microprocessors and have few KB of RAM memory, being designed primarily for energy saving.
- **Ultra low power operating mode:** typical PC operating system features such as multitasking and multithreading, using many load-save operations, are in many case too energy consuming.
- **Concurrency-intensive operation:** hard real-time constraints need a specific operating system. The correct multiplexing of radio and sensors implies an high concurrency tasks management: a large number of low-level events can be interleaved with higher level processing.
- **Diversity in Design and Usage:** WSNs can be used in very different contexts, so only the modules really necessary to fulfill the requirement of a specific sensor set or environment should be implemented, neglecting other part of the operating system.
- **Robustness:** WSN are intended for prolonged and hard work. The software, like the hardware, must be as reliable as possible, because in most cases no manned reboot or repair is possible.

### 3.2.1 TinyOS: general overview

Probably the most famous and diffused WSN OS is *TinyOS*. At the moment the release of this open source project is TinyOS 2.1, while throughout the thesis a proprietary solution provided by Moteiv Corporation, called Boomerang, has been adopted. It is a hybrid version between TinyOS 1.x and 2.x.

#### Task and concurrency

TinyOS is a open source project started by University of California, Berkeley. It is a modular OS with many modules implemented for microprocessor, radio and sensors access and management. The language used is named *nesC*, that is a C dialect. This OS has no *kernel* but only a *scheduler* that executes tasks according to a First In First Out (FIFO) queue. It is a event-based OS, only one task per time can be handled (TinyOS is a single tasking OS), so each task runs until completion. The scheduler has two degrees of priority: one, normal, for synchronous tasks, and the other, higher, for asynchronous hardware events. Only hardware interrupts can interrupt a task and potentially create race conditions in the code. Memory is statically allocated at compile time: RAM occupation is hence known and errors due to memory overflow are impossible.

# Chapter 4

## *Adopted software*

**T**HE results presented in this thesis have been obtained using two software applications employing Tmote Sky hardware platform. A first application, called *Factory Sniffer* controls an industrial monitoring system. This software has been used as a traffic generator for Packet Error Ratio (PER) evaluation in the presence of interference. Another software, called *Power Meter*, has been used to measure in-channel power providing an user-friendly Graphical User Interface (GUI). Both these software have been implemented primarily for the ease of performance measurement. To this aim different configurations are available, comprising the use of other instrumentation such as signal generators and analyzers. The release of the software described in this chapter is a generic one: minor differences will be described together with the different test-bed deployed. Only the user interface and the main features of these softwares will be described; code description and implementation issues are beyond the scope of this chapter. Focus will be paid on aspects directly related to performance issues and to measurement methods. Code is written in Java for the PC-mote interface GUI, while the mote firmware is written in nesC for the TinyOS OS.

### **4.1 Factory Sniffer: Industrial monitoring**

An industrial environment needs reliable and robust monitoring system in order to control factory processes. Every automation system is made up of three fundamental blocks:

- Sensor sub-system that follows some physical phenomena;
- Actuators sub-system that apply control signals to the plant;
- Elaboration sub-system that manages sensors and actuators.

The connection among these sub-systems can be centralized, when there is a central processing unit that manages all the process or distributed, when local sub-systems can manage events and react without central coordination.

Different constraints and dead-line are to be met in different contexts: sensor networks intended for data logging and event alerting have typically less stringent requirements than sensor networks employed within a process control chain. In the former case the *jitter* (variability in the sampling time) can be managed using buffers because the system is not intended for real-time purposes. In the latter case the *determinism*, i.e. the almost perfect predictability of the sampling instants, is needed.

#### 4.1.1 General description

Factory Sniffer is a centralized industrial monitoring system based on IEEE 802.15.4 compliant platforms. The main aim of this system is to evaluate performance of a wireless link when used in quasi real-time applications.

A general purpose high layer protocol, based on a master-slave relationship, has been designed and implemented on a wireless sensor network. The network comprises one master and a number  $N$  of slaves. The protocol performs two classic tasks: a periodical polling of each slave for receiving data from monitoring sensors, and an asynchronous alarm transmission, able to handle possible dangerous events from peripheral sensors.

#### Cyclic polling

The periodical task is performed according to a *round robin* fashion. The master node assigns a specific *polling time* to each slave, and after its expiration passes to the next slave, in order to maintain a sort of time-slot division. In figure 4.1, a schematic representation of master and slave communication within a polling window is shown.

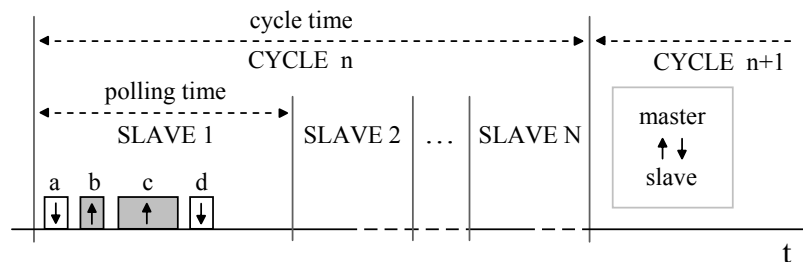


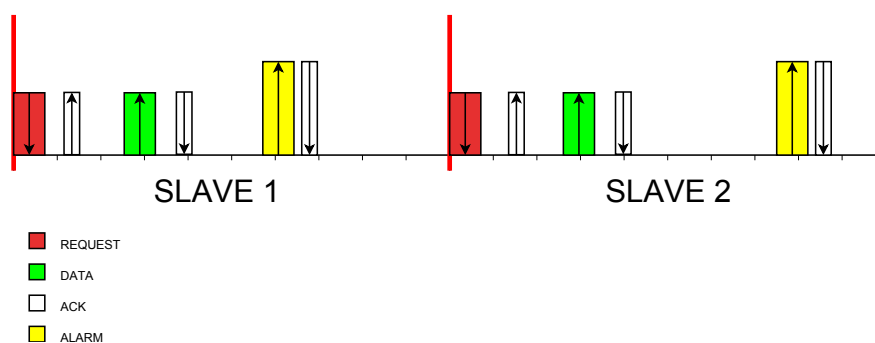
Figure 4.1: Packets exchange between master and slave inside a polling time.



At the MAC layer, both the master and the slave must perform a carrier sensing in order to access the channel. Once gained the channel, the master begins querying the slaves and transmits a frame (request) (a) to the first node, containing information like the destination node address, the sequence number of the performed cycle and time stamps. The queried slave, after a well-defined short delay (12 symbol periods) replies with an ACK frame (b), and then sends a message containing the process data (c), after a second carrier sensing. In case of correct reception, the master issues an ACK (d), waits for the expiration of the polling window before passing to the following slave. If the sensor does not receive the ACK from the master, then the polling is lost. At the expiration of the polling time, if ACK has not been received, the data packet from that sensor is considered lost. Once terminated the polling of each slave, the cycle restarts from the first slave. The time needed for a cycle is called *cycle time*.

### Alarm management

The acyclic task allows a slave to break the cyclic period and create a direct communication with the master for the transmission of alarms (see Figure 4.2). In this case, we consider the possibility for a slave to exploit a memory area to implement a queue of alarms. For this queue, in order to minimize the alarm latency, an *immediate policy* to manage alarms is considered [9]. In practice, once an alarm is launched, it is put into the queue and sent as soon as the channel is idle and the slave can access it. Only after the correct alarm delivery, the task can go on and process the other alarms similarly until the queue is empty. Obviously, the presence of acyclic events may negatively affect the cyclic operation through polling jitter or even packet loss. Alarms are software generated according to a Poisson Process [10] whose mean ca be changed with the GUI. The basic random variable outcomes are not generated on-board, but are stored into a look-up-table.



**Figure 4.2:** Polling and alarm packets inside a polling time.

## Sensing feature

The normal network functioning performs cyclical polling and alarm management. A further testing feature, called sensing, has been implemented in order to evaluate RSSI indication from each slave and the master. When the network is in the sensing state, no polling nor alarms are exchanged and no traffic over the air is present at all. The master and the slaves start evaluating the in channel power for a determined period (SENSING PERIOD in the GUI) and then evaluate minimum, mean, variance and maximum of the RSSI. This feature is used to estimate the interference power or the noise floor. The sense message is sent by radio and by USB, so that it can be delivered to the PC even when the interference level does not allow the radio communication.

### 4.1.2 Packet structure

The MAC payload, i.e. the data written inside each packet, is joint with an header comprising both fields compliant to the IEEE 802.15.4 standard, first group in listing 4.1 and management fields used by TinyOS (second group) and not actually transmitted. The transmitted packet is hence 19 byte longer than the payload.

---

```

0 typedef struct TOS_Msg
  { //transmitted
    uint8_t length;
    uint8_t fcfhi;
    uint8_t fcflo;
5    uint8_t dsn;
    uint16_t destpan;
    uint16_t addr;
    uint8_t type;
    uint8_t group;
10   int8_t data[TOSH_DATA_LENGTH]; //this is the payload

    //not transmitted
    uint8_t strength;
    uint8_t lqi;
15   bool crc;
    bool ack;
    uint32_t time;
  } __attribute__((packed)) TOS_Msg;

```

---

**Listing 4.1:** Generic TinyOS message, TOSMsg (see Figure 2.7)

The packet exchanged over the air are of five different types: *StartMsg*, *PollReq*, *PollResp*, *AlarmMsg*, *SenseMsg*. In the following a brief explanation is provided.

**StartMsg** This message is used only for network configuration. The network parameters are collected from the GUI and the network is set up. This message is used only for starting, stopping and resetting the network. It is typically a broadcast message. It occupies the channel for 1.024 ms (32=13+19 byte).

**PollReq** This is a downlink message used to poll a specific slave in a unicast manner. It stores timestamps and other control fields, such as a synchronization field used to synchronize the slave clock to the master one. This packet occupies the channel for 0.928 ms (29=10+19 byte).

**PollResp** This packet is a uplink packet sent in unicast to the master in response to a *PollReq*. It stores sensor reading (light sensor) and timestamps. This packet occupies the channel for 1.248 ms (39=20+19 byte).

**AlarmMsg** This is an uplink packet sent asynchronously to the master when a specific slave generates an alarm.

**SenseMsg** Used to collect mean, variance, minimum and maximum of the RSSI of a node. It is sent both by radio and by USB.

### 4.1.3 GUI and main features

In this section an in-depth overview on the software Graphical User Interface (GUI) is provided, especially describing configuration parameters and buttons. The firmware installed on motes is not further described while the attention is focused on message exchanged over the air and on data logged by the java program used to interface the WSN with a PC.

The Factory Sniffer system assumes the network to be already created. No association or de-association is managed by this software. The master is connected through an USB connector to a central PC used for network configuration and mass storage. The polling process takes care to maintain a degree of synchronization among master and slaves' clocks. At each polling the slaves' clock is adjusted to follow the master's one. Considering the precision of crystal clocks mounted on motes and this periodic adjustment, the time difference between master and slaves' clock is in the order of the microsecond, and, hence, negligible for the time scales used in all the thesis (tents of milliseconds).

All the hardware platforms of the motes are identical. The only difference among master and slaves is due to the firmware. Energetic issues have been neglected in this thesis, and hence all nodes use the USB as power supply. No tests have been performed on batteries lifetime nor sleep function has been used. Factory Sniffer is intended as a versatile and customizable performance tester.

### Java code

The Java GUI is responsible for network setup, start-reset-stop commands and for data acquisition in a user-friendly matrix. In Figure 4.3 a typical screenshot of the GUI is provided. It is a sort of oscilloscope (based on the *Oscilloscope* application embedded in the TinyOS package): for each slaves two virtual channel are created, storing the polling results on one side and the alarms one the other. There is a graphic panel showing the sensor readings.

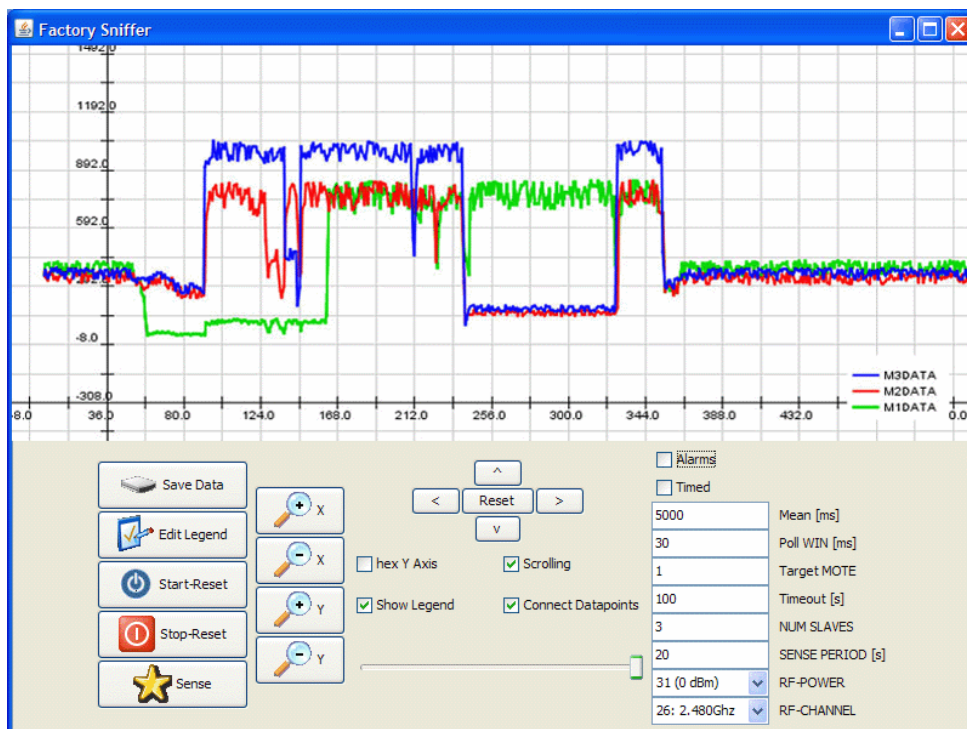


Figure 4.3: Factory Sniffer GUI.

The leftmost button column is used for sending start, stop and reset messages and also to save data before quitting the program. The sensing feature can also be used. The saved data are formatted both in a plain text format and in a MATLAB<sup>®</sup> matrix. This function is provided by the package JMatIO (Copyright 2006, Wojciech Gradkowski) freely available on-line.

The sensing estimates are only saved as text. The MATLAB-friendly format is very helpful for the elaboration and stores the data with higher precision (all data are stored as double precision floating point numbers) and less disk space (being a compressed format). The central panel is used for graphical purposes and can modify the view of the graphic panel.

The rightmost panel is the most important, allowing to set the main network parameters. A brief explanation is described in the following. By default, average values are inserted.

- **Alarms:** determines if alarms are enabled or not;
- **Timed:** if checked, stops the network after *Timeout* seconds;
- **Mean:** sets the mean inter-arrival time of the Poisson process for the alarm generation. Active only if *Alarms* is checked;
- **Poll WIN:** polling window, that is the time assigned for the polling of each slave. Multiplied by the number of slaves gives the *cycle time*;
- **Target MOTE:** used for testing and debug purposes;
- **Timeout:** used in the Timed mode, stops the network after this period is expired;
- **NUM SLAVES:** number of slaves to be polled;
- **SENSE PERIOD:** used only in the sensing state, sets the estimation duration;
- **RF-POWER:** transmit power, based on the [2] standard (see Figure 3.6);
- **RF-CHANNEL:** transmit channel. The channel list is greater than the one described by equation (2.1) because the radio can operate on channels separated by 1 MHz.

## Data fields

Each virtual channel linked to a slave contains several useful information. An example of this data is visible in listing 4.2.

---

```

0 #:MIDATA,10 SAMPLES:(seqno-delta-val-MReq-S-MResp-MSpower-SMpower)
  #:alarms:false,mean[ms]:5000,win[ms]:30,target:1,power:30,
  #:channel:26,
  0 0 127 32.70932 32.71368 32.72061 -31.0 -33.0
  1 1 100 32.79721 32.80157 32.80850 -29.0 -31.0
5  2 1 140 32.88510 32.88946 32.89639 -30.0 -32.0
  3 1 162 32.97299 32.97735 32.98428 -29.0 -31.0
  4 1 145 33.06088 33.06524 33.07220 -29.0 -31.0
  5 1 121 33.14877 33.15313 33.16006 -29.0 -31.0
  6 1 126 33.23666 33.24102 33.24795 -29.0 -31.0
10 7 1 157 33.32455 33.32891 33.33584 -29.0 -31.0
  8 1 166 33.41244 33.41680 33.42373 -29.0 -31.0
  9 1 147 33.50033 33.50469 33.51162 -29.0 -31.0

```

---

**Listing 4.2:** Saved data sample.

First three lines are header containing a summary of the GUI settings. In the first line the channel name, the total number of received packets, and a brief description of the data field are provided.

Data lines are formatted as following:

- **Sequence Number:** identification number for a specific packet;
- **$\Delta$ -Sequence Number:** differential sequence number, helpful to detect packet loss at a first glance. If  $\Delta > 1$  then  $\Delta - 1$  packets have been lost;
- **Val:** sensor reading;
- **Master request timestamp:** timestamp (ms) of the master request;
- **Slave response timestamp:** timestamp (ms) of the slave response;
- **Master response timestamp:** timestamp (ms) of reception time;
- **MSpower:** power (dBm) detected by the slave during the reception of the master request;
- **SMpower:** power (dBm) detected by the master during the reception of the slave response.

Alarm channels are identical to polling ones except for the fact that several fields are not used.

## 4.2 Power Meter

A second software, useful to evaluate power measurement suitability of motes has been implemented. It does not use radio transmission and hence is not intended for PER evaluation. The typical testbed employs a single mote with Power Meter firmware installed on it and a PC used for data logging and mote setup. The is intended as a stand alone mote connected to a PC.

### 4.2.1 Main features

The measurement intent of this software is best used if the default on board printed antenna is substituted by a SMA coaxial connector. With this approach the measurand RF power can be directly injected into the connector even with a cable, avoiding a lot of uncertainty due to the un-calibrated default antenna.

In order to collect high automated data, Power Meter can be used both in a free-running and triggered mode. The triggered mode is intended to collect an  $M$  number of samples related to exactly  $M$  power levels.

The results show the in channel power, that is an integral of the whole power passing through the channel filter. Spectral properties of the Power Meter will be described in the following.

#### Free-running mode

A general purpose feature is the free-running functionality. Once chosen the receive channel, the input power is sampled with a constant sample rate and the results are put in packets and sent through the USB connector to the PC. The minimum sample rate that can be used is 10 ms.

#### Triggered mode

The idea behind the triggered mode is to use the user button in an automated way. The user interrupt can be also launched using the User Interrupt (UserInt) pin (pin 5 in Figure 3.5). An external instrumentation, such as a signal generator, with an output trigger port that can cause a high-to-low transition can be used to this purpose.

Triggered mode is used when the number of power steps to be measured and the number of the events triggering these measurements points is known. It can be used only in calibrating stages where the measurand is a synthetic signal generated by a signal generator. In this case, also this generator must be software driven to perform a fully automatic calibration process.

### 4.2.2 GUI and data format

The GUI is very similar to the one of Factory Sniffer software. The main panel is a graphic panel showing the results as soon as the packets are sent to the PC. The leftmost control panel is used to start, stop, reset and save the measurements. The central control panel is used for graphic panel view management. The most important panel, the rightmost one, is used to select the operating mode (triggered or not), the channel to be monitored, the sample period (in non-triggered mode) or the samples per step, *i.e.* the number of samples for each trigger event (in triggered mode only).

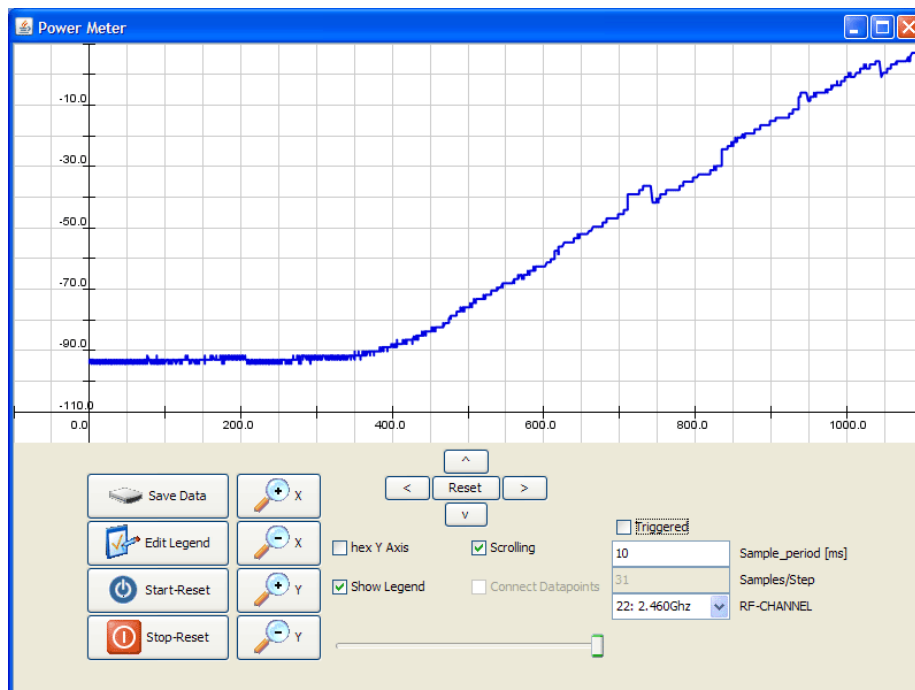


Figure 4.4: Power Meter GUI.

As for Factory Sniffer, also Power Meter saves data both in text and Matlab formats. The first column is used for sequence number, the second for delta sequence number and the third, for power samples in dBm, as visible in listing 4.3.

```

0 # : MOPOWER, 5 SAMPLES: sample_period: 10 [ms] channel: 22
  0 0 158
  1 1 158
  2 1 158
  3 1 158
5 4 1 159

```

Listing 4.3: Power Meter data.



# Chapter 5

## *WSN based industrial control system*

**W**SNS are nowadays a promising and powerful novelty in the scenario of industrial communications [11]. They provide distributed sensing features having interesting properties with respect to the wired networks, such as the absence of infrastructures, low cost, scalability and flexibility. Nevertheless, some key drawbacks are actually delaying a wide deployment of these systems for industrial applications, due to some technical problems still far to be completely solved. One critical issue is, for instance, the poor reliability of WSNs, when some types of in-channel radio interference occur. This situation is typical for WSNs which transmit on radio channels shared with other communication systems and hence crowded of radio disturbances, like for instance the 2.4 GHz ISM band (2.4 - 2.4835 GHz). In this band, several sources of interference can be encountered, including for example IEEE 802.11b/g wireless local area networks (WLANs) [1], IEEE 802.15.4 [2], and IEEE 802.15.1 (Bluetooth) enabled devices, microwave ovens, etc. Moreover, severe timing constraints may be often required, meaning that relevant delays and/or uncertainties in data transfer might not be tolerated.

The most typical effect of interference is signal degradation, which occurs at the WSN receivers when the incoming useful signal is affected by interference. In order to reduce such effect, some communication standards employ sensing mechanisms like the CSMA/CA protocol (see Section 2.3.1). This mechanism enables multiple users to share the same physical medium, avoiding collisions and consequently signal degradation but if the interference at the WSN receivers is higher than a prefixed threshold, the channel is assumed as busy, and any node wishing to transmit is forced to wait and delay the delivery of data packets [11].

To properly deal with these interference issues, a special effort is required just at the early stage of a design. Helpful information can be found either theoretically, through the use of suitable simulation models, or experimentally, through measurements. Useful application notes can be also found in the scientific literature. For instance, in [12] the behavior of the IEEE 802.11 physical layer in an industrial environment is deeply investigated; in [13] a saturation analysis for IEEE 802.11 networks is reported; more generally, papers [11], [9] consider the usage of wireless technologies for industrial applications. Finally, papers [14], [15] are concerned with the use of IEEE 802.15.4. Conversely, very limited analyses are available in terms of measurement techniques for WSN characterization in the presence of interference.

In this chapter, the performance of a CSMA/CA-based WSN in the presence of interference is investigated through a set of experimental tests. In particular, the tests have been performed on a testbed enlisting a real life IEEE 802.15.4 WSN for which a specific high layer protocol is used, capable both of cyclic polling and acyclic alarm management. The purpose of this chapter is to show how to measure some specific parameters of an industrial WSN, in order to obtain valuable information for its setup optimization in the presence of interference. All the analysis and results collected in this chapter are taken from [16, 17].

## 5.1 High Layer Monitoring protocol

In this chapter the Factory Sniffer system has been used as an industrial monitoring test system. As described in chapter 4, the protocol performs two classic tasks: a periodical polling of each slave for receiving data (*e.g.* readings of temperature, luminosity, humidity, pressure, rotation angle, etc.) from monitoring sensors (cyclic task), and an asynchronous alarm transmission, able to handle critical events from peripheral sensors (acyclic task).

### 5.1.1 Cyclic task

The cyclic task is performed according to a round robin fashion. The master node assigns a specific *polling time*,  $T_p^i$ ,  $i = 1, 2, \dots, N$  (where  $N$  is the total number of slaves) to each slave, and after its expiration passes to the next slave, in order to maintain a sort of time-slot division. In Figure 4.1 on page 34, a schematic representation of master/slave communication is shown.

As can be seen, once gained the channel, the master begins querying the first slave and transmits a frame (a) to it, containing information like the destination node address, the sequence number of the performed cycle and time stamps. The slave, after a fixed delay defined by the IEEE 802.15.4 standard (12 symbol periods), replies with an ACK (b), and then sends a message containing the process data (c), after a second carrier sensing. In case of correct reception, the master issues an ACK (d) and waits for the expiration of the polling time before passing to the following slave. If the handshake is not performed correctly (i.e. either the master or the slave does not receive the ACK frame), then the polling of that slave is considered failed. Once the last slave has been queried, the master restarts from the first one, implementing in such a way the polling cycle. The time needed for querying all the slaves is called *cycle time*.

A time representation of the master/slave communication is shown in Figure 5.1 (here, for commodity, they are indicated as M and S respectively), where  $n$  indicates the sequence number of the corresponding cycle.

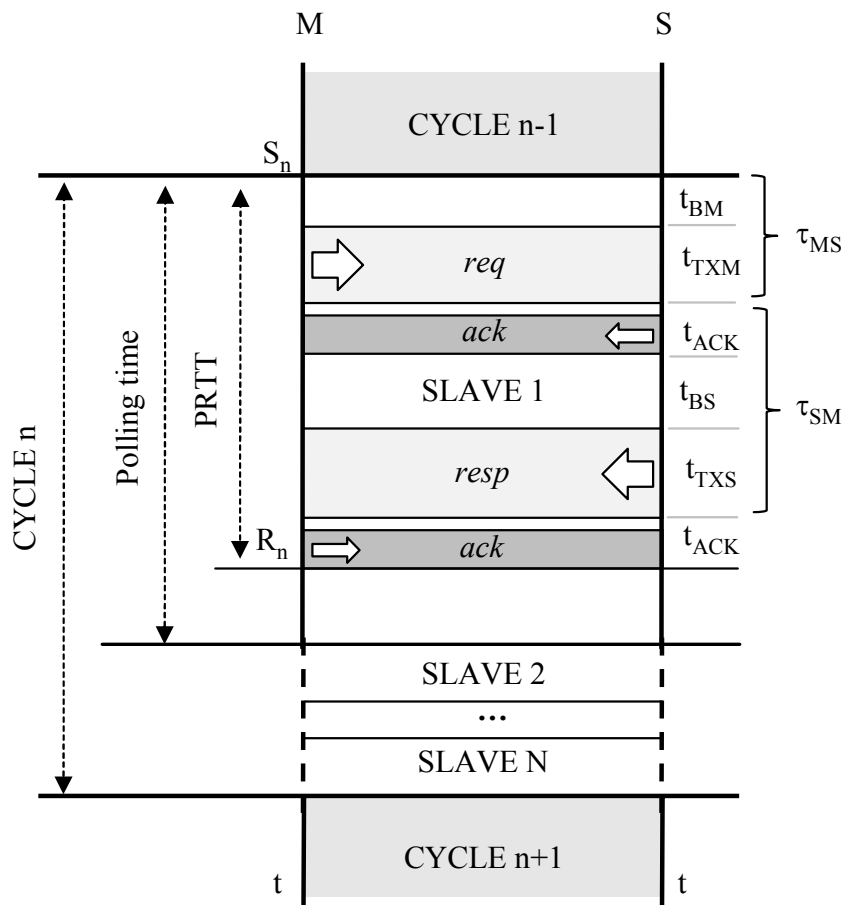


Figure 5.1: Time diagram of master and slave communication.

At the instant  $S_n$ , the master transmits a *request*, which is received after a time interval  $\tau_{MS}$  given by:

$$\tau_{MS} = t_{BM} + t_{TX_M} \quad (5.1)$$

where  $t_{BM}$  is the *initial backoff* period needed to access the channel by the master and  $t_{TX_M}$  is the time actually necessary to transmit the frame from master to slave.

Then, after the time necessary to send the acknowledgment from slave to master ( $t_{ACK}$ , which includes also the 12 symbols periods delay) the queried slave replies with a *response*, received by the master after a time interval  $\tau_{SM}$  given by:

$$\tau_{SM} = t_{ACK} + t_{BS} + t_{TX_S} \quad (5.2)$$

where,  $t_{BS}$  is the *initial backoff* period needed to access the channel by the slave and  $t_{TX_S}$  is the time actually necessary to transmit the frame from slave to master.

Finally, the master sends an acknowledgment frame to the slave and the polling ends at the instant  $R_n$ .

The difference between  $R_n$  and  $S_n$  is the Polling Round Trip Time (*PRTT*), which represents the time employed by the master to execute a complete query of a slave. It may vary from cycle to cycle and, clearly, it depends on the queried slave. The general expression of *PRTT*, where for commodity the reference to the specific slave being queried has been omitted, is:

$$PRTT = R_n - S_n = t_{BM} + t_{TX_M} + t_{BS} + t_{TX_S} + 2t_{ACK} \quad (5.3)$$

*PRTT* is a measurable parameter from which interesting information about both the presence and the effect of interfering sources influencing the wireless channel can be deduced. Indeed, if interference is not present, only one channel access attempt is performed by both master and slave and, consequently,  $\tau_{MS}$  e  $\tau_{SM}$  are uniformly-distributed independent random variables. In this case, a triangular distribution of the values assumed by *PRTT* is expected. Conversely, in presence of interference, more attempts to access the channel are possible, since the interference may have the effect of emulating a busy channel situation. In this case, both  $\tau_{MS}$  and  $\tau_{SM}$  are no longer uniformly-distributed independent random variables and, hence, different distribution of *PRTT* values are expected, depending on the interference characteristics.

A second parameter useful in the analysis of interference effects is the experimental cycle-time ( $ECT$ ), defined as the difference between two consecutive arrival times,  $R_n^i$  and  $R_{n-1}^i$ , from the  $i$ -th slave:

$$ECT^i = R_n^i - R_{n-1}^i = \underbrace{S_n - S_{n-1}}_{\text{deterministic}} + \underbrace{(PRTT_n^i - PRTT_{n-1}^i)}_{\text{random}} \quad (5.4)$$

As shown in equation (5.4),  $ECT^i$  accounts for two different contributions: the first one,  $S_n - S_{n-1}$ , is deterministic since it represents the time elapsed between two subsequent transmissions of the request frame from the master to the same slave. Indeed, this value is given by the sum of polling times assigned to each slave:  $S_n - S_{n-1} = \sum_{i=1}^N T_p^i$ . The second contribution,  $PRTT_n^i - PRTT_{n-1}^i$ , is a random variable, since it is given by the difference between  $PRTT$  values evaluated for cycles  $n$  and  $n - 1$  for the  $i$ -th slave. In this case, a different distribution of  $ECT^i$  values is expected depending whether the interference is present or not. In particular, the outcoming experimental Probability Density Function (PDF) will be gaussian-like only in the absence of interference, since in this case both  $PRTT_n^i$  and  $PRTT_{n-1}^i$  have triangular shaped PDFs and their difference leads to a gaussian-like PDF.

In case of failed pollings, more than one cycle may elapse between two consecutive successful queries of the same slave. Indeed, supposing that slave  $i$  is not queried correctly at cycle  $n$ , when its polling time expires, the master will move to slave  $i + 1$  regardless of the failure. As a consequence,  $PRTT_n^i$  in this case can not be evaluated due the lack of response from slave  $i$  within the polling time  $T_p^i$ . Nonetheless, it is interesting to know when slave  $i$  will be eventually polled with success. To such an extent, a further parameter to be monitored has been introduced,  $\overline{PRTT}_n^i$ , which represents an extension of  $PRTT_n^i$  since it is defined as the actual time necessary to poll slave  $i$ , even if such an operation takes more than one cycle. Similarly, an extension of the experimental cycle time has been defined,  $\overline{ECT}^i$ , as the time elapsed between two consecutive successful responses of slave  $i$ , occurring at a “distance” of more than one cycle, given by:

$$\overline{ECT}^i = S_n - S_{n-1} + \overline{PRTT}_n^i - \overline{PRTT}_{n-1}^i \quad (5.5)$$

Clearly,  $\overline{PRTT}_n^i - \overline{PRTT}_{n-1}^i$  is equal to  $PRTT_n^i - PRTT_{n-1}^i$  and  $\overline{ECT}^i$  is equal to  $ECT^i$  if computed for two successful queries of slave  $i$  occurred in two consecutive cycles.

### 5.1.2 Acyclic task

The acyclic task allows for the direct transmission of alarms from slave to master. A specific queue, used to store alarms, is assigned to each slave. When an alarm occurs, it is put in the queue and the acyclic task is responsible of its transmission. Several techniques could be used by the acyclic task to access the physical medium in order to transmit alarms [9]. The proposed experiments rely on the *immediate policy* described in [9]. In practice, when an alarm has to be sent, the slave, irrespective of the activity carried out of the network, tries to gain access to the channel and, if successful, transmits the alarm. This is possible thanks to IEEE 802.15.4 CSMA/CA technique which makes each device able to autonomously access the network.

## 5.2 Measurement System and Setup

The experimental session has been carried out by using the testbed sketched in Figure 5.2, which comprises a set of sensor nodes, a Personal Computer (PC) and a Arbitrary Waveform Generator (AWG).

The WSN enlists Tmote Sky wireless sensors (motes) available from Moteiv [5, 6], based on the IEEE 802.15.4 communication radio system [7] (for an in-depth description see Chapter 3). They are equipped with a USB port for programming, a 12-bits analog-to-digital converter, and a light sensor. The channel used is number 20 (centered at 2.450 GHz). In particular,  $N$  motes are used as slaves  $s_i$ , with  $1 \leq i \leq N$ , and one as master,  $m_0$ , all positioned at a height of 1 m from the ground floor, uniformly distributed along an half-circumference of radius  $r = 0.5$  m, and centered around  $m_0$ . The PC is used both to setup the network parameters at the beginning of the test, and to collect all the data coming from  $m_0$  via the USB port during the test. The PC is also equipped with a purposely developed software interface providing an user-friendly way to control the network and to perform the monitoring-process.

In the case of interfered environment, a suitable signal generator Agilent Technologies E4433B (250 kHz - 4 GHz) and a log periodic antenna EMCO 3146 have been used, oriented toward the testbed at a distance of  $d = 1$  m from it. Finally, motes have been set in CCA mode 1, with CCA threshold equal to -76 dBm (default value), light sensors enabled, and a transmission power equal to 0 dBm. In the analyzed configuration, such a transmission power provides a -40 dBm signal at the motes receivers. All the timestamp measurement have been performed using 32 bit software counters offered by the TinyOS stack.

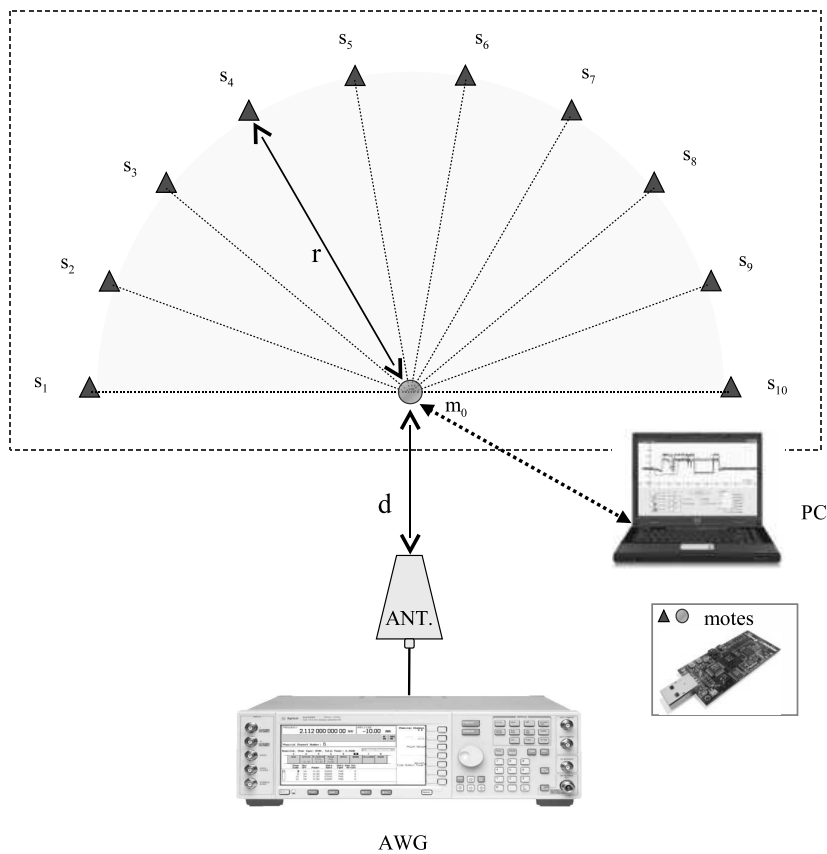


Figure 5.2: Testbed architecture in the case of ten slaves ( $N = 10$ ).

Polling timestamp measurements are performed using the master clock, and hence the uncertainty in the time measurement is negligible.

The uncertainty associated to the alarm latency estimates has been evaluated according to [18], and taking into account both synchronization error between master and slave's clock. A final uncertainty of alarm latency estimates approximatively of tenths of microseconds has been thus obtained (i.e. negligible with respect to the achieved alarm latency estimates).

### 5.3 Cyclic task experiments

An high number of experimental tests have been conducted under different configurations of the WSN, and in the absence/presence of interfering radio signals.

### 5.3.1 Without interference

A first set of tests has been carried out with a reduced number of slaves (three), namely,  $s_1$ ,  $s_5$ , and  $s_{10}$ , in the absence of interference signals. A total of 3000 polling cycles have been analyzed. The aim of the test, without alarm generation, was to verify the effectiveness of the cyclic transfer of data for variable values of the polling time which, for commodity, and without loss of generality, was set to the same value for all the slaves (i.e.  $T_p^i = T_p$ , and  $i = 1, 2, \dots, N$ ). As a first step, the lowest obtainable value of  $T_p$  has been taken into account. From the practical tests carried out, an increased percentage of failed pollings ( $\Phi_p$ ) for  $T_p$  less than 25-30 ms has been observed. The outcomes of such a test allowed us to determine a value of the polling time considered suitable for a stable operation of the WSN. Indeed,  $T_p = 30$  ms has been set and moved to evaluate the polling round trip times. Since in our experiments similar results were observed for all the slaves, the following figures refer to a specific slave and, for such a reason, the suffix  $i$  has been omitted.

In Figure 5.3, the estimated values of  $PRTT$  are shown. The behavior represents the measured  $PRTT$  PDF, whereas the vertical dashed line indicates the mean value.

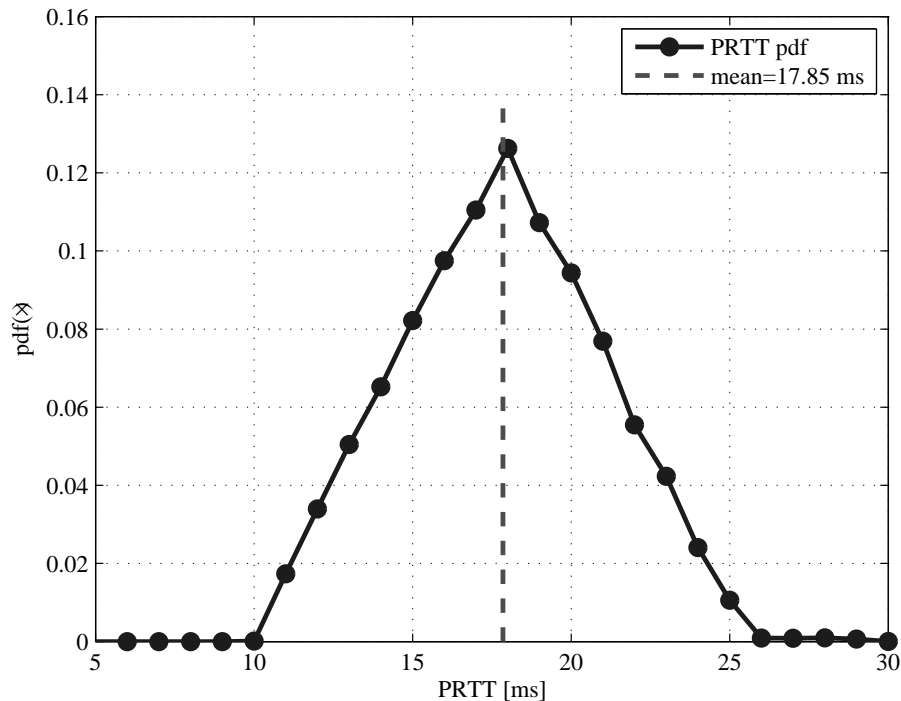


Figure 5.3: Measured  $PRTT$  probability density function.



As can be seen,  $PRTT$  never overcomes 26 ms, which represents a sort of threshold for the polling time. It is worth mentioning that similar values have been obtained also varying both the position of sensors and their number, from 1 to 10. The triangular shape confirms that  $PRTT$  is the sum of two uniform distributed variables, ( $\tau_{MS}$  and  $\tau_{SM}$ ) and, hence, that no interference affects the operation of the WSN.

In Figure 5.4 the measured experimental cycle time ( $ECT$ ) is shown. As can be seen,  $ECT$  behaves according to a Gaussian-like PDF, confirming the evidence of Eq. 5.4.  $PRTT$  and  $ECT$  estimates have obtained by using the counter embedded in the master, and counting the number of clock periods within the time intervals  $R_n - S_n$  and  $R_n - R_{n-1}$ , respectively. The measurement accuracy has been estimated from the knowledge of master clock accuracy, lower than 60 part-per-million, and by propagating the uncertainty according to [18]. A final uncertainty estimate lower than  $2 \mu\text{s}$  for both  $PRTT$  and  $ECT$  has been thus obtained (i.e. negligible with respect to the achieved estimates of  $PRTT$  and  $ECT$ ).

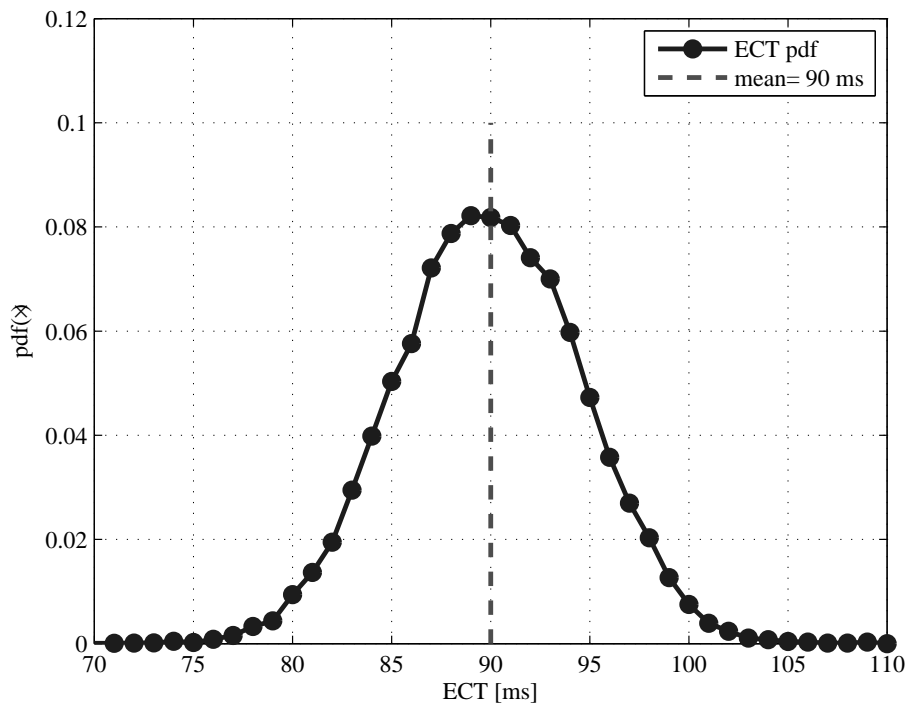


Figure 5.4: Measured ECT probability density function.

### 5.3.2 With interference

Further tests have been carried out at the same conditions of the above experimental case, but in the presence of *ad-hoc* generated interference. In particular, the following two types of disturbance have been experimented.

- a) Additive White Gaussian Noise (AWGN) interference of 5 MHz bandwidth and centered at the same frequencies of the WSN exploited channel.
- b) Impulsive interference, obtained modulating the amplitude of the aforementioned AWGN interference with a periodical sequence of bursts, with variable duty cycle,  $\lambda_B$ , and burst period,  $T_B$ , of either 100 or 300 ms.

Such two signals allow to emulate some typical disturbances arising in the presence of near operating WLANs and Bluetooth devices [2, 19, 20, 21]. In fact, their spectrum is typically noise-like and communication is commonly performed inside well-defined time slots. The chosen burst periods of interference (b) allow to investigate on critical worst-case situations in which the burst time duration can be in the order of, or much greater than, the WSN cycle time (here set to 90 ms).

In the case of interference (a), the percentage of failed polling,  $\Phi_p$ , has been assessed varying the interference power at slaves receiver,  $P_i$ , which has been in turn measured through the CCA power detector embedded into the motes. The obtained results are summarized in Figure 5.5.

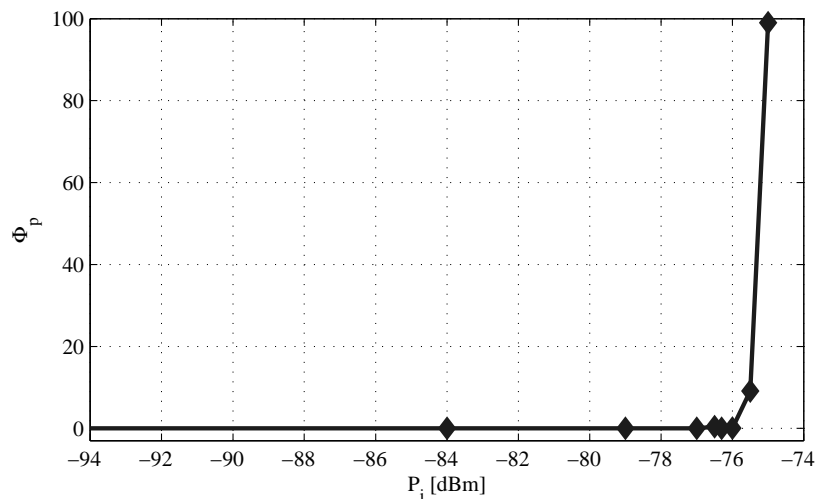
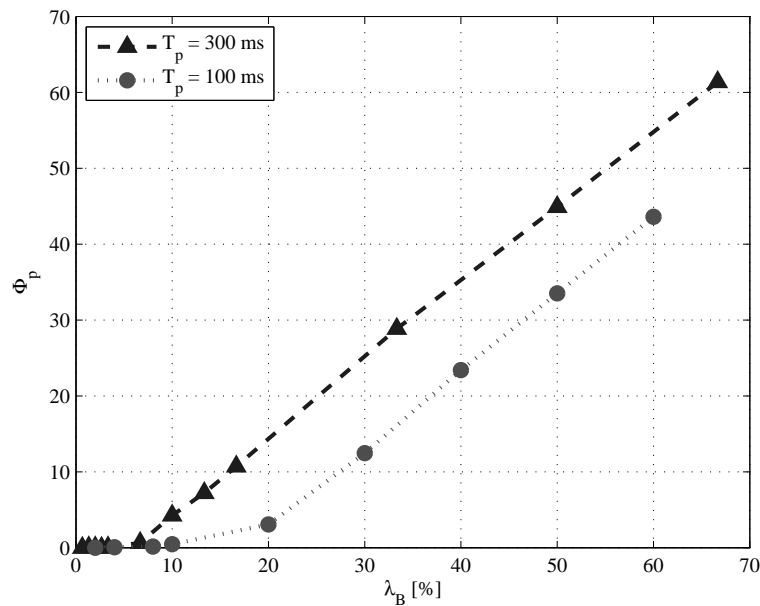


Figure 5.5: Percentage of failed polling,  $\Phi_p$ , vs received interference power  $P_i$ .

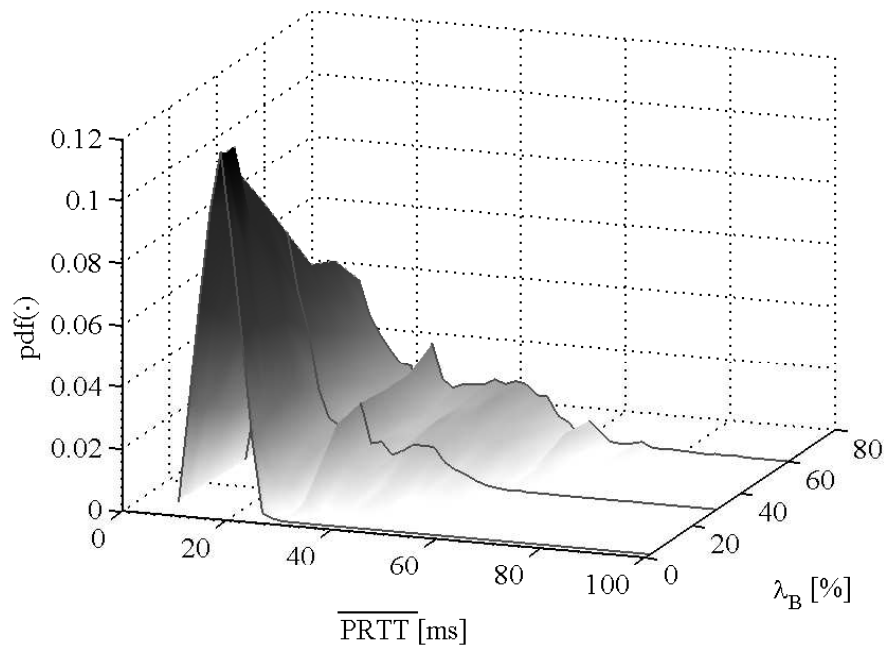


**Figure 5.6:** Percentage of failed pollings,  $\Phi_p$ , versus burst duty cycle, 3 slaves.

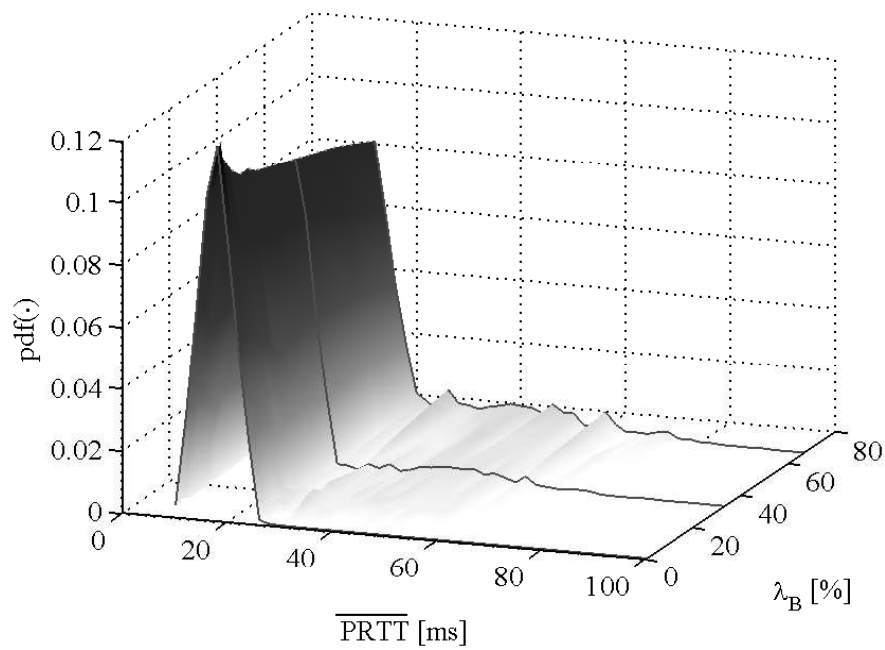
The diagram clearly shows that polling is successful until the received interference power  $P_i$  does not overcome the CCA threshold (-76 dBm). Beyond this limit,  $\Phi_p$  abruptly increases and the WSN performance considerably worsens. This result allows to deduce that the effects of interference on WSN performance may be very critical, even up to completely paralyze the whole network ( $\Phi_p = 100$  %). Furthermore it is interesting to observe that the performance worsening depends only on both the received interference power level ( $P_i$ ) and the CCA threshold, irrespective of the received useful power  $P_s$  which in our experiments was set to -40 dBm, i.e. 36 dB above the CCA threshold.

The effect of impulsive interference (b) has been investigated through experiments performed with the same configuration of WSN used for the previous test and with a received interference power intensity,  $P_i$ , in the absence of burst modulation, greater than the CCA threshold. The results obtained in terms of  $\Phi_p$  versus  $\lambda_B$  are shown in Figure 5.6.

The diagram shows that  $\Phi_p$  increases linearly, with unitary slope, when the duty cycle of bursts increases. Moreover, the performance of WSN are better for the case  $T_B = 100$  ms, with respect to  $T_B = 300$  ms. This is obvious, since in the former case the absolute durations of the bursts are shorter and, hence, the polling of slaves may eventually be successful. To this regard, it should also be noted that  $\Phi_p$  starts increasing just when the product  $\lambda_B \cdot T_B$  is comparable with or greater than *polling time*.



(a)



(b)

**Figure 5.7:** Measured  $\overline{PRTT}$  probability density function *vs*  $\lambda_B$  and for: (a)  $T_B = 100$  ms ; (b)  $T_B = 300$  ms.

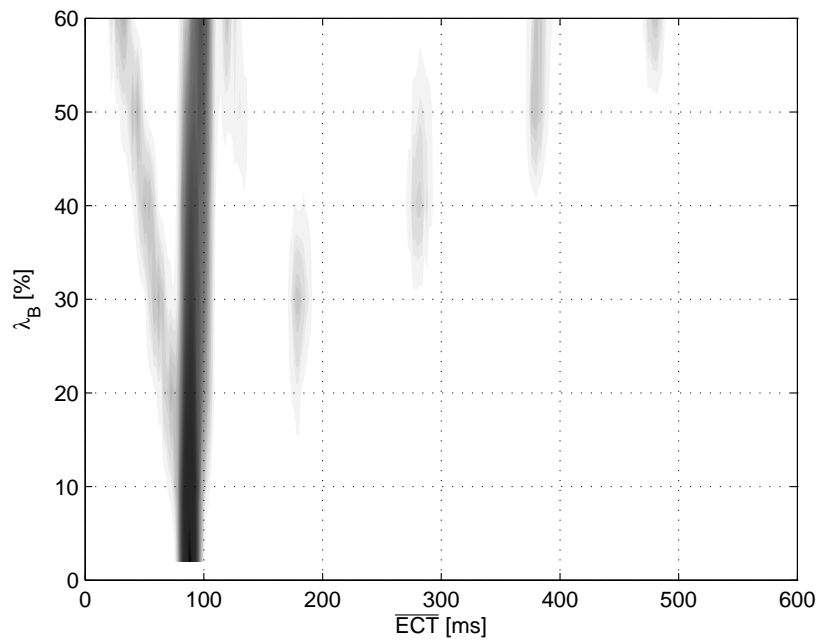
In fact, in this case, both master and slaves may not have sufficient time to exchange packets  $a$ ,  $b$ ,  $c$  and  $d$  of Figure 4.1 between two consecutive interference bursts. The figure also allows to note that, in case of impulsive interference, the WSN may tolerate AWGN interference even with power intensity  $P_i$  beyond the CCA threshold, but only if duty cycle is below a specific threshold. The PDFs of  $\overline{PRTT}$  and  $\overline{ECT}$  are shown in Figure 5.7 and Figure 5.8 respectively. Both the behaviors are given versus  $\lambda_B$ .

Some relevant observations can be drawn from Figure 5.7:

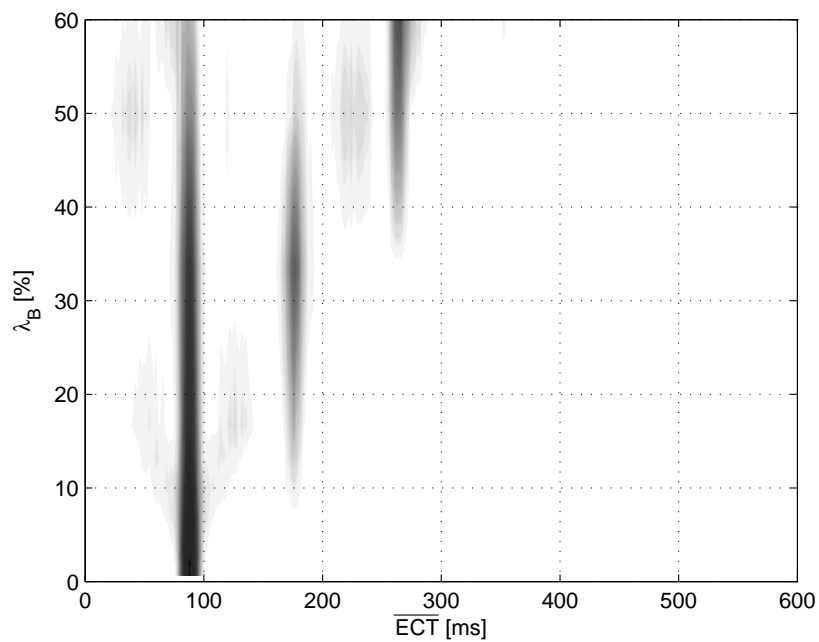
- With  $\lambda_B = 0\%$ , the two  $\overline{PRTT}$  behaviors of both Figure 5.7(a) and 5.7(b) have the same triangular shape of Figure 5.3 distribution. In fact, with  $\lambda_B = 0\%$ , interference is null, as in the case of Figure 5.3 experiments and, consequently, it results  $\overline{PRTT} = PRTT$ .
- In case  $T_B = 100$  ms, increasing  $\lambda_B$ , the  $\overline{PRTT}$  PDF loses the triangular shape, and spreads toward much greater values, beyond the fixed polling time of 30 ms. These are the cases in which  $PRTT$  can not longer be evaluated and the analysis is referred to  $\overline{PRTT}$ .
- In case  $T_B = 300$  ms, a less visible spreading of  $\overline{PRTT}$  values beyond the polling time of 30 ms is present. In particular, the  $\overline{PRTT}$  distribution maintains the triangular shape obtained with  $\lambda_B = 0\%$ , even for higher values of  $\lambda_B$ , unless a small part of  $\overline{PRTT}$  values is greater than 30 ms. This accounts for a sort of selectivity effect of the interference which, when occurs, is able to obscure entire polling windows (i.e. no frames are sent nor from the master, neither from the slave). This is in accord with the high percentage of failed polling shown in Figure 5.6

Some meaningful remarks can also be obtained from Figure 5.8:

- With  $\lambda_B = 0\%$ , the estimated distribution of  $\overline{ECT}$  is mono-modal and centered around 90 ms, which represents the term  $S_n - S_{n-1}$  in both eqs. 5.4 and 5.5. The single mode is essentially the one shown in Figure 5.4.
- For higher  $\lambda_B$ , the  $\overline{ECT}$  distribution presents one or more secondary modes, each located at values multiple of 90 ms. The presence of these modes indicate that some consecutive pollings have not been successfully executed: specifically, if  $\Pi$  consecutive pollings failed, a mode appears at  $(\Pi + 1) \times \text{cycle time}$ . For example, in Figure 5.8(a), the mode located at nearly  $\overline{ECT} = 270$  ms accounts for all those slaves whose polling failed during the two previous cycles.



(a)



(b)

**Figure 5.8:** Measured  $\overline{ECT}$  probability density function *vs*  $\lambda_B$  and for: (a)  $T_B = 100$  ms ; (b)  $T_B = 300$  ms.

- Comparing the two cases shown in Figure 5.8, for  $\lambda_B \leq 60\%$ , the number of  $\overline{ECT}$  values greater than 300 ms is negligible for  $T_B = 300$  ms. Nevertheless, the secondary modes of Figure 5.8(b) are rather larger and darker than those of Figure 5.8(a). This means that, with  $T_B = 300$  ms and  $\lambda_B \leq 60\%$ , a slave is quite always successfully polled within the first three polling cycles, while for  $T_B = 100$  ms the number of attempts can be up to five (with  $\lambda_B \leq 60\%$ ). On the other hand, the number of slaves not successfully polled at the first polling cycle (main mode) is higher for  $T_B = 300$  ms with respect to  $T_B = 100$  ms case.
- In both the diagrams of Figure 5.8, a secondary mode at  $\overline{ECT}$  values below 90 ms is observed. This is because, if the polling of a slave is delayed and the following is successful, then the resulting cycle time may be considerably shorter than the expected value of 90 ms.

## 5.4 Acyclic task experiments

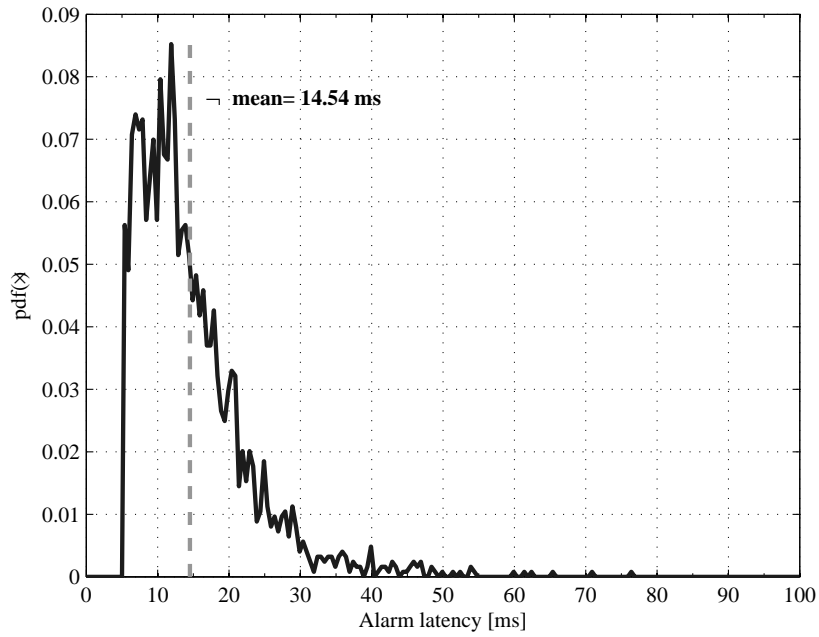
In order to verify the performance of the deployed WSN in presence of alarms, several experimental sessions have been carried out with the acyclic task enabled. Similarly to the previous section, experiments have been executed in both cases of presence and absence of interfering radio signals. The same testbed described in Figure 5.2 has been used.

For the generation of alarm events, a specific software task has been purposely designed and implemented on each sensor node. This task enables sensors to generate alarms according to a Poisson process.

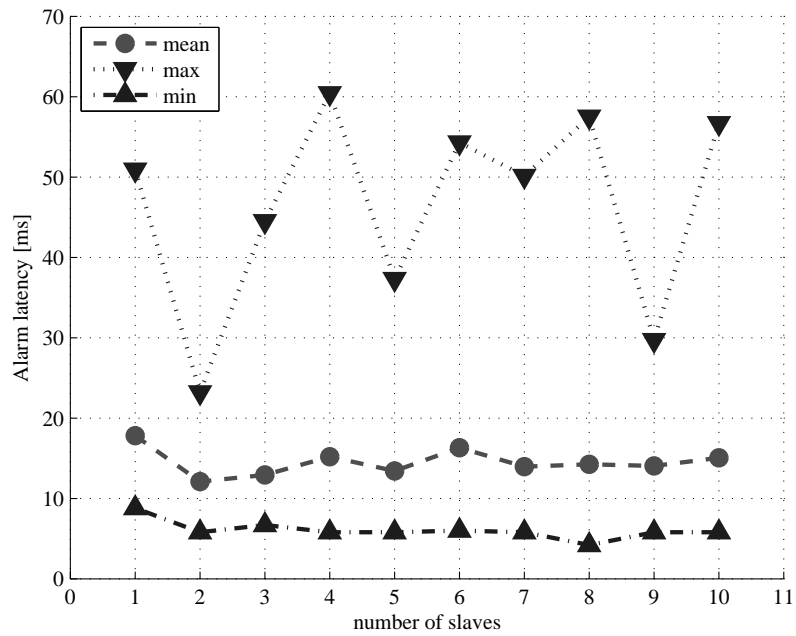
### 5.4.1 Without interference

In the first set of experiments, alarms have been generated in absence of interference with a variable mean inter-arrival time ranging from 500 to 5000 ms which are typical of industrial applications [22]. The same configuration of the previously described experiments, comprising one master and 3 slaves, has been adopted. A total of 3000 polling cycles have been carried out. The polling time of each slave was set to 30 ms. The PDF of estimated alarm latency values, for a mean inter-arrival time of 1000 ms is shown in Figure 5.9. Very similar results have been achieved for all the other inter-arrival times considered.

These results show that the alarm latency, regardless of the alarm generation rate, does not vary significantly and remains well bounded in the range 5 - 60 ms. Moreover, as can be noticed in Figure 5.9, more than 90 % of the collected values are below 30 ms.



**Figure 5.9:** Measured alarm latency probability density function ( $N = 3$ , mean inter-arrival time = 1000 ms).



**Figure 5.10:** Alarm latency vs number of slaves.



The same experiment has been repeated varying the number of slaves and keeping constant the mean inter-arrival time (1000 ms). The results obtained are shown in Figure 5.10 where, for commodity, a different representation has been used. From a straightforward comparison it follows that no appreciable differences with the results provided in Figure 5.9 are noticed. The diagram also shows that the maximum value of alarm latency is always lower than 60 ms, irrespective of the number of deployed slaves. This latter parameter represents an essential information to be taken into account in the design of a WSN aimed at managing alarm events, and can be easily estimated as described.

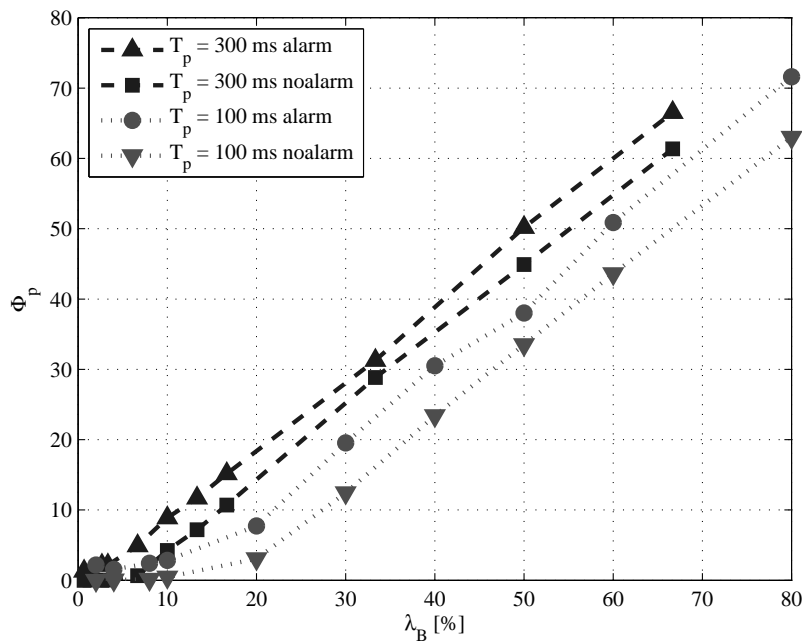
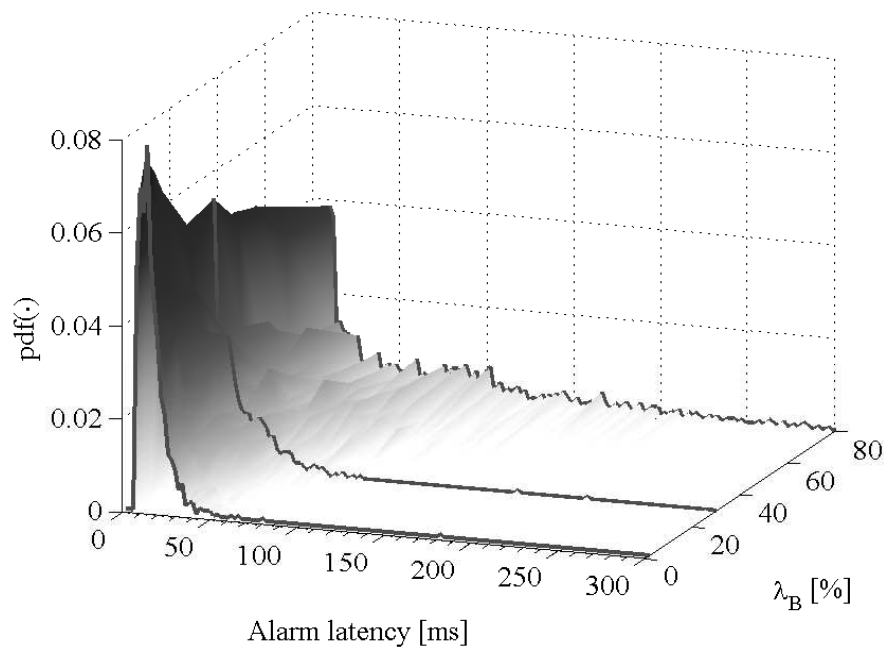


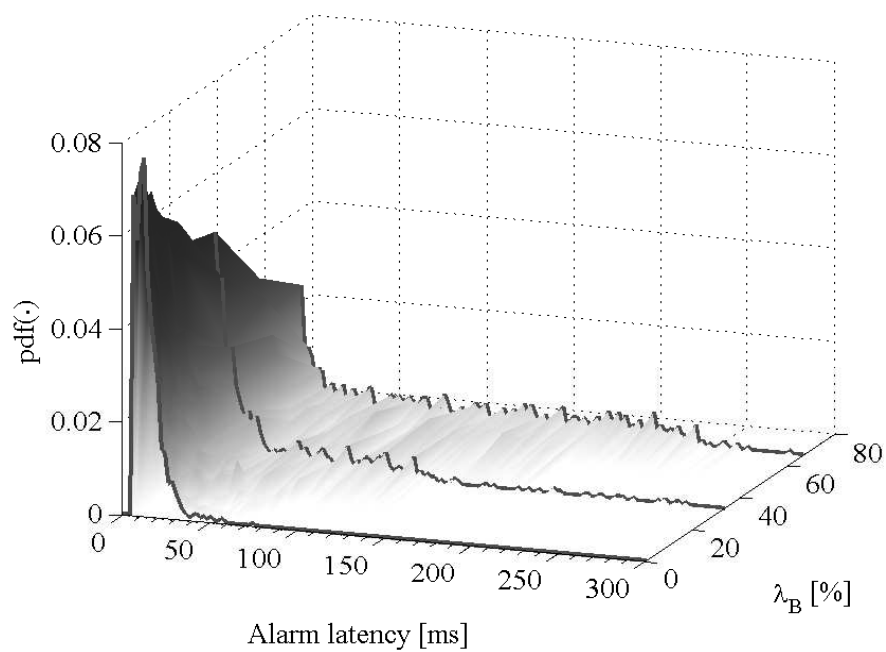
Figure 5.11: Percentage of failed pollings,  $\Phi_p$ , versus burst duty cycle, 3 slaves.

### 5.4.2 With interference

A second set of experiments has been carried out in presence of the impulsive interference (b), defined in Section 5.3.2. A first result is summarized in Figure 5.11, for the case of a mean alarm inter-arrival time of 1000 ms. In the figure, the estimated values of the percentage of successful polling,  $\Phi_p$ , are given versus  $\lambda_B$ , and for the two periods  $T_B = 100$  ms and  $T_B = 300$  ms respectively. The diagram shows (recalling Figure 5.6) that the presence of alarms degrades the behavior of the cyclic task, increasing the percentage of failed pollings. In Figure 5.12, the estimated alarm latency PDFs are given.



(a)



(b)

**Figure 5.12:** Measured alarm latency probability density function *vs*  $\lambda_B$  and for: (a)  $T_B = 100$  ms ; (b)  $T_B = 300$  ms.

They have been obtained, also in this case, for a mean alarm inter-arrival time of 1000 ms, versus  $\lambda_B$  and for the two aforementioned burst periods. As obvious, the increase of  $\lambda_B$  leads to alarm latencies greater than those shown in Figure 5.9. In particular, it may be observed that, for  $T_B = 100$  ms, and  $\lambda_B \rightarrow 100\%$ , the alarm latency PDF increases significantly, assuming values in the range 50 – 100 ms. This happens because alarms occurring during a burst period (which in this case is significantly long) may be delivered only at the end of that period. Similarly, for  $T_B = 300$  ms, and  $\lambda_B \rightarrow 100\%$ , the alarm latency PDF falls mainly in the range 50 – 300 ms.

## 5.5 Conclusions

In this chapter, a general and meaningful experimental case study has been presented, involving a real life IEEE 802.15.4 WSN using CSMA/CA employed for industrial applications, under the presence of interference. The analysis has been performed with the network operating according to a both cyclic (periodical polling) and acyclic task (alarm management).

The obtained results have clearly underlined the usefulness of preliminary measurements at prototype level, in order to detect the occurrence of possible interference effects and understand how to optimize the network setup. Examples of how to measure some key parameters of the network have been given. For instance, in the cyclic task, the minimum polling time that can be set can be measured assessing the percentage of failed polling ( $\Phi_P$ ) versus polling time, while the presence of interference can be detected by analyzing the probability density function of the polling round trip time ( $\overline{PRTT}$ ). The maximum level of interference power beyond which the network abruptly worsens its performance can be estimated measuring  $\Phi_P$  versus the interference power intensity (in the case of AWGN interference), or versus duty cycle ( $\lambda_B$ ) (in the case of burst-like interference). The failure of a polling cycle in the reading of a sensor can be instead clearly recognized by the appearing of secondary modes in the diagrams of measured experimental cycle-time ( $\overline{ECT}$ ) versus  $\lambda_B$ . In the acyclic task, it has been observed that the alarm latency, regardless of the alarm generation rate and the number of slaves, does not vary significantly and remains well bounded in a fixed range.

All the presented measurement activities can be also applied to already WSNs installed and operating in a real industrial environment, *e.g.* for distributed sensing of physical quantities like temperature, humidity, pressure.



# Chapter 6

## *Optimizing a WSN based wireless control system against interference*

**A**FTER investigating performance of the wireless control system proposed in Chapter 5 in the presence of synthetic interference, in this Chapter a real-life application of Factory Sniffer system is tested.

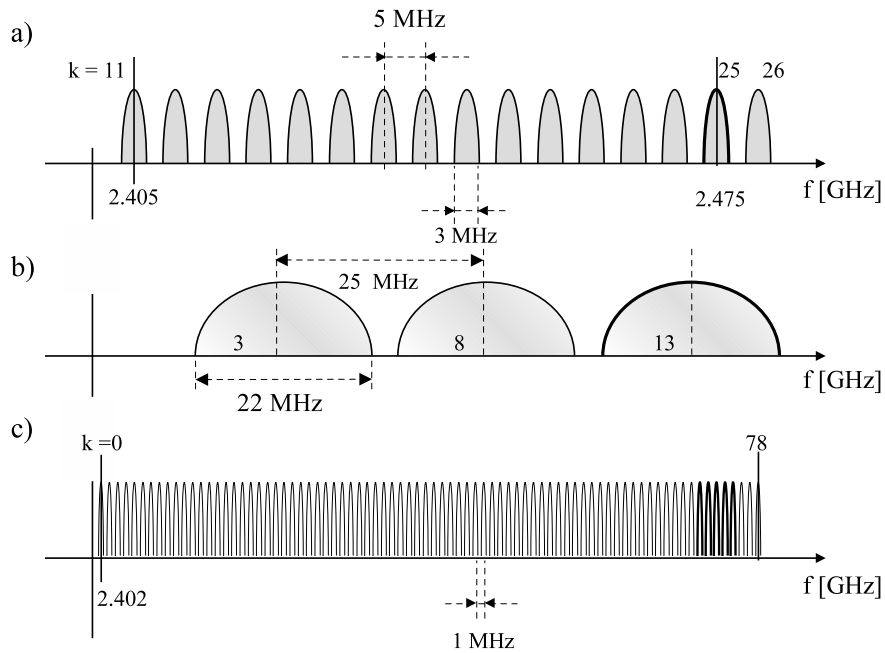
In particular, an in-depth analysis of the CCA modes provided by the IEEE 802.15.4 standard (see Section 2.2.3) is described.

It is shown how the choice of the CCA mode can affect the performance of a IEEE 802.15.4 network in the presence of interference: as can be easily understood, CCA mode 1 is the most sensible to interference, leading often to worse performance.

The aim is to evaluate the feasibility of a wireless control system in the presence of heavy interference, such as IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 (ZigBee) two-node networks. All the material described in this chapter is taken from [23].

### **6.1 Standards Overview**

As already said, 2.4 GHz ISM band is crowded of radio appliances and band superposition is unavoidable. Figure 6.1 sketches this phenomenon: IEEE 802.15.4 standard has been extensively described in Chapter 2, while in the following a brief summary on IEEE 802.15.1 (Bluetooth) and IEEE 802.11 (WiFi) is provided. The band superposition is a worst case situation: transmission is always packet-based, so time scheduling may help avoiding disruptive interference.



**Figure 6.1:** Frequency channels of (a) IEEE 802.15.4, (b) IEEE 802.11 and (c) IEEE 802.15.1 inside the ISM band.

### 6.1.1 IEEE 802.15.1 standard

Another standard deployed for WPANs is IEEE 802.15.1 [19], in the following simply referred with the commercial term Bluetooth. This standard defines the first two layers, PHY and MAC of the protocol stack. Bluetooth systems operate in the 2.4 GHz ISM band. In a majority of countries, the range of this frequency band is 2400 MHz - 2483.5 MHz. As depicted in Figure 6.1(c), the frequency spectrum is divided into 79 channels, spaced 1 MHz apart and characterized by carrier frequencies  $f_k$ , with  $f_k = 2402 + k$  MHz and with  $k = 0, \dots, 78$ . The most interesting feature of Bluetooth systems is the transmission technique. In fact Bluetooth employs a technique called Frequency Hopping Spread Spectrum (FHSS) whereby the carrier frequency of the transmitter changes up to 1600 times per second.

Another interesting issue of Bluetooth is the transmitted power. In particular, devices are divided into three classes, each associated to a maximum transmitted power level. The devices deployed in the following experiments are in class 1, *i.e.* with a maximum 100 mW power and power control.

### 6.1.2 IEEE 802.11 standard

The IEEE 802.11 [1] is a WLAN standard defining a total of 14 frequency channels, each of which characterized by 22 MHz bandwidth. As sketched in Figure 6.1(b), these channels are partially overlapped, and only three channels at a time, e.g. 3, 8, and 13, can be used without mutual interference.

In a typical communication, an Access Point (AP) transmits periodically a frame called *beacon*, which contains information like a network identifier (ID), the beacon and channel parameters, and other traffic information. Each network station receives the beacon, and if it is intentioned to access the network, it sends a request for authentication. Once authenticated, a station may communicate to the AP or *vice versa* according to a CSMA/CA protocol. Usually, to assess the channel status, CCA technique in *mode 1* is implemented [1]. In this mode, the channel is assumed idle if the channel power level is below a given user selectable threshold, called CCA threshold, otherwise it is assumed busy.

## 6.2 Industrial Monitoring protocol

In this Chapter the Factory Sniffer system is used to perform all the test needed. The monitoring protocol described in Section 5.1 is here recalled and the notation slightly simplified to better describe a single master-slave link.

The protocol performs two classic tasks: a periodical polling of each slave for receiving data and an asynchronous alarm transmission, able to handle critical events from peripheral sensors (acyclic task).

### 6.2.1 Cyclic task

The cyclic task consists of a round robin polling of each slave. In the case of a single slave, the master node queries the slave every *polling time*,  $T_p$  maintaining a sort of time-slot division. The master samples a virtual sensor that is physically mounted on the slave: this task is basically an abstraction of the physical sampling. A key feature of a sampling process is to assure an almost fixed sampling period ( $T_p$ ), especially if the industrial monitoring application is involved in a control chain.

In Figure 6.2, a simplified version of a master/slave communication is shown ( $n$  indicates the sequence number of the corresponding cycle). Once gained the channel, the master begins querying the slave with a *request* packet (containing information like the destination node address, the sequence number of the performed cycle and time stamps).

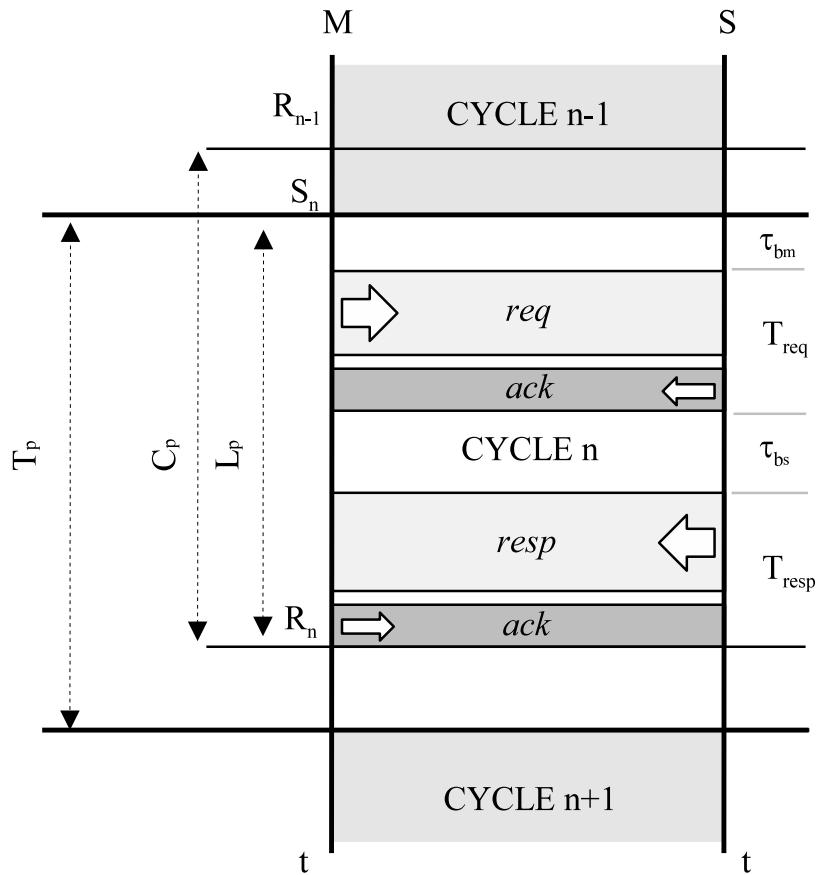


Figure 6.2: Packets exchange between master and slave inside a polling time.

The slave, after a fixed delay defined by the IEEE 802.15.4 standard (12 symbol periods), replies with an **ACK**, and then sends a message containing the process data, after a second carrier sensing. In the case of correct reception, the master issues an **ACK** and waits for the expiration of the polling time before restarting the next cycle. If the handshake is not performed correctly (*i.e.* either the master or the slave does not receive the **ACK** frame), then the polling of that slave is considered failed and no retransmission is attempted. Within a cycle, the entire polling process consists of the following steps. At the instant  $S_n$ , the master starts the *request*: a first variable period,  $\tau_{b_m}$ , is used to gain the channel and the transmission of the request packet plus the **ACK** takes another fixed period,  $T_{req}$ . After that, the slave performs several tasks and, once obtained the channel (after a variable  $\tau_{b_s}$  period), sends the *response* to the master (the transmission of the response plus the **ACK** takes a fixed period,  $T_{resp}$ ). Hence,  $\tau_{b_m}$  and  $\tau_{b_s}$  are the *initial backoff* period needed to access the channel by the master and the slave, according to the CSMA/CA protocol.



The master completes the polling cycle at the instant  $R_n$ , that is:

$$R_n = S_n + \tau_{b_m} + T_{req} + \tau_{b_s} + T_{resp}. \quad (6.1)$$

The difference between  $R_n$  and  $S_n$  is the *polling latency*,  $L_p$ , which represents the time employed by the master to execute a complete query of a slave or, equivalently, the delay between the actual sampling instant and the ideal one. It may vary from a cycle to another. The general expression of  $L_p$  is:

$$L_p = R_n - S_n = \underbrace{\tau_{b_m} + \tau_{b_s}}_{random} + \underbrace{T_{req} + T_{resp}}_{deterministic}. \quad (6.2)$$

The master samples his virtual sensor connected to the slave at  $S_n$  but receives the wanted data sample only at  $R_n$ . From the study of this parameter interesting results can be deduced about the interference effects on the network. For instance in the absence of interference, only one channel access attempt is performed by both master and slave and, consequently,  $\tau_{b_m}$  e  $\tau_{b_s}$  are expected to be uniformly-distributed independent random variables. Consequently, a triangular distribution of  $L_p$  values is also expected. Conversely, in the presence of interference, more attempts to access the channel are possible, since the channel may be erroneously sensed as busy. Consequently, both  $\tau_{b_m}$  e  $\tau_{b_s}$  are no longer expected to be uniformly-distributed independent random variables and, hence, different distribution of  $L_p$  values are expected, depending on the interference characteristics.

Another fundamental parameter is the effective cycle time ( $C_p$ ), defined as:

$$C_p = R_n - R_{n-1} = \underbrace{S_n - S_{n-1}}_{deterministic} + \underbrace{(L_{p(n)} - L_{p(n-1)})}_{random}. \quad (6.3)$$

Theoretically,  $C_p$  should be as fixed as possible, because it is the effective polling period visible at the application layer that a control chain can use.

As shown in equation (6.3),  $C_p$  accounts for two different contributions: the first one,  $S_n - S_{n-1}$  is deterministic since it represents the time elapsed between two subsequent transmissions of the request frame from the master to the same slave. The second contribution,  $L_{p(n)} - L_{p(n-1)}$ , is a random variable, since it is given by the difference between  $L_p$  values evaluated for cycles  $n$  and  $n - 1$ .

In this case, a different distribution of  $C_p$  values is expected depending whether the interference is present or not. In particular, the outcoming experimental Probability Density Function (PDF) will be gaussian-like only in the absence of interference, since in this case both  $L_{p(n)}$  and  $L_{p(n-1)}$  have triangular shaped PDFs and their difference leads to a gaussian-like PDF.

### 6.2.2 Acyclic task

The acyclic task allows for the direct transmission of alarms from slave to master. When an alarm occurs at the slave side, it is put in a specific queue and the acyclic task is responsible of its transmission. Several techniques could be used by the acyclic task to access the physical medium in order to transmit alarms [9]. The proposed experiments rely on the *immediate policy* described in [9]. In practice, when an alarm has to be sent, the slave, irrespective of the activity carried out of the network, tries to gain access to the channel and, if successful, transmits the alarm. This is possible thanks to IEEE 802.15.4 CSMA/CA technique which makes each device able to autonomously access the network. Also this transmission is acknowledged by an ACK frame as in the case of polling task. The alarm can not be lost, so the transmission is retried until success. This queued policy does not allow to predict any upper bound to the *alarm latency*,  $L_a$ , that is the time that the slave takes to successfully deliver the alarm to the master.

## 6.3 Measurement System and Setup

The experimental session has been carried out by using the testbed sketched in Figure 6.3, which comprises a two-nodes IEEE 802.15.4 network connected to a PC and a two-node interfering network. For the sake of simplicity, a single link network has been chosen since the monitoring protocol induces a time scheduling that gives similar results regardless of the number of slaves, as reported in [16]. The experiments have been performed in a non-anechoic room in order to emulate a real life environment. Preliminary measurements made with a Agilent E4407B spectrum analyzer inside the used band have been performed to assure the absence of external uncontrolled interferences.

The WSN enlists t-mote sky wireless sensors (motes) available from Moteiv [5, 6], based on the IEEE 802.15.4 communication radio system [7].

They are equipped with an USB port for programming, a 12-bits ADC, and a light sensor. The channel used is number 25 (centered at 2.475 GHz).

This channel has been chosen suitably far away from external interference due to near operating WiFi Access Points. The master, M, and the slave, S, are positioned at a height of 1 m from the ground floor, with a distance  $r = 1$  m from each other, assuring at least a -50 dBm received signal power, *i.e.* far above the receiver's sensitivity. The PC is used both to setup the network parameters at the beginning of the test, and to collect all the data coming from M via the USB port during the test.

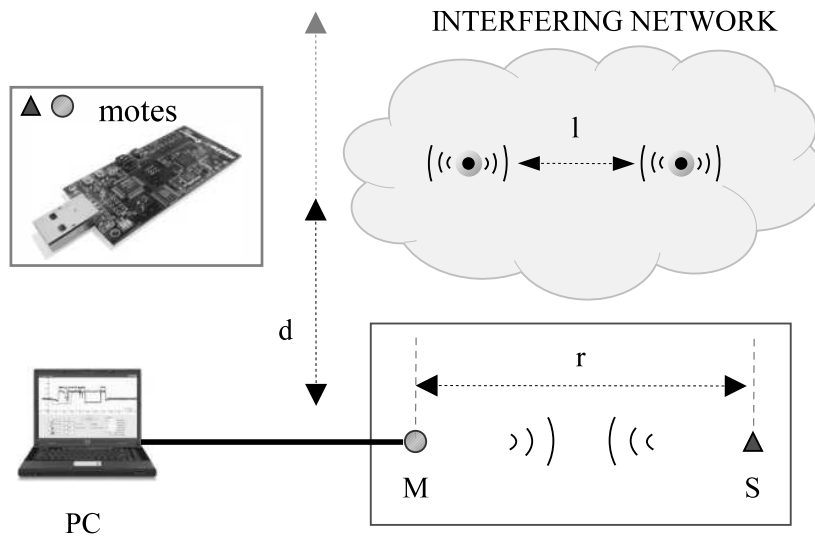


Figure 6.3: Testbed architecture.

The PC is also equipped with Factory Sniffer (see Section 4.1) to control the network and to perform the monitoring-process. The slave's clock is also synchronized to the master's one at each polling cycle, so that the polling and alarm latencies can be correctly measured.

Each experimental session consists of 10000 cycles, with a polling time  $T_p = 30$  ms. This value guarantees a null packet loss in absence of alarms and interference [16]. Alarms are software generated by a Poisson process with mean alarm inter-arrival time of 1000 ms (*i.e.* a stressful condition of nearly one alarm/second). All the collected data are post processed with Matlab.

The following CCA modes have been used for the sensor network, in order to evaluate the immunity of the network with respect to interference:

- **CCA 1:** the channel is sensed busy if an arbitrary interfering source causes:  $RSSI \geq CCA_{TH}$ , with  $CCA_{TH} = -77\text{dBm}$  (default value).
- **CCA 2:** the channel busy only if another IEEE 802.15.4 transmission is revealed (Carrier detection).
- **NO CCA:** motes do not perform CCA, and the backoff basic period (BP defined in Section 2.3.1) is set to zero, switching the backoff delay off.

The interfering network has been set-up with different communication standards. The physical layout (distance  $d$  and  $l$ ) of the interfering network has been chosen to emulate a typical indoor factory environment and according to specific transmission ranges of each standard:

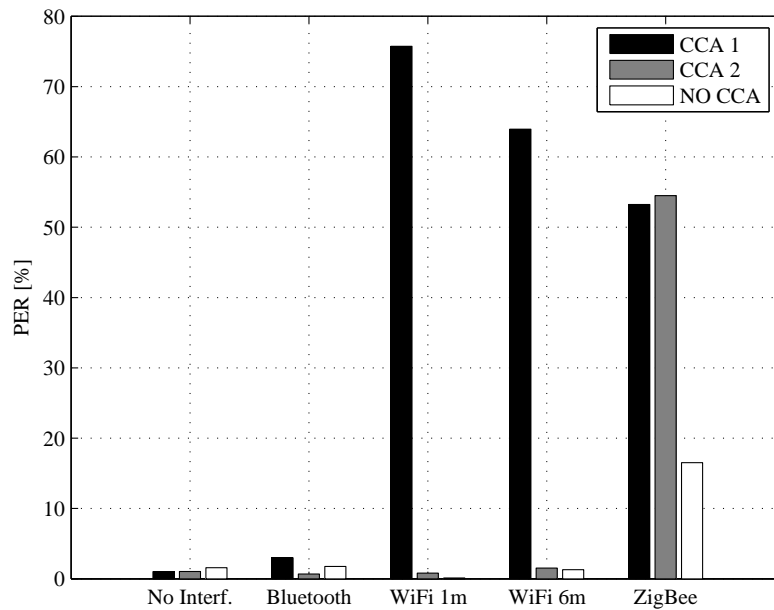
- **No Interference:** no interfering network is active. This case is used to evaluate best achievable performance.
- **Bluetooth:** a file transfer (always on) between two Bluetooth (IEEE 802.15.1) class 1 usb adapters (100 mW power) is performed at the maximum achievable rate. The distance between this network and WSN is  $d=1$  m, while the distance between the two Bluetooth devices is  $l=2$  m.
- **WiFi:** a file transfer (always on) between a WiFi (IEEE 802.11g) access point (Netgear KWGR614) and a wifi usb adapter (D-link DWLG122) is performed at the maximum achievable rate (nominal 54 Mbps). In order to evaluate the influence of the distance on the WSN performance, two distances are considered:  $d=1$  m and 6 m. The two WiFi devices are separated by a distance  $l=2$  m and the chosen channel is 13 (overlapping the transmission channel of the WSN).
- **ZigBee:** a same network of the one under test is used as a further interferer. It deploys the same channel, in CCA mode 1 and with the same  $T_p$  and mean alarm interarrival time. It operates at a distance  $d=0.5$  m, and with  $l=1$  m between the two ZigBee devices.

## 6.4 Experimental Results

### 6.4.1 Packet Error Rate

A first set of results from the experiments have been obtained in terms of PER. For every interference configuration described in section 5.2, three tests have been performed varying the CCA mode. The results are summarized in Figure 6.4.

As can be seen, with no interference, only minimum values of PER occur, which are due to higher-priority transmissions, like alarms. It is worth noting that the CCA has little or no effect on performance: in fact in the mode NO CCA the PER does not worsen significantly (1.5%). Bluetooth does not vary significantly the performance of the WSN. This is due to its frequency-hopping nature and short time-slots, which do not interfere significantly the WSN cycles. Also in this case the CCA mode does not play a key role. In fact, the maximum PER is 2.5% for CCA mode 1, *i.e.* even greater than the PER in NO CCA mode ( $\approx 2\%$ ). The effect of CCA mode is instead really visible in the case of WiFi interferers. In fact, the WiFi channel always overlaps the WSN one, and lets very few “windows” into which ZigBee can transmit.



**Figure 6.4:** Measured PER for all the considered configurations.

For both the considered distances (1 and 6 m) from the WSN, the effects are similar and a great distance implies only a weak improvement of performance. The network in CCA mode 1 is heavily impaired by this WiFi interferer, because the received power (about -50 dBm) is always above the CCA threshold (-77 dBm). The behavior in CCA 2 and NO CCA modes leads to a similar PER. In fact, the interferer is not an IEEE 802.15.4 signal and so CCA 2 senses the channel free. The performance improvement in these modes is relevant: for WiFi 1 m away, the PER changes from 75% to less than 0.5%. In this case the transmission depends only on the Signal To Interference plus Noise Ratio (SINR), that is far above the value for a correct reception. In the case of a ZigBee interferer, a heavy packet loss is visible: long ZigBee packets do not allow the network under test to correctly perform the polling and lead to a PER of about 55% for both CCA 1 and 2. In fact, in this case the interferer is of type IEEE 802.15.4 and its power is above the threshold, causing a busy-channel both in CCA 1 and 2 modes. Therefore, in this case WSN performance are weakly improved by disabling the CCA (PER= 16.5%).

### 6.4.2 Polling Cycle Time

Classical performance evaluation methods consider only packet error rate. For real-time systems, as the monitoring system above described, packet loss is not the only important metric. For the correctness of a control chain above a sampling process it is fundamental to maintain as stable as possible the sampling period.

In this subsection, the results in terms of PDF of the polling cycle  $C_p$ , *i.e.* the effective polling period, are reported upon the varying of CCA modes. The reported plots are normalized histogram, made of the large amount of collected data (10000 samples). They provide an almost complete statistical knowledge of the random variable  $C_p$ .

As stated in section 6.2.1, the expected PDF in both CCA 1 and 2 modes is gaussian-like (cubic splines), due to the convolution of four equal distributed uniform random variables (due to the initial back-offs). The curve is centered at 30 ms, with a minimum value of 10 ms and a maximum value of 50 ms. In Figure 6.5 the polling cycle PDF in the case of no interference is shown. As predicted, for CCA mode 1 and 2 the back-offs of CSMA/CA protocol lead to an identical shape of the two curves. With CCA disabled, the presence of an impulsive PDF centered exactly at 30 ms (the chosen  $T_p$ ) means an almost constant and deterministic value of  $C_p$ .

In fact back-offs are disabled and, with no interference, the polling works correctly at the first attempt. For the sake of clarity, in the following only subsequent cycles are analyzed. The cycles comprising a packet loss are ignored to prevent secondary modes in the PDF.

Although not reported, even with a Bluetooth interferer the PDFs remain the same of Figure 6.5. This interferer only increases the PER without modifying substantially the polling latency.

In Figure 6.6 the effect of a 6 m far away WiFi interference is shown (the same values are obtained with distance 1 m). As expected, CCA 2 and NO CCA modes results are quite the same of those observed in the non-interfered example. In these cases the channel is sensed free (because the interferer is not IEEE 802.15.4) or not sensed at all and so the back-offs are not increased.

Conversely, in the case of CCA 1 the channel is often sensed busy and the CSMA/CA protocol tries to wait to by-pass the interference. This PDF reveals that many attempts to gain the channel have been tried, causing a great dispersion of the PDF that now spans from 10 to 95 ms and is much flatter than the one obtained in the absence of interference. The results summarized in Figure 6.7 reveal that in this case also with NO CCA the polling cycle is no longer constant, probably due to the large amount of failed pollings.

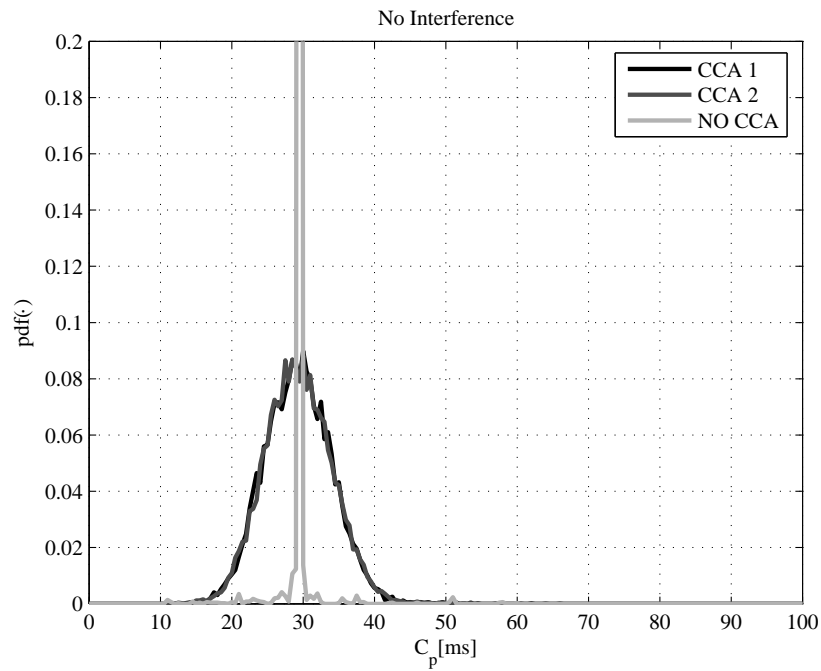


Figure 6.5: Probability density function for Polling cycle ( $C_p$ ) with no interference.

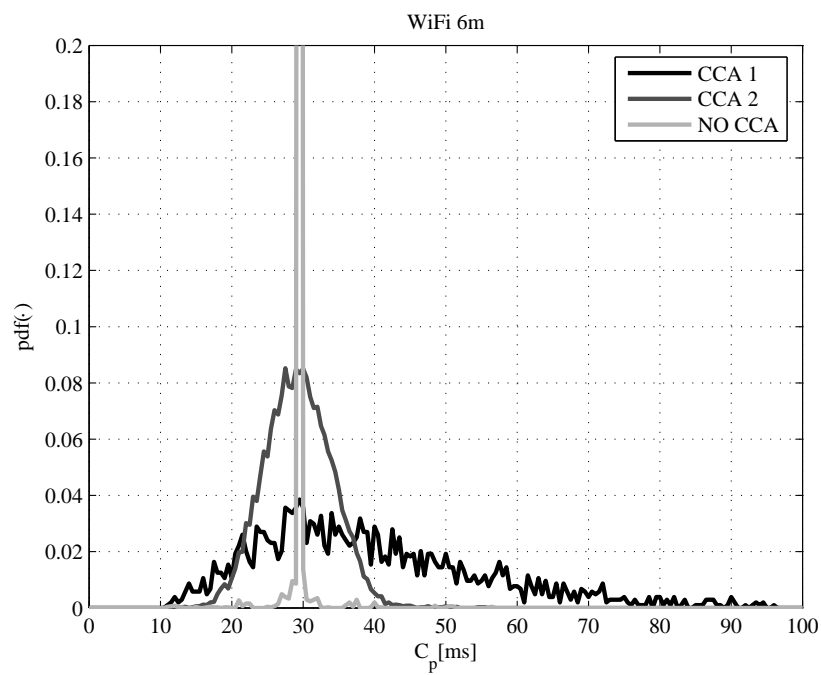
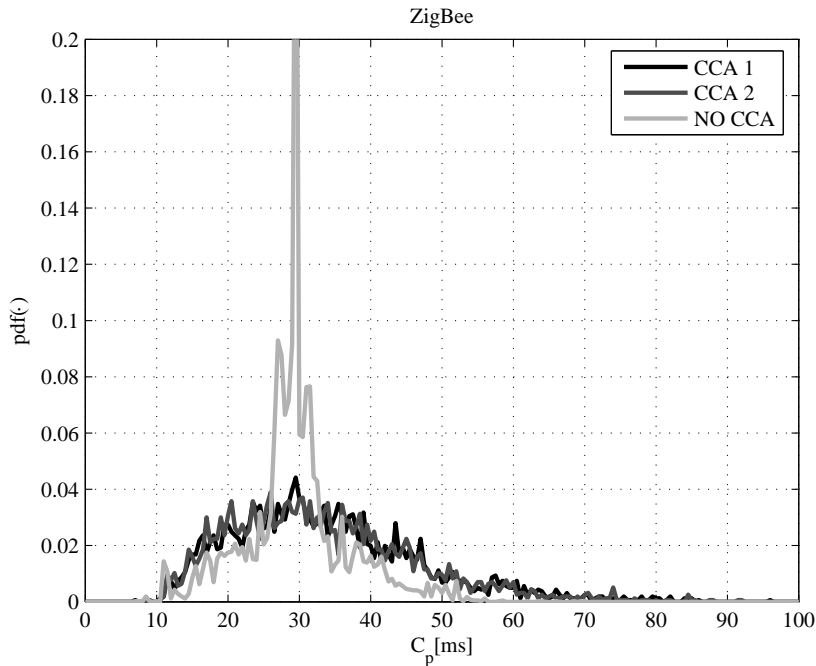


Figure 6.6: Probability density function for Polling cycle ( $C_p$ ) with WiFi interference.



**Figure 6.7:** Probability density function for Polling cycle ( $C_p$ ) with ZigBee interference.

As expected, both modes 1 and 2 lead to the same results, because with a IEEE 802.15.4 interferer above the CCA threshold the channel is sensed busy by both methods. With a ZigBee interferer the choice of NO CCA is no longer an efficient solution, even if it gives a better behavior.

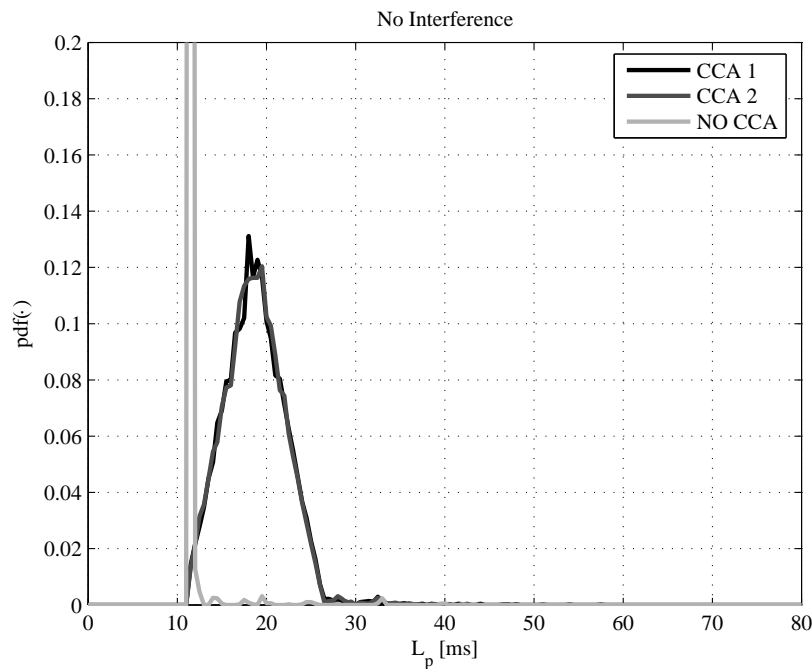
## 6.5 Polling Latency

Another important parameter in a sampling process is the polling latency. If the measurand is a fast-changing process the polling latency may not be negligible. Similarly to what happens for  $C_p$ , even  $L_p$  is heavily affected by the choice of CCA mode. The reported plots are normalized histogram, made of the large amount of collected data (10000 samples) as for  $C_p$ .

As stated in section 6.2.1, the expected PDF in both CCA 1 and 2 modes is triangular, due to the convolution of two equal distributed uniform random variables (due to the initial back-offs). The curve is centered at 18 ms, with a minimum value of 11 ms and a maximum value of 27 ms. In Figure 6.8 the polling latency PDF in the case of no interference is shown. CCA mode 1 and 2 lead to a similar shape of the two curves. NOCCA mode causes an impulsive PDF centered at 11.5 ms.



This is the almost deterministic delay due to hardware and software latencies of the microcontroller and the transceiver. Although not reported, even with a Bluetooth interferer the PDFs remain the same of Figure 6.8. This interferer only increases the PER without modifying substantially the polling latency.

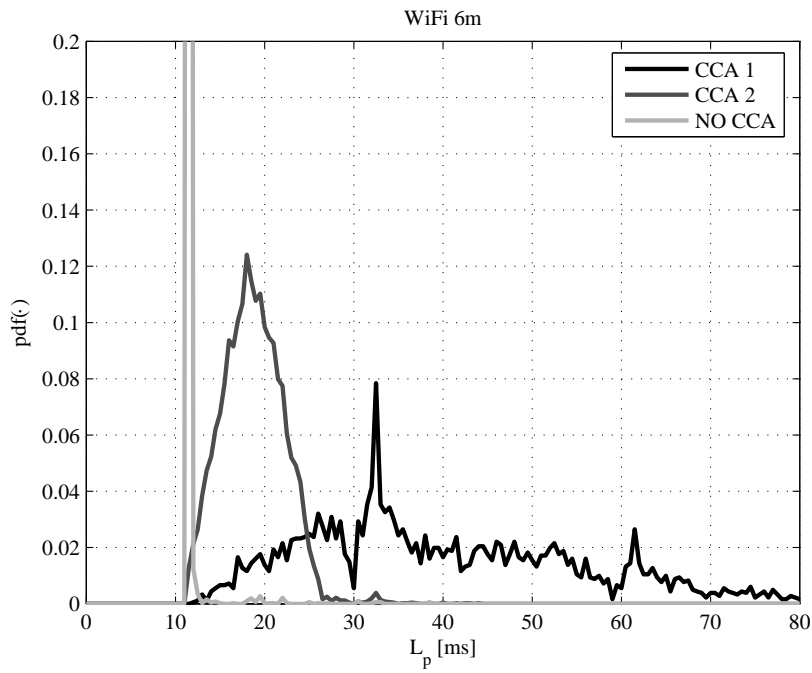


**Figure 6.8:** Probability density function for Polling latency ( $L_p$ ) with no interference.

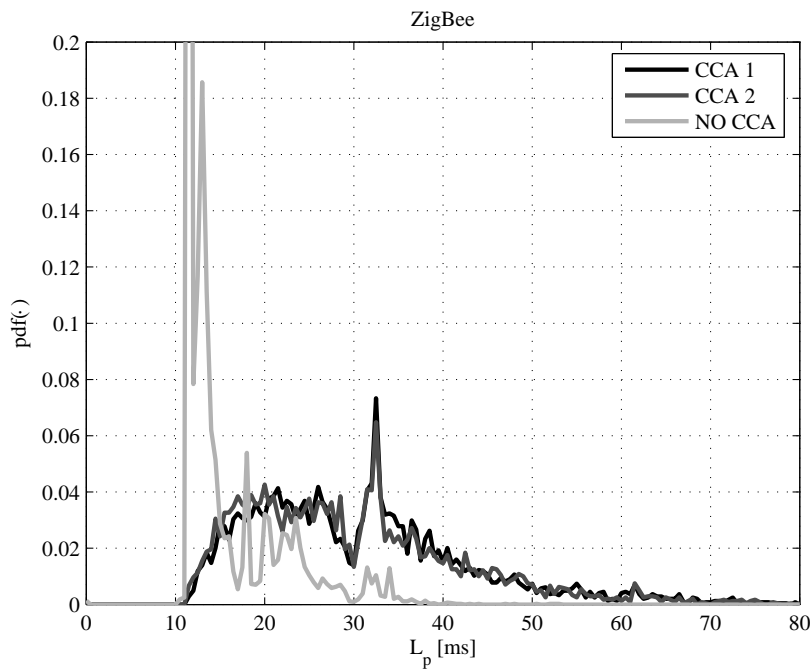
In Figure 6.9 the effect of a 6 m far away WiFi interference is shown (the same values are obtained with distance 1 m). As expected, CCA 2 and NO CCA modes results are quite the same of those observed in the non-interfered example. In these cases the channel is sensed free (because the interferer is not IEEE 802.15.4) or not sensed at all and so the back-offs are not increased.

In the case of CCA 1 the channel is sensed busy and the CSMA/CA protocol tries to wait to overcome the interference. The PDF has a great dispersion from 10 to 80 ms revealing many attempts to gain the channel.

Figure 6.10 reveal that the ZigBee interferer, even with NO CCA, causes wide spread in the PDF. As expected, both modes 1 and 2 lead to the same results, because with a IEEE 802.15.4 interferer above the CCA threshold the channel is sensed busy by both methods. With a ZigBee interferer the choice of NO CCA is no longer a efficient solution, even if gives a better behavior.



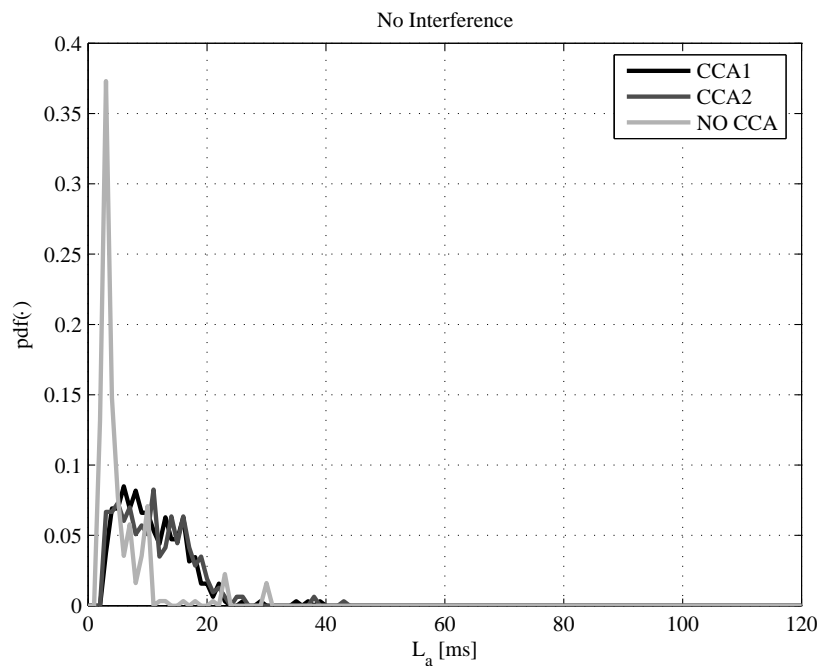
**Figure 6.9:** Probability density function for Polling latency ( $L_p$ ) with WiFi interference.



**Figure 6.10:** Probability density function for Polling latency ( $L_p$ ) with ZigBee interference.

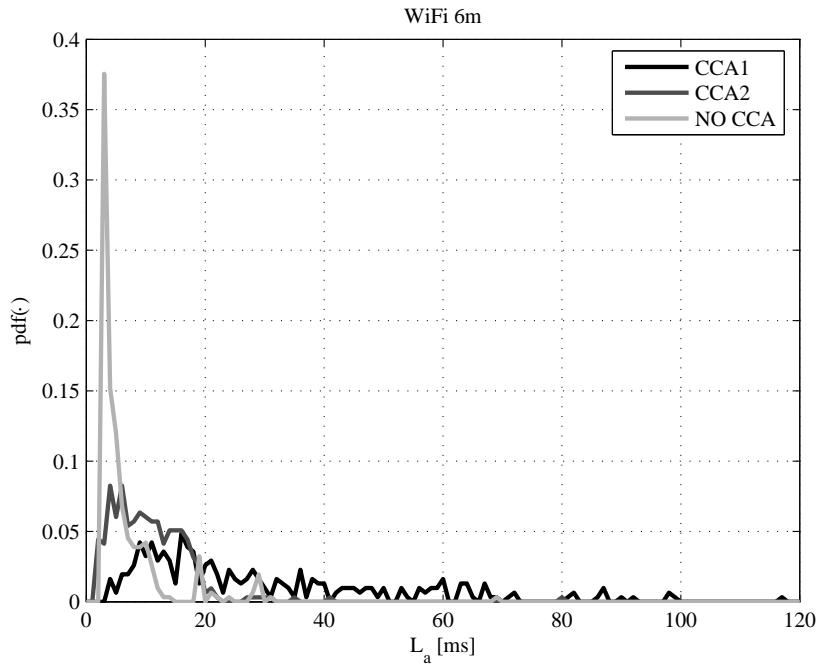
### 6.5.1 Alarm Latency

In this section the alarm latency PDF analysis is presented, highlighting the CCA mode influence on it. The reported plots are normalized histogram, typically made of 300 samples, *i.e.* the number of alarms generated, in mean, in a total of 10000 polling cycles of 30 ms. The statistical behavior suggested in the following figures, even if based on a small amount of data, remains the same even with a larger and more complete set of samples, as reported in [16]. In Figure 6.11 the alarm latency PDF in the absence of interference is shown. For CCA mode 1 and 2 the shape is asymmetrical, exponential like: latency is upper bounded to 40 ms. For NO CCA latency is smaller, more concentrated below 10 ms with a spike at 3 ms. As for polling cycle, and polling latency Bluetooth does not influence alarm latency, leading to PDFs similar to those just described.

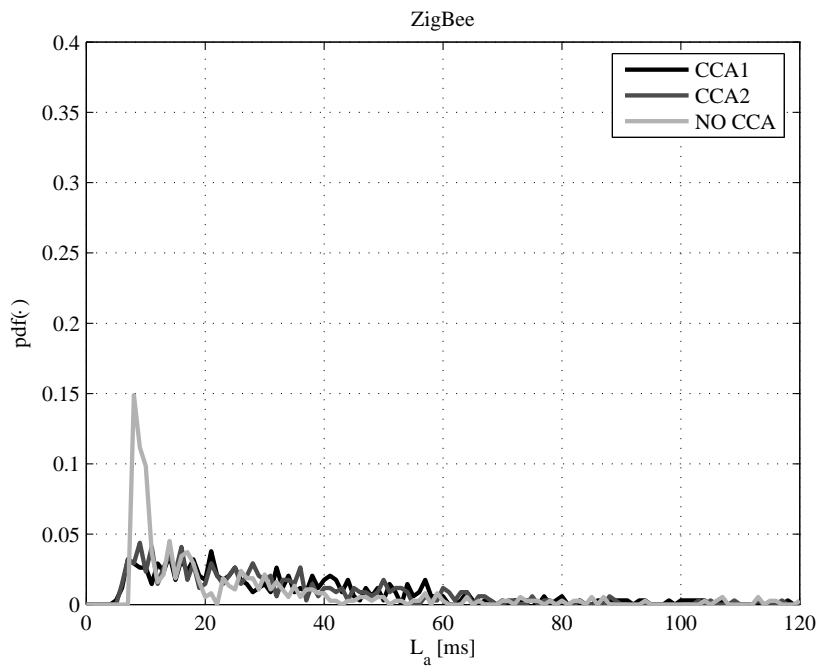


**Figure 6.11:** Probability density function for Alarm latency ( $L_a$ ) with no interference.

The effect of WiFi interference (Figure 6.12), as for polling cycle, is greatly mitigated by the use of CCA 2, or, even better, with NO CCA. If mode 1 is used, alarm latency is spread toward greater values, even above 100 ms. The effect of ZigBee interferer is deleterious also on alarm latency. Both with CCA 1 and 2 the latency is enlarged toward 120 ms, and also with a NO CCA mode, although more concentrated below 40 ms, the latency still reaches 120 ms.



**Figure 6.12:** Probability density function for Alarm latency ( $L_a$ ) with WiFi interference.



**Figure 6.13:** Probability density function for Alarm latency ( $L_a$ ) with ZigBee interference.

## 6.6 Conclusions

In this chapter an experimental answer to the problem of interference on WSN have been provided. The obtained results, although in a qualitative manner, have clearly underlined the effects of CCA and Backoff periods on parameters like packet error rate, polling cycle PDF and alarm latency PDF. In particular the best performance are obtained with NO CCA configuration, disabling both the channel sense and the back-offs.

Bluetooth interference, thanks to its frequency hopping nature, does not disturb the network, and in this case the choice of CCA mode is pointless. Only a little packet error rate and PDF distortion is visible.

WiFi has revealed itself as the most disturbing system. In this case the choice of the correct CCA mode is essential: CCA 1 led to poor performance, while using CCA 2 or, better, NO CCA, performance improvement is huge. A drop of packet loss and PDFs distortion is noticed in these latter cases.

The worst performance have been obtained with a ZigBee interferer. As expected, in this case CCA mode 1 and 2 are perfectly equivalent, and the use of NO CCA mode little improves the performance. The use of CSMA/CA, although optimal in a distributed environment, reveals its problems when used in a real-time system, where latencies must be very low and a worst case bound should be always known. The CSMA/CA is a "best effort" solution for MAC layer, not optimal for a monitoring system, and even less adequate in an industrial system inserted in a control chain. Moreover, the conservative nature of CSMA/CA under-appreciate the robustness of IEEE 802.15.4 modulation layer, preventing transmission (and hence increasing the PER) even when the SINR is good enough for a successful communication.



# Chapter 7

## *Cross layer CSMA/CA modeling*

**I**NTERFERENCE represent a key issue to be carefully considered in the design and set-up of a wireless network. It may cause several network problems, as packet loss, reduced throughput, delay, jitter, bad synchronization, missed alarms, etc. A critical frequency band, in which interference phenomena frequently occur. Such problems are even more important in such a crowded un-licensed band such as the 2.4 GHz ISM band (2.4 - 2.4835 GHz).

Analyzing interference issues is often a difficult task, both at design and installation stages. To this aim, helpful information can be obtained by monitoring some suitable parameters of the network through simulations and measurements. This task is however very complex in the presence of wireless networks employing the CSMA/CA protocol, like IEEE 802.11b/g WLANs and IEEE 802.15.4 WSNs. Interference may, in fact, act at two different levels: at PHY layer, in terms of data packet collisions, and at MAC layer, in terms of channel occupation. The two effects are typically very difficult to distinguish from the analysis of some measurable parameters, thus making it impractical also to choose a suitable mitigating solution. In the presence of performance degradations, network designers are often prone to associate the origin of the problem only with PHY layer interference. In these situations, the effects of MAC layer interference are often erroneously interpreted as related to PHY layer interference, and, consequently, non optimized or inefficient solutions are applied. As a matter of fact, MAC layer interference is seldom taken into account in the literature to evaluate the overall network performance [21, 24, 25]. Despite several models and studies concerning interference and coexistence issues among different networks are presented, attention is mainly paid to the PHY layer interference [15, 26], and only a few information is available for MAC layer interference [27].

The chapter presents a simple and original model for the separate analysis of PHY and MAC layer interference with respect to CSMA/CA-based wireless networks (for an in depth description of CSMA/CA see Chapter 2). The goal is to propose a simple tool for an efficient investigation of the origin (PHY or MAC) of an observed performance degradation and prediction of the level of PER in the presence of interference. Theoretical details underlying the model, special hints for its use, and results from many experiments aimed at validating the model, are given. In the experimental stage, a suitable testbed including a IEEE 802.15.4-compliant wireless sensor network is exploited.

The PHY and MAC joint analysis presented in this chapter is described in [28, 29].

## 7.1 Theoretical background

### 7.1.1 PHY layer interference

Interference at PHY layer has been receiving great attention both in the literature and practice. It is due to collisions among useful and interfering signals at the receiver, and it is commonly studied under the hypothesis of a channel affected by AWGN. Curves of expected BER and PER versus Signal to Noise Ratio (SNR) are commonly given in the most relevant communication standards [1, 2]. The curves depend on many factors, such as the adopted modulation scheme and error correction techniques. A qualitative example of a generic PER vs SINR curve is shown in Figure 7.1(a) (measurement units have been omitted to highlight the qualitative nature of the curves). PER is the sum of two contributions:  $PER_{phy}$  and  $PER_{mac}$ .  $PER_{phy}$  is the ratio between the number of erroneously received data packets (due to collisions),  $N_{er}$ , and the total number  $N_t$  of data packets delivered under ideal conditions (without interference).  $PER_{mac}$  is the ratio between the number of not delivered data packets (due to channel occupation),  $N_{nt}$ , and  $N_t$ . SINR is given by:

$$SINR = \frac{P_s}{P_{i_p} + P_n}, \quad (7.1)$$

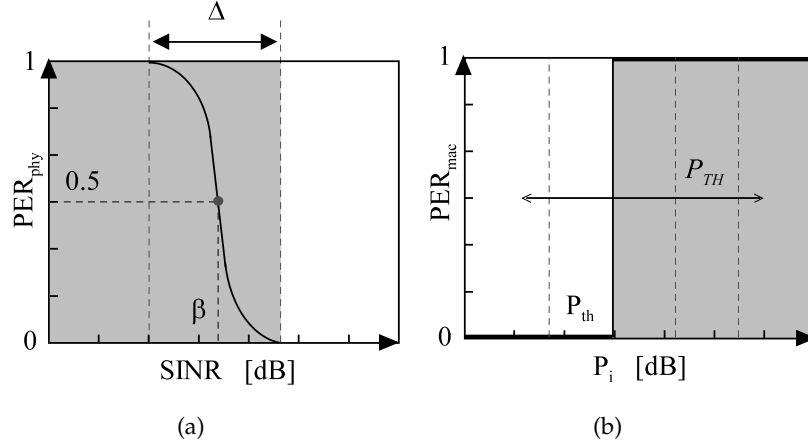
In (7.1),  $P_s$  and  $P_{i_p}$  are respectively the useful and interference in-channel power at the antenna input connector of the receiving node, while  $P_n$  is the thermal noise power level at the same connector:

$$P_n = k \cdot T \cdot B \cdot NF, \quad (7.2)$$



where  $k$  is the Boltzmann's constant,  $T$  the absolute temperature (Kelvin),  $B$  the bandwidth of the involved channel, and  $NF$  the noise figure of the receiver front-end. In decibel units (dBm) and at 20°C, it results:

$$P_n \text{ [dBm]} \approx -174 \text{ dBm/Hz} + 10 \log_{10}(B) + NF \text{ [dB]}.$$



**Figure 7.1:**  $PER$  versus  $SINR$  and  $P_i$ : (a) at physical layer, (b) at MAC layer.

Figure 7.1(a) highlights two regions: the first, white, in which  $PER_{phy} < PER^* \approx 0$  (full packet reception), where  $PER^*$  is the maximum acceptable value of  $PER$ , usually very close to 0; the second, darker, in which  $PER_{phy} \geq PER^*$ . In the latter region, the zero packet reception range, defined as the range in which  $PER_{phy} > 1 - PER^*$ , can be distinguished. A further transitional region of width  $\Delta$  is noticed, in which  $PER_{phy} \in [PER^*, 1 - PER^*]$  [30, 31, 32]. The curve of Figure 7.1(a), with the  $PER$  axis in linear units and the  $SINR$  axis in logarithmic ones, can be approximated to a simple invertible function by means of a non-linear least square fit [33], yielding:

$$PER_{phy} \left( P_s, P_i, P_n \right) = \frac{1}{1 + e^{\alpha(SINR - \beta)}}, \quad (7.3)$$

where the coefficient  $\alpha$  defines the slope of the curve in the transitional region, while  $\beta$  is the curve offset, *i.e.* the  $SINR$  level for which  $PER_{phy} = 0.5$ . From (7.3), the following expression of  $\Delta$  can be derived:

$$\Delta = \frac{2 \cdot \ln \left( PER^{*-1} - 1 \right)}{\alpha}. \quad (7.4)$$

Eq. (7.4) can be used to determine  $\alpha$ , for a desired level of  $PER^*$  and for a given  $\Delta$ , to be estimated from standardized curves like that of Figure 7.1(a).

In most cases of practical interest,  $PER_{phy}$  can be considered dependent only on  $P_s$  and  $P_{i_p}$ . The term  $P_n$ , in fact, depends only on the hardware characteristics and  $T$ .

It is worth noting that equation (7.3) can be very appropriate to describe the behavior of a wireless network in terms of  $PER$  vs  $SIR$ . This can be seen in Figure 7.2, where a theoretical graph of  $PER$  vs  $SIR$ , available from the IEEE 802.15.4 standard [2], is shown along with its approximation obtained by using (7.3) with  $PER^* = 10^{-3}$ . Logarithmic scale has been used to better highlight differences between the two curves. The graph clearly shows small differences between the two curves when  $PER > 10^{-3}$ , confirming the good accuracy of (7.3) when  $PER > PER^*$ , *i.e.* when the interference effect is greater.

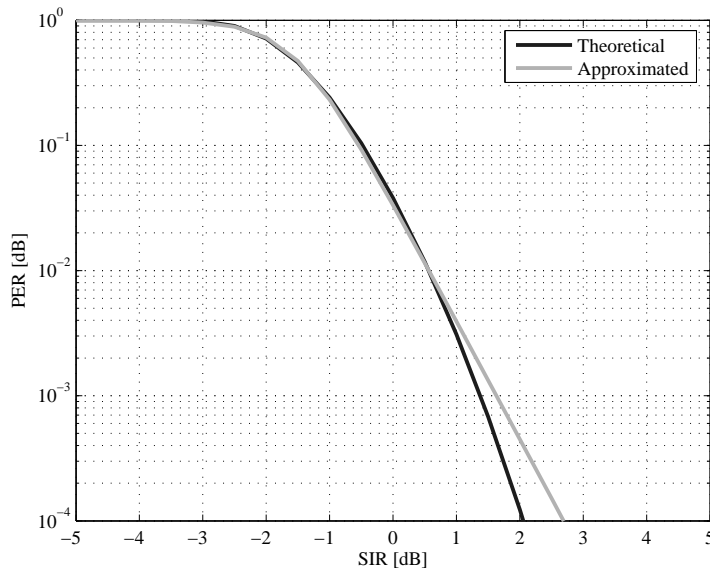


Figure 7.2: Theoretical and approximated PER for IEEE 802.15.4 systems.

### 7.1.2 MAC layer interference

Interference at MAC layer, although little investigated in the literature and considered in practice, is a major cause of performance degradation in a CSMA/CA-based wireless network. Such a phenomenon is also dependent on the selected CCA mode, and is typically much more severe in the CCA mode 1 (energy detection mode). This is the reason why, in the following, the attention will be paid only to CCA mode 1, which causes the delivery of packets to be stopped and deferred any time the interference power,  $P_{i_m}$ , at the node wishing to transmit, is greater than the selected CCA threshold,  $P_{th}$ .

Consequently, no packet at the receiver side is received, causing an abrupt degradation of  $PER_{mac}$ , as depicted in Figure 7.1(b). Also in this case, two regions are highlighted: one, darker, in which  $PER_{mac} = 1$  ( $P_{i_m} \geq P_{th}$ ); the other, white, in which  $PER_{mac} = 0$  ( $P_{i_m} < P_{th}$ ). The curve that separates the two regions of Figure 7.1(b) can thus be modeled as:

$$PER_{mac}(P_{i_m}, P_{th}) = H(P_{i_m} - P_{th}), \quad (7.5)$$

where  $H(\cdot)$  is the Heaviside unit step function [34]. As shown in (7.5),  $PER_{mac}$  depends only on the interference power level at the transmitting node, rather than on the levels assumed by  $P_s$  or  $P_{i_p}$  at the destination receiving node.

### 7.1.3 Multi-link network

The above equations (7.3) and (7.5) of  $PER_{phy}$  and  $PER_{mac}$  refer to the specific case of a single communication link, between one transmitting and one receiving device. The results they provide are essential in order to evaluate the performance of a single specific node and, if suitably combined, they can also be used to estimate the overall  $PER$  in a multi-link network. To this aim, the network can be modeled as an oriented graph, in which the nodes represent the considered wireless devices and the links are the radio communications between them.

A probabilistic approach is presented hereinafter to evaluate the overall  $PER$  of a multi-link wireless network. Specifically, a general purpose wireless network is considered having  $K$  operative communication links,  $l_k$  with  $\{k = 1, \dots, K\}$ , each of which characterized by a packet loss probability  $\pi_k$ . From the Bayes theory [35], the overall network packet loss probability,  $\bar{\pi}$ , can be expressed as:

$$\bar{\pi} = \sum_{k=1}^K \pi_k \cdot \eta_k, \quad (7.6)$$

where  $\eta_k$  is the probability that  $l_k$  is on. In (7.6),  $\pi_k$  and  $\bar{\pi}$  can be considered as the probabilistic estimate respectively of the  $PER$  associated to the single link  $l_k$  and of the overall  $PER$ .  $\bar{\pi}$  can also be viewed as a weighted mean of the  $PER$  values associated to the  $K$  single links. The weighting coefficients  $\eta_k$  are instead dependent on the corresponding link radio characteristics and on how long they are operative. For instance, in a fully symmetric topology network, *e.g.* an ad-hoc or star network having all links operating at the same rate:  $\eta_k = \frac{1}{K}$  for any  $\{k = 1, \dots, K\}$ .

In this case, as well as when there is no a-priori knowledge about how long each single link is operative, equation (7.6) becomes a simple arithmetic mean of  $\pi_k$ . For the sake of conciseness only the generic case of a fully symmetric topology is considered; therefore, the overall  $PER$  will be computed by averaging the  $PER$  estimates of the various links.

## 7.2 Proposed Model

A model is proposed for the evaluation of the overall  $PER$ ,  $PER_k$ , associated to each single link,  $l_k$ , with  $\{k = 1, \dots, K\}$ , in a CSMA/CA-based wireless network. The model can be extended to the case of a multi-link network according to the notes given in Section 7.1.3.

In the generic link  $l_k$ , the transmitting and receiving terminals are assumed to be characterized by the same interference power level  $P_i$ :  $P_i = P_{i_p} = P_{i_m}$ . Under this assumption, the overall  $PER$  associated to the link  $l_k$  can be modeled as:

$$PER_k(P_s, P_i, P_{th}) = \max(PER_{phy}, PER_{mac}), \quad (7.7)$$

which combines both equations (7.3) and (7.5), and where  $P_{i_p}$  and  $P_{i_m}$  indicated in (7.1), (7.3) and (7.5) have been replaced by  $P_i$ . A problem in the use of (7.7) is its three-dimensional (3D) nature,  $\mathbb{R}^3 \rightarrow [0, 1]$ , which makes more difficult its graphical representation and practical use for the assessment of  $PER_k$ . To avoid this problem, an efficient and simplified approach is introduced, which considers the iso-surface  $\Sigma^*$  defined as:

$$\Sigma^* = \{\mathbf{P} \in \mathbb{R}^3 \mid PER_k(\mathbf{P}) = PER^*\}, \quad (7.8)$$

where  $\mathbf{P}$  is a combination of the parameters  $(P_s, P_i, P_{th})$ , and  $PER^*$  is the maximum value of  $PER$  that can be accepted. Such an iso-surface allows the 3D domain of  $(P_s, P_i, P_{th})$  to be subdivided into two regions: one where the network operates at very high performance levels, *i.e.*  $PER < PER^*$ , another in which the detrimental effect of the interference is unacceptable, *i.e.*  $PER > PER^*$ .

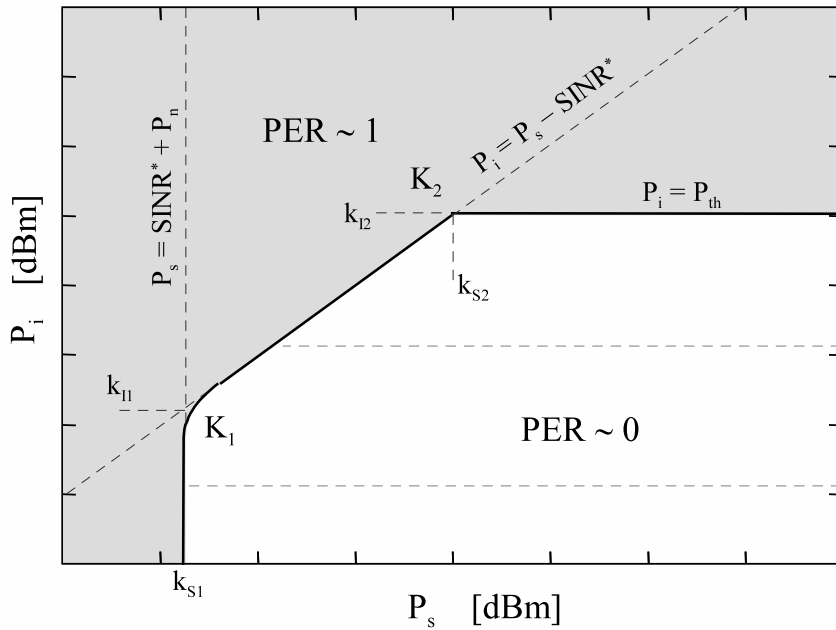
The two regions can efficiently be analyzed by fixing one of the three parameters  $(P_s, P_i, P_{th})$  and plotting the remaining ones. This way, three different charts can be obtained, representing the projection on the plane  $(P_s, P_i)$ ,  $(P_i, P_{th})$ , or  $(P_s, P_{th})$  of the surface  $\Sigma^*$ , for a chosen value of  $P_{th}$ ,  $P_s$  and  $P_i$ , respectively.

### 7.2.1 SI-chart: fixed threshold

A simplified sketch of the first iso-surface projection, called SI-chart, is visible in Figure 7.3:  $P_{th}$  is the fixed parameter. In the chart, the solid curve represents the values of  $(P_s, P_i)$  belonging to the iso-surface  $\Sigma^*$  for a given value of  $P_{th}$ . The curve highlights two regions: one, darker, in which  $PER_k \geq PER^* \approx 0$  (zero packet reception), another, white, in which  $PER_k < PER^*$  (full packet reception). The measurement units in the axes are purposely omitted in order to highlight the qualitative nature of the plot.

One should note that Figure 7.3 is very easy to plot. It can be obtained by drawing the following three straight lines:

1.  $P_i = P_s - SINR^*$  where  $SINR^*$  is the minimum  $SINR$  beyond which  $PER_k < PER^*$ .  $SINR^*$  can be obtained inverting (7.3) and assuming  $PER_{phy} = PER^*$ :  $SINR^* = \beta + \Delta/2$ .
2.  $P_s = SINR^* + P_n$ .
3.  $P_i = P_{th}$ .



**Figure 7.3:** SI chart: modeling  $PER_k$  in a CSMA/CA-based wireless network.

The intersection of the above three lines identifies two knee points:

$K_1 = (k_{S1}, k_{I1})$ , with  $k_{S1} = SINR^* + P_n$  and  $k_{I1} = P_n$ , and  $K_2 = (k_{S2}, k_{I2})$ , with  $k_{S2} = P_{th} + SINR^*$  and  $k_{I2} = P_{th}$ . The horizontal dashed lines instead represent line 3 for different levels of  $P_{th}$ .

The use of this graph in a design context allows to efficiently: 1) recognize if a given setup is suitable to achieve the expected performance level or not, 2) suggest how to improve the setup itself, 3) avoid erroneous setup.

For example, once chosen  $PER^*$  (e.g.  $PER^* = 0.001$ ) and determined  $SINR^*$ ,  $\alpha$  and  $\beta$ , it is straightforward to plot the lines and localize the point in the graph for a given couple  $(P_s, P_i)$ , and then to deduce if the overall  $PER_k$  (PHY and MAC) is good or not. The graph also allows to easily determine the maximum allowed level of external interference  $P_i^*$  to achieve  $PER_k < PER^*$ . It is sufficient to draw a vertical line at the expected value of  $P_s$  and determine the  $P_i$ -axis coordinate,  $P_i^*$ , of the intersection point between this latter line and the solid curve. This value of  $P_i^*$  represents the maximum level of interference beyond which  $PER_k$  starts abruptly increasing. It can also be considered as the overall immunity level to interference of the network under test. Accordingly, the difference  $\Delta_i = P_i^* - P_i$  can be considered as immunity margin to interference. Therefore, the greater  $\Delta_i$ , the better the immunity to interference.

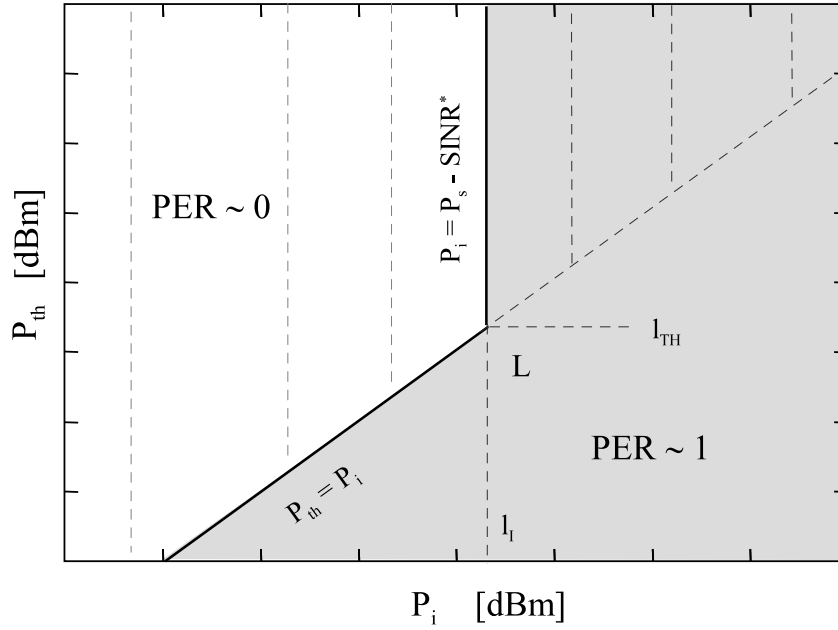
The figure shows that  $P_i^*$  grows proportionally with  $P_s$  in the range  $k_{S_1} < P_s < k_{S_2}$ , while it is constant and equal to  $P_{th}$  when  $P_s > k_{S_2}$ . Such different behavior depends on the interference phenomenon involved (PHY or MAC). Specifically, in the first range  $k_{S_1} < P_s < k_{S_2}$ , interference acts at PHY layer, so if  $P_s$  grows, a greater interference  $P_i$  is needed to reduce  $SINR$  and increase  $PER_k$ . In the second range,  $P_s > k_{S_2}$ , interference acts at MAC level. In this range, the maximum allowed  $P_i^*$  saturates to the chosen  $P_{th}$ . Therefore, in the case of  $P_i > P_i^*$ , the channel is always sensed busy and  $PER_k \approx 1$  regardless of the value chosen for  $P_s$ . The figure also underlines the importance of  $P_{th}$  (the parameter  $d$  in the figure), which considerably changes the extension of the zero packet reception (white) area. The vertical part of the curve in Figure 7.3,  $0 < P_i < k_{I_1}$ , represents the case in which both the external interference ( $P_i$ ) and useful signal ( $P_s$ ) are weak and the thermal noise significantly contributes to increase  $PER_k$ .

### 7.2.2 ITH-chart: fixed signal power

The second projection of the iso-surface  $\Sigma^*$ , called ITH-chart, is shown in Figure 7.4:  $P_s$  is the fixed input parameter.

Only two straight lines need to be drawn:

1.  $P_{th} = P_i$ .
2.  $P_i = P_s - SINR^*$ .



**Figure 7.4:** ITH chart: modeling  $PER_k$  in a CSMA/CA-based wireless network.

The intersection of the lines identifies one knee point:  $L = (l_I, l_{TH})$ , with  $l_I = l_{TH} = P_s - SINR^*$ . The vertical dashed lines represent line 2 for different levels of  $P_s$ . Also in this case, the use of this graph in a design context can be very advantageous. With respect to the SI-chart, it allows more easily determining the range of  $P_{th}$  values that make the wireless link be more immune to interference, *i.e.*  $PER_k < PER^*$ . It is sufficient to draw a vertical line at the expected value of  $P_i$ , and determine the  $P_{th}$ -axis coordinate,  $P_{th}^*$ , related to the intersection between it and the solid curve. This value of  $P_{th}^*$  represents the minimum threshold level below which  $PER_k$  is 1. One should note that such an intersection exists only if  $P_i < l_I$ , while it does not appear for  $P_i > l_I$ . In the former case, the difference  $\Delta_{th} = P_{th} - P_{th}^*$  can be considered as an immunity margin to interference. Therefore, the higher  $\Delta_{th}$ , the better the immunity to interference. In the latter range of  $P_i$  values, a completely degraded  $PER_k$  is always obtained regardless of the chosen  $P_{th}$ . This allows highlighting the two conditions needed to verify the performance of the link in the presence of interference, in terms of the coordinates  $P_i$  and  $P_{th}$ , and for a given  $P_s$ : (i)  $P_i < l_I$ , (ii)  $P_{th} > P_{th}^* = l_I$ , with  $l_I = P_s - SINR^*$ . From Figure 7.4, it can also be observed that increasing  $P_{th}$  much beyond  $P_{th}^*$  is: (i) advantageous in terms of the immunity margin  $\Delta_{th}$ , (ii) useless in terms of  $PER_k$  (it is always null for  $P_{th} > P_{th}^*$ ), (iii) negative in terms of the CSMA/CA mechanism. In fact, upon the increasing of  $P_{th}$ , the capacity of the mechanism in avoiding collisions decreases.

In the opposite case, in which  $P_{th}$  is fixed, the graph also gives the opportunity of easily determining the maximum allowed level of external interference  $P_i^*$  to achieve  $PER < PER^*$ . In this case, it is sufficient to draw a horizontal line at the chosen value of  $P_{th}$ , and determine the  $P_i$ -axis coordinate,  $P_i^*$ , of the intersection point between it and the solid curve. From the plot it results:  $P_i^* = P_{th}$  if  $P_{th} < l_{TH}$ , and  $P_i^* = l_I = P_s - SINR^*$ , if  $P_{th} > l_{TH}$ .

Figure 7.4 finally allows easily determining the type of interference occurring, PHY or MAC. Specifically, in the range  $P_{th} < l_{TH}$  the main interference effect is at MAC layer:  $P_i^*$  grows proportionality with  $P_{th}$ . On the other side, in the range  $P_{th} > l_{TH}$  the main interference effect has to be considered at PHY layer:  $P_i^*$  is constant regardless of the value chosen for  $P_{th}$ .

### 7.2.3 STH-chart: fixed interference power

The third projection of the iso-surface  $\Sigma^*$ , called STH-chart, is shown in Figure 7.5:  $P_i$  is the fixed input parameter. Also in this case, the plot can easily be sketched. Only two straight lines need to be drawn:

1.  $P_{th} = P_i$ ,
2.  $P_s = P_i + SINR^*$ .

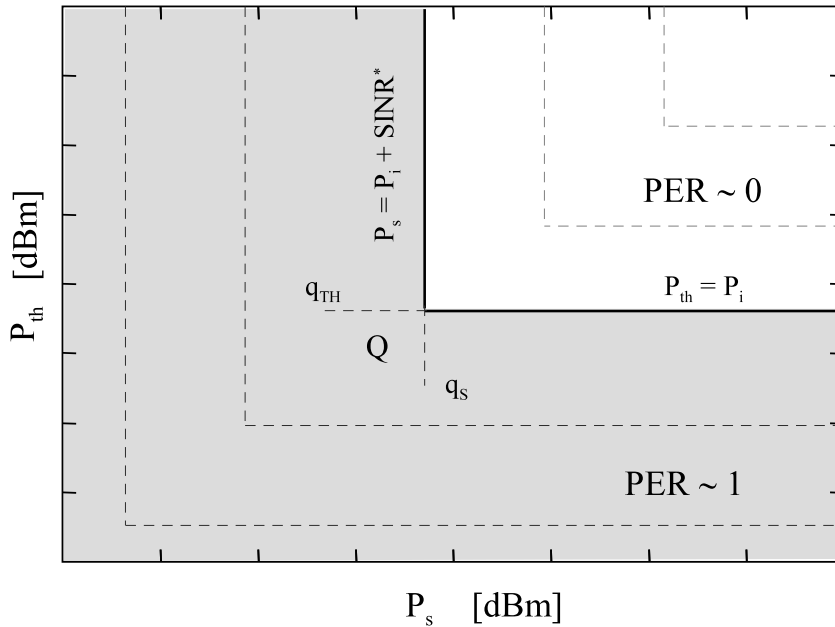


Figure 7.5: STH chart: modeling  $PER_k$  in a CSMA/CA-based wireless network.



The intersection of the two lines identifies the knee point  $Q$ , having coordinates  $(q_S, q_{TH})$ , with  $q_S = P_i + SINR^*$  and  $q_{TH} = P_i$ . The horizontal and vertical dashed lines represent lines 1 and 2 for different levels of  $P_i$ .

Also in this case, the use of this graph in a design context can be very advantageous. It can be very helpful to easily determine the minimum value both of  $P_s$  and  $P_{th}$  needed to obtain  $PER_k = 0$ . In particular, in the range  $P_s > q_S$ , the minimum value of  $P_{th}$ ,  $P_{th}^*$ , needed to obtain  $PER_k = 0$  is simply the intersection of the solid line with an horizontal line at the expected value  $P_{th}$ . In the range  $P_s < q_S$ , instead, the graph highlights a unitary value of  $PER_k$  for any  $P_{th}$  level. Similarly, in the range  $P_{th} > q_{TH}$ , the minimum value of  $P_s$ ,  $P_s^*$ , needed to obtain  $PER_k = 0$ , is simply the intersection of the solid line with a vertical line at the expected value of  $P_s$ . In the range  $P_{th} < q_{TH}$ , instead,  $PER_k$  is always unitary for any  $P_s$  level.

Differently from first two charts, the type of the occurring interference, PHY or MAC, cannot be established. In fact, the values assumed by  $P_i^*$ , *i.e.* the maximum level of  $P_i$ , are not clearly highlighted as in Figure 7.3 or in Figure 7.4.

## 7.3 Numerical results

To show the effectiveness of the proposed model, a significant case-study has been considered. It has involved a WSN with  $N = 11$  IEEE 802.15.4 nodes and an interfering external source: 5 MHz bandwidth AWGN signal operating in the same channel used by the WSN and providing the same interference power level  $P_i$  at each sensor node. A symmetric star topology has been considered, with one device operating as master and nine nodes operating as slaves. The master operates according to a round robin protocol, performing a periodical polling of nine enabled slaves, *i.e.*  $K = 10$ ,  $\eta_k = \frac{1}{10}$  for any  $\{k = 1, \dots, 10\}$ .

### 7.3.1 Two-dimensional analysis

A first set of results obtained by applying the proposed model is reported in Figure 7.6. The obtained SI-chart highlights a close agreement between these simulation curves and those of Figure 7.3. For a given level of  $P_s$ , with  $P_s = -25, -50, \text{ or } -75$  dBm, the intersections of the corresponding vertical straight lines and the dashed curves of the diagram are also highlighted (square, circle and triangle symbols). The  $P_i$ -coordinate of such points represents the maximum  $P_i$ ,  $P_i^*$ , beyond which  $PER$  abruptly increases to 1.

In the case of  $P_s = -50$  dBm, the plot shows that  $P_i^*$  grows almost proportionally with  $P_{th}$  in the interval  $-100$  dBm  $\leq P_{th} \leq -60$  dBm, while for  $P_{th} > -60$  dBm, it saturates nearly to  $-58$  dBm. This proves that the interference acts mainly at MAC layer in the interval  $-100$  dBm  $\leq P_{th} \leq -60$  dBm (the intersection is with an horizontal line), and at PHY layer in the range  $P_{th} > -60$  dBm (the intersection is with the oblique part of the curves). Similar considerations can be drawn for the other analyzed values of  $P_s$ . In particular, with  $P_s = -25$  and  $-75$  dBm,  $P_i^*$  respectively saturates to  $-32$  and  $-85$  dBm. It is worth noticing that such power levels  $P_s$  are typical in a WSN having the master set with a  $0$  dBm transmitting power, and the receiver devices placed at a distance from the master of nearly  $0.5$  m,  $1$  m, and greater than  $10$  m, respectively.

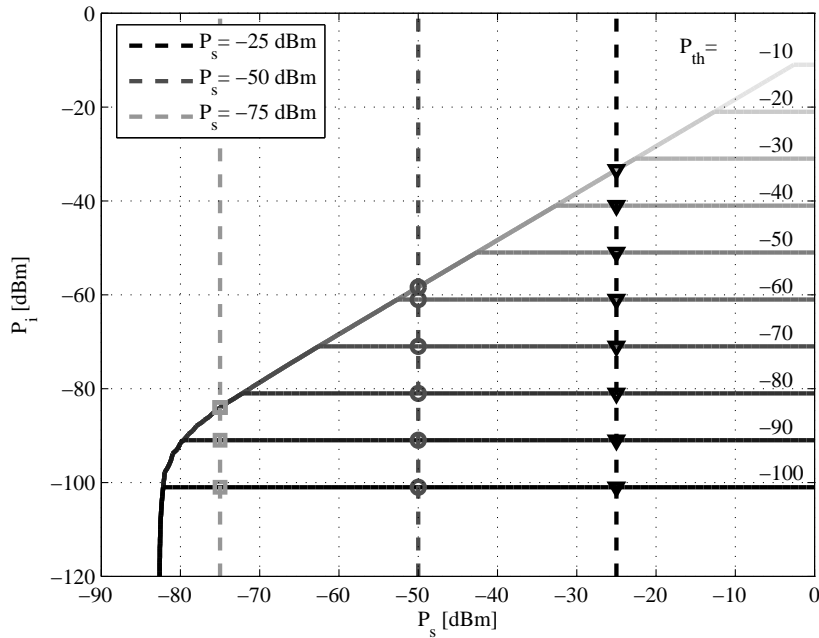


Figure 7.6: SI chart.

The obtained ITH-chart is reported in Figure 7.7, along with the intersection points  $(P_s, P_i, P_{th})$  indicated in Figure 7.6 with square, circle and triangle symbols. Also in this case, a close agreement can be noted between the ITH-chart and the curves of Figure 7.4. The three power levels  $P_s = -25$ ,  $-50$ , or  $-75$  dBm, represented in Figure 7.6 as vertical lines, appear as dashed curves. The  $P_i$ -coordinate of each point of these curves indicates the maximum interference level,  $P_i^*$ , that the network can tolerate for a given  $P_{th}$  and  $P_s$ .

For example, in the specific case of  $P_s = -50$  dBm, the chart shows that  $P_i^*$  grows proportionally with  $P_{th}$  within the interval  $-100$  dBm  $\leq P_i \leq -60$  dBm (MAC layer interference), and saturates nearly to  $-58$  dBm for  $P_i > -60$  dBm (PHY layer interference). In this latter range, any variation of  $P_{th}$  does not produce significant changes of  $P_i^*$ . Similar considerations hold for  $P_s = -25$  and  $-75$  dBm, in correspondence of which  $P_i^*$  saturates to  $-32$  and  $-85$  dBm respectively.

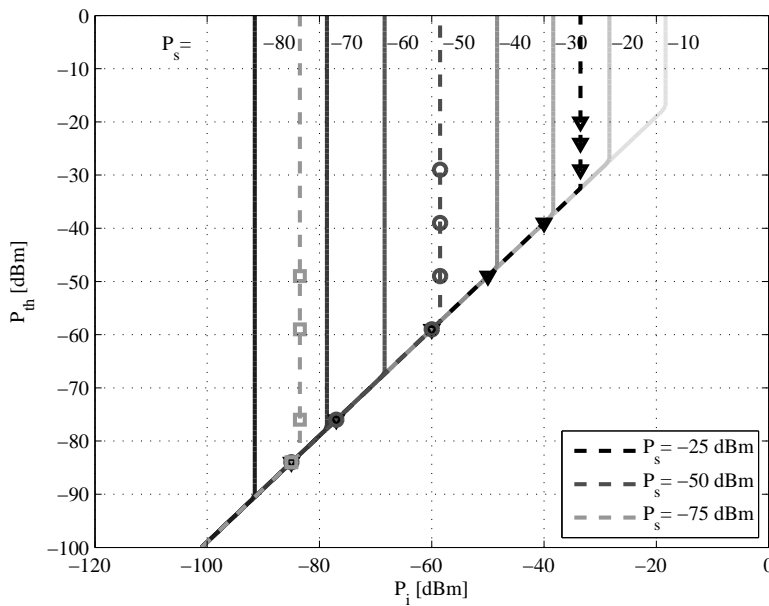


Figure 7.7: ITH chart.

The obtained STH-chart is finally shown in Figure 7.8 along with the intersection points  $(P_s, P_i, P_{th})$  indicated in Figure 7.6 with square, circle and triangle symbols. Once again, the obtained values are very similar to the qualitative lines of Figure 7.5. The three dashed lines associated to the quantities  $P_s = -25$ ,  $-50$ , or  $-75$  dBm are also highlighted. For a given level of  $P_s$ , the  $P_{th}$ -coordinate of the intersection point between the vertical dashed line and an horizontal solid line provides the corresponding minimum value of  $P_{th}$ ,  $P_{th}^*$ , needed to obtain  $PER_k = 0$ . The chart shows that  $P_{th}^*$  grows proportionally with  $P_i$  until the two lines do not intersect anymore. The minimum value of  $P_i$  in correspondence of which this condition is verified is just the immunity level of the WSN,  $P_i^*$ . As it can be noted and previously observed, evaluating  $P_i^*$  is not so simple as in the case of SI and ITH-charts. For instance, with  $P_s = -50$  dBm, the dashed line intersects a solid one only in the interval  $-100$  dBm  $\leq P_i \leq -58$  dBm.

Beyond this interval, *i.e.*  $P_i > -58$  dBm, no further intersections occur, and thus:  $P_i^* = -58$  dBm. Similar considerations can be drawn for the other chosen values of  $P_s$ . In particular, with  $P_s = -25$  and  $-75$  dBm,  $P_i^*$  respectively saturates to  $-32$  and  $-85$  dBm.

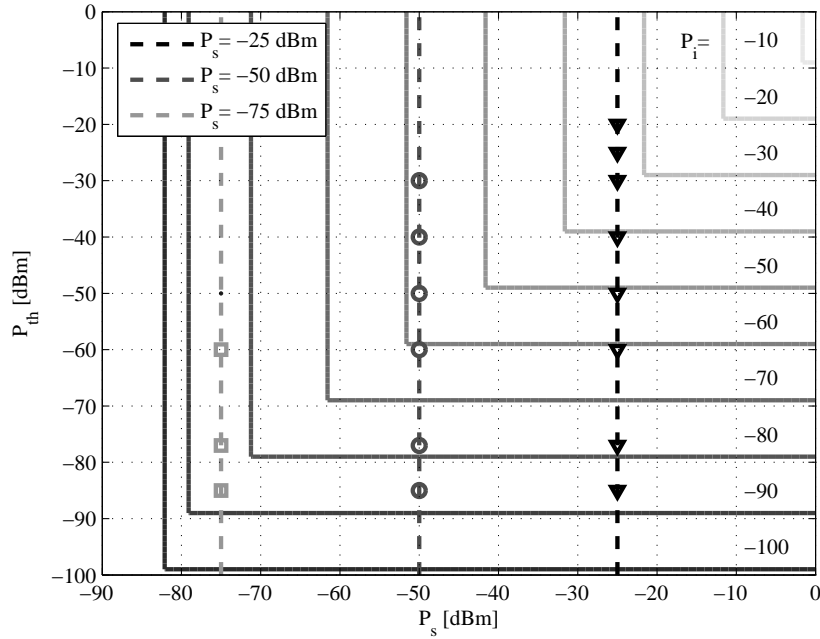


Figure 7.8: STH chart.

### 7.3.2 Three-dimensional analysis

A 3D representation of the results obtained by applying the model is given in Figure 7.9. The obtained iso-surface  $\Sigma^*$  consists of three planes,  $\sigma_1$ ,  $\sigma_2$ , and  $\sigma_3$ , defined, according to Section 7.2, as:

1.  $\sigma_1 : P_i = P_s - SINR^*$ ,
2.  $\sigma_2 : P_s = SINR^* + P_n$ ,
3.  $\sigma_3 : P_i = P_{th}$ .

As it can be noted, the 3D representation is not so easy to use if practical information about some key network parameters like  $P_s^*$ ,  $P_{th}^*$  and  $P_i^*$  is desired. The proposed  $\Sigma^*$  projection charts can prove very successful.

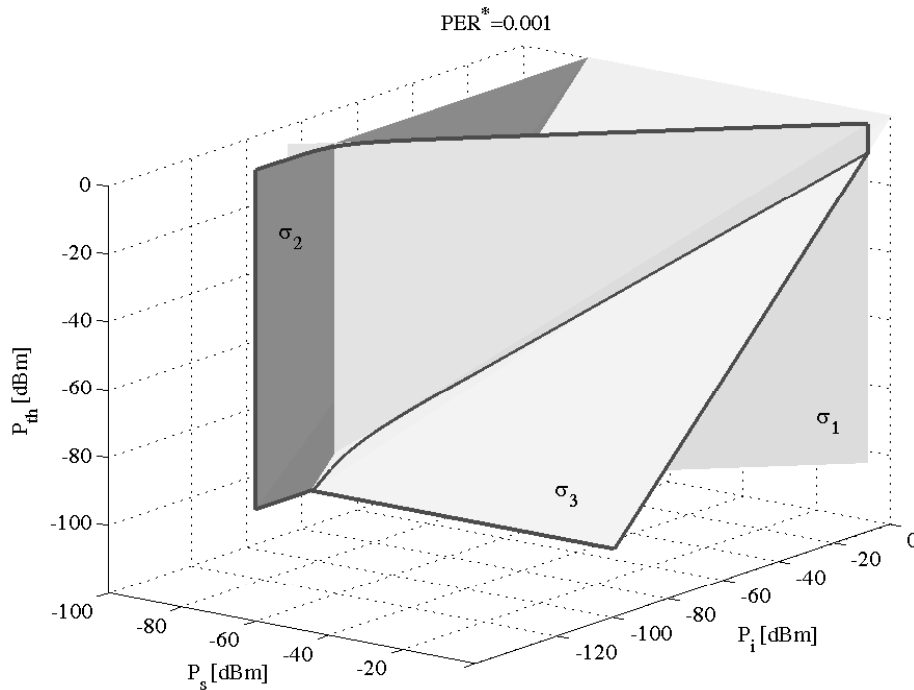


Figure 7.9: Simulation results, 3D representation.

## 7.4 Experimental Validation

To assess the efficacy and reliability of the proposed model, some laboratory tests have been carried out.

### 7.4.1 Testbed

The adopted testbed, depicted in Figure 7.10, consists of a simple WSN, made of  $N = 11$  wireless sensor nodes (motives), one acting as master,  $M$ , and the other as slaves,  $s_k$ , with  $\{k = 1, \dots, K\}$  and  $K = 10$ . Wireless nodes are Tmote Sky sensors compliant with the IEEE 802.15.4 standard, available from Moteiv [5]. The mounted radio chip, CC2420 from Chipcon [7], allows the choice of CCA mode and threshold. A PC is connected to  $M$  via a wired link, for storing collected data and post-processing. In particular, traffic is generated and monitored through the tool Factory Sniffer (see chapter 4), which allows the setup of master and slave and displays the data acquired from sensors. The slaves are uniformly positioned along a circumference of radius  $r$  and with  $M$  in the center. Master executes a periodical polling of the slaves (polling request), which reply sending the acquired data. An ACK frame is used to confirm correct reception; in case of not received ACK, polling is assumed failed and no re-transmission is performed.

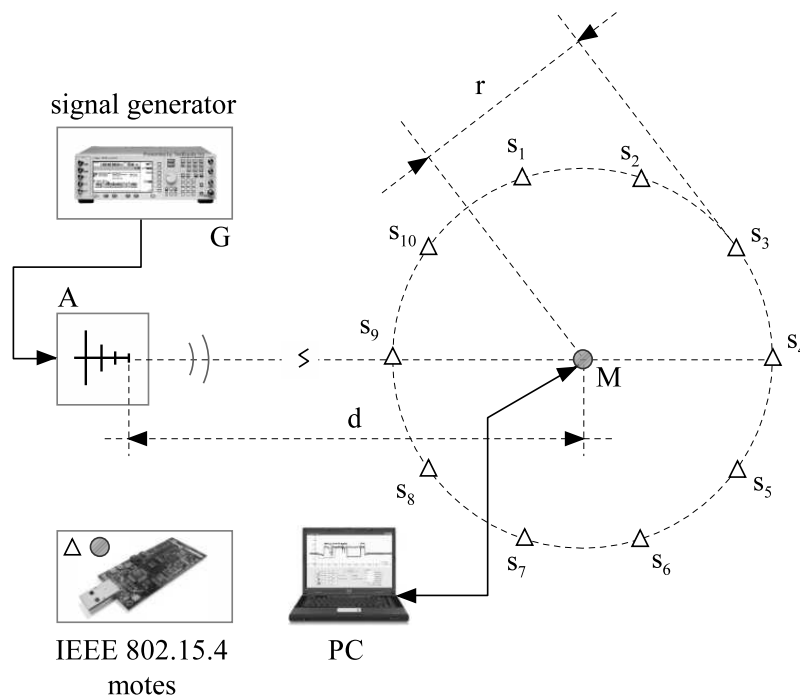


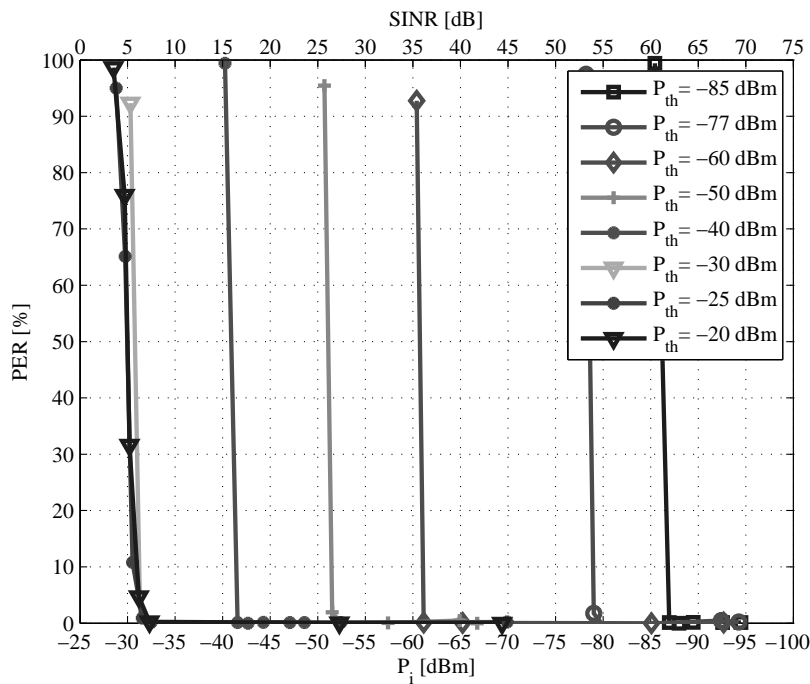
Figure 7.10: Adopted testbed.

The evaluation of  $PER_k$  for each salve is based on the number of correctly received pollings compared to the total number of pollings attempted. The overall  $PER$  is computed according to (7.6). Timestamps and RSSI readings are also added to the received data packets in order to monitor some key transmission performance parameters. For all the performed tests, polling period has been set equal to 30 ms, which is the minimum value allowing lossless transmission in the absence of interference [17]. The channel number 26 centered at 2480 MHz has been also chosen in order to minimize frequency overlap with bands of near-operating WLANs. In order to emulate in-channel interference, an arbitrary RF signal generator, G, Agilent Technologies AT E4433B (250 kHz–4 GHz) and a log-periodic antenna, A, EMCO 3146 (220 MHz–1 GHz), oriented toward M, have been used. Despite the limited range of frequencies of A, a good level of interference power has been obtained at the WSN position also at 2.480 GHz. The generated interference was an in-channel AWGN with 5 MHz bandwidth and centered at 2480 MHz. This choice makes noise assuming a spectral content close to which of some typical interfering contributions of ISM band, e.g. WLAN signals [1]. The antenna has been positioned at a sufficiently large distance  $d$  from M in order to obtain the same desired levels of power  $P_i$  at each mote. Accordingly, the radius  $r$  has been suitably arranged in order to obtain the same desired levels of power  $P_s$  at each mote.

Power measurements of  $P_s$  and  $P_i$  have been executed exploiting the in-band power detector of motes, which provides RSSI readings. The values so obtained provide estimates of the power effectively measured at the antenna output connector of motes. The testbed has been physically positioned as shown in the figure, above wood tables at one meter from ground floor. Tests have been conducted in open-site and real-life conditions, and in the absence of external interfering sources. In fact, in the absence of intentional interference, *i.e.* disabling  $G$ , a null  $PER$  has been assessed.

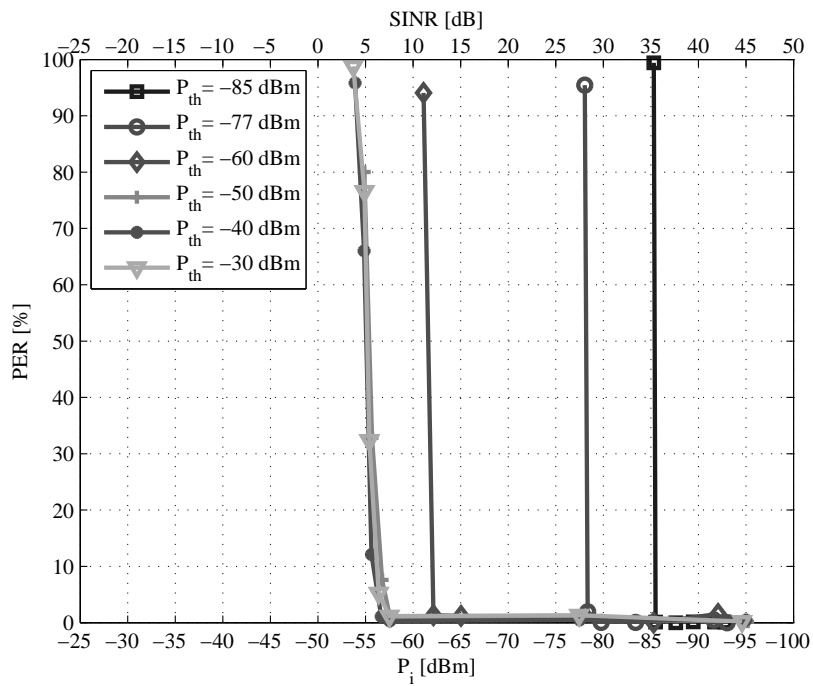
### 7.4.2 Measurement Results

A first set of results from experiments in the case of  $P_s = -25$  dBm is shown in Figure 7.11. This value of  $P_s$  has been obtained with a motes transmitting power equal to 0 dBm and  $r = 0.5$  m. By properly varying  $d$ , the interference level at the motes' antenna has been changed in the interval  $-100$  dBm  $\leq P_i \leq -25$  dBm, to assure a  $SINR$  greater than 0 dB.

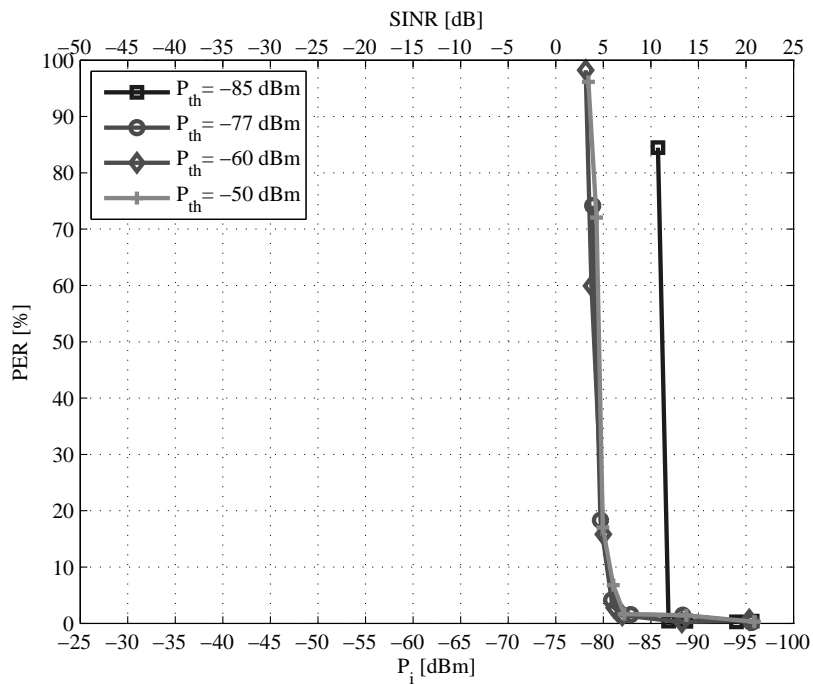


**Figure 7.11:** Experimental results:  $PER$  versus  $P_i$  with  $P_s = -25$  dBm and different CCA thresholds,  $P_{TH}$ .

From the diagrams, the maximum allowed levels of external interference  $P_i^*$  that the network can tolerate can easily be estimated:  $P_i^* = -86, -79, -61, -51, -42, -32, -32, -32$  dBm for  $P_{th} = -85, -77, -60, -50, -40, -30, -25, -20$  dBm respectively.



**Figure 7.12:** Experimental results:  $PER$  versus  $P_i$  with  $P_s = -50$  dBm and different CCA thresholds,  $P_{TH}$ .



**Figure 7.13:** Experimental results:  $PER$  versus  $P_i$  with  $P_s = -75$  dBm and different CCA thresholds,  $P_{TH}$ .



It can be observed that for values of  $P_{th}$  in the range -85 through -50 dBm,  $P_{th}$  and  $P_i^*$  assume almost the same values and are nearly proportional with each other. For instance, when  $P_{th}$  varies from -77 to -40 dBm (increase of 37 dB),  $P_i^*$  increases from -79 to -42 dBm, hence of 37 dB. For  $P_{th} > -30$  dBm, the curves overlap with each other and  $P_i^*$  saturates to a value nearly equal to -32 dBm. This clearly confirms the results obtained in Section 7.3 by applying the proposed model.

Another example is shown in Figure 7.12, where  $r = 1$  m and the transmit power is consequently  $P_s = -50$  dBm. The saturation effect of MAC layer threshold is still visible: in this case  $P_i^* = -87, -79, -62, -58, -57, -57$  dBm for  $P_{th} = -85, -77, -60, -50, -40, -30$  dBm respectively. It can be observed that the curves start to collapse, due to the low  $SINR$ , for lower values of interference. For values of  $P_{th}$  in the range -85 through -50 dBm,  $P_{th}$  and  $P_i^*$  assume quite the same values and vary quite proportionally with each other. For instance, when  $P_{th}$  varies from -77 to -60 dBm (increase of 17 dB),  $P_i^*$  increases from -79 to -62 dBm, hence of 17 dB. For  $P_{th} > -50$  dBm, the curves overlap with each other and  $P_i^*$  saturates to a value nearly equal to -57 dBm. Also this result confirms the outcomes obtained in Section 7.3 by applying the proposed model.

In Figure 7.13 a last experimental proof of the proposed model is given. In this case the effect of CCA threshold is almost dominated by the PHY layer effect. For typical values of  $P_{th}$  ( $P_{th} \geq -77$  dBm) all the  $PER$  curves appear superimposed regardless of the chosen  $P_{th}$ . In this case the curves collapse to a maximum value of  $P_i^*$  nearly equal to -83 dBm, confirming once gain the results obtained in Section 7.3.

## 7.5 Conclusions

The results obtained from the conducted experiments confirm that in-channel interference can affect the operation of a CSMA/CA-based wireless network in two different manners: (i) at PHY layer, in terms of signal collisions at the receiver side, (ii) at MAC layer, in terms of channel occupation at the transmitter side. Such effects may occur simultaneously and cause detrimental effects on the network performance; they can not easily be evaluated separately and solved selectively, both theoretically, through simulations, and experimentally. The good agreement noticed between the results obtained from the application of the proposed model and those achieved from real-life experiments confirm the accuracy of the model and its suitability to be used as an efficient design tool for: (i) forecasting the type of interference mainly occurring (PHY or MAC), (ii) forecasting the final performance in terms of *PER*, (iii) evaluating some key parameters of the network such as its immunity level,  $P_i^*$ , immunity margins  $\Delta_i$  and  $\Delta_{th}$ , minimum CCA threshold to be set at each transmitter device,  $P_{th}^*$ , minimum useful power needed at each receiver device to obtain good performance,  $P_s^*$ . Details and guidelines on how to apply the model, measure key parameters and properly interpret the presented two-dimensional charts have been given.

# Chapter 8

## *Assessing of interference effects on RSSI*

**R**ECEIVED signal strength indication (RSSI) consists of a simple channel power measurement that a wireless device performs within the assigned communication bandwidth. From RSSI estimates, the wireless transceiver usually classifies the status of the assigned channel as free or busy, and decides whether to transmit or not, according to the adopted channel access protocol. RSSI is also largely used in order to assess the useful power strength at the receiver side in order to perform automatic gain control to fit the received signal to the receiving circuitry. Two typical and widespread examples of wireless systems using RSSI are IEEE 802.11 [1] WLANs and IEEE 802.15.4 [2] WSNs.

A typical drawback of RSSI circuits is the poor measurement accuracy, essentially due to their simple and low cost architectures. Another important issue is the presence of interference, that may cause inaccuracy in the RSSI readings, and severe consequences in terms of network overall performance.

For instance, in the case of an erroneously detected free channel, a node may transmit despite the presence of other operating devices, causing disruptive collisions. Conversely, in the case of an erroneously detected busy channel, the node is forced to defer the transmission, with consequent delays, jitter, loss of data packets, etc.

In the literature, some papers specifically deal with interference issues in IEEE 802.11 and IEEE 802.15.4 wireless networks. For instance, in [25, 36], the mutual interference between IEEE 802.15.4 and 802.11b networks is investigated through simulations and measurements, respectively. In [24], the models of [21] are applied to the case an IEEE 802.15.4 network affecting a IEEE 802.11b network. In [26], results from measurements performed on an IEEE 802.15.4 network are offered.

Other useful information is given in [27], even though referred to the only case of an IEEE 802.11 network operating in the proximity of a Bluetooth network. From all such contributions, useful information and hints can be gained and used to improve the design of IEEE 802.11 and 802.15.4 networks.

After a brief overview on receiver circuitry and RSSI chip in particular, the effect of in band and out-of-band interference on CSMA/CA compliant devices is analyzed through the RSSI measurement under different interference scenarios. After that, an experimental analysis of non-ideal behaviors of RSSI is given, showing how such a power measurement can be corrupted by external interference.

All the material presented in this chapter is taken from [37, 38, 39].

## 8.1 Preliminary notes

In this section a brief description of integrated receivers and, in particular, RSSI circuitry is provided. For basic concepts on interference phenomena on CSMA/CA systems, *i.e.* PHY and MAC layer interference, the reader is referred to Section 7.1.

### 8.1.1 Low-IF receiver architecture

In modern WSNs, the usage of single chip technologies and new efficient radio architectures provides valuable advantages in terms of reduced costs, dimensions and power consumption per each single sensor device.

The most commonly used architectures are based on a conventional *heterodyne* scheme [40, 41], which performance are superior compared to other more integrable receiver architectures, with respect to selectivity and sensitivity. This is achieved with the use of high-quality-factor (Q) discrete components, *i.e.* inductors and varactor diodes. The use of such high-Q elements has also some drawbacks. A critical issue is that high-Q filters can not be easily realized in integrated solutions, *i.e.* the integrated inductors have at best only moderate Q-factors.

A second efficient architecture is the *direct conversion*, also known as *homodyne* or *zero-IF conversion* [40, 41], which is based on a direct down-conversion of the RF signal directly to base band. This architecture performs low-pass filtering in the baseband to suppress nearby interferers and select the desired channel. The main advantage of this architecture is its more suitability to monolithic integration.

Nevertheless, a number of non-negligible drawbacks are stressed in this configuration, like for instance: (i) the *DC-offset*, due to the self-mixing phenomenon of the Local Oscillator (LO), which can corrupt the desired signal and/or saturate the following stages; (ii) the *LO leakage*, due to the leakage of the LO signal to the antenna, which produces in-band interference; (iii) the *I/Q mismatch* that is greater for wide band and high-data rate systems.

A third architecture is the *Low-IF* [40, 41], whose topology is sketched in Figure 8.1. The goal of this scheme is to combine the advantages of both heterodyne and homodyne receivers.

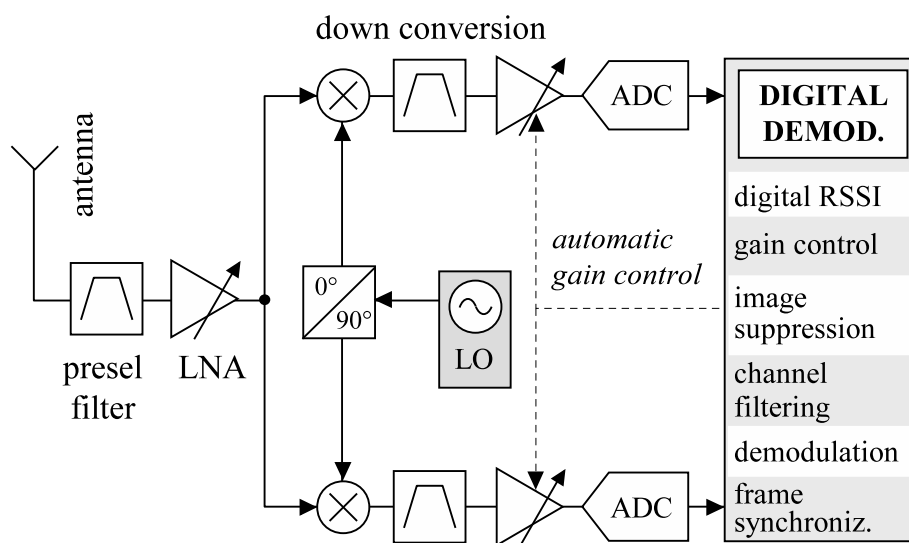


Figure 8.1: Architecture of a low-IF receiver.

In the figure, the RF front-end (pre-selection filter) is used to remove out-of-band signal energy and partially reject image band signals. After the pre-filtering, the received signal is amplified by a Low Noise Amplifier (LNA). The system then employs two quadrature down-conversion paths, whose outputs contain all the required information for the separation of the wanted signal from the unwanted signals, such as images. An important peculiarity of the low-IF receiver is just the Intermediate Frequency (IF), which is chosen as low as one or two times the channel bandwidth. This mitigates the DC offset problem, simply because after the first downconversion the wanted signal is not located around DC. A second characteristic of low-IF topology is the more suitability to sample the low-IF signal after the first mixer stage with a high resolution ADC. This sampling process requires an ADC with higher resolution because most of the image rejection task is done in the digital domain. After the first mixer stage, the unwanted image can be much larger than the desired signal.

Although the low-IF architecture requires higher-performance ADCs, the signal path to the ADC can be AC-coupled in the low-IF architecture, which in turn eliminates the need for complicated DC offset cancellation circuitry. Another advantage of this low-IF topology is that part of the complex image rejection mixer is implemented in the digital domain with no gain and phase I/Q mismatches. The I/Q imbalances introduced in the previous analog sections can be corrected using adaptive techniques.

A final Digital Signal Processor (DSP) following the ADCs is used to provide the following tasks: digital demodulation, digital RSSI, automatic gain control, image suppression, channel filtering, and frame synchronization.

### 8.1.2 RSSI architecture

The architecture of a generic integrated radio receiver supporting the RSSI function is sketched in Figure 8.2. The signal picked up by the antenna is conditioned by an analog stage, which performs the following tasks: (1) low noise amplification, (2) down conversion to a low intermediate frequency (IF) [40], (3) automatic gain control. The outgoing analog signal is then digitally converted by the ADC. A digital processing stage performs the tasks needed for the demodulation: (1) channel filtering, (2) image rejection, (3) DC cancellation, (4) signal equalization [40]. The signal is demodulated and the original transmitted information recovered. The RSSI circuitry can be implemented either in an analog or digital way. In the former case, the analog down-converted IF signal is filtered and amplified to provide a voltage value proportional to the signal power (typically in a logarithmic scale). In the latter case, the RSSI estimation is performed through a digital processing of the acquired samples. In the figure, both the possibilities are sketched in order to better highlight the location of the RSSI stage in the two cases.

In the following, the only digital RSSI case is considered, which at the moment is the solution most commonly adopted in the available integrated transceivers, as described in the previous section.

In this scenario, the incoming signal power  $p(k)$  at the instant  $k \cdot T_S$ , where  $k$  is an integer and  $T_S$  is the sampling time, is estimated as follows:

$$\hat{p}(k) = A \cdot \frac{1}{L} \sum_{l=0}^{L-1} x^2(k-l), \quad (8.1)$$

where  $\hat{p}(k)$  is the estimate of  $p(k)$ ,  $x(k-l)$  is the voltage sample at the instant  $(k-l) \cdot T_S$  with  $l \in \{0, 1, \dots, L-1\}$  and  $L$  is the number of the last acquired samples.

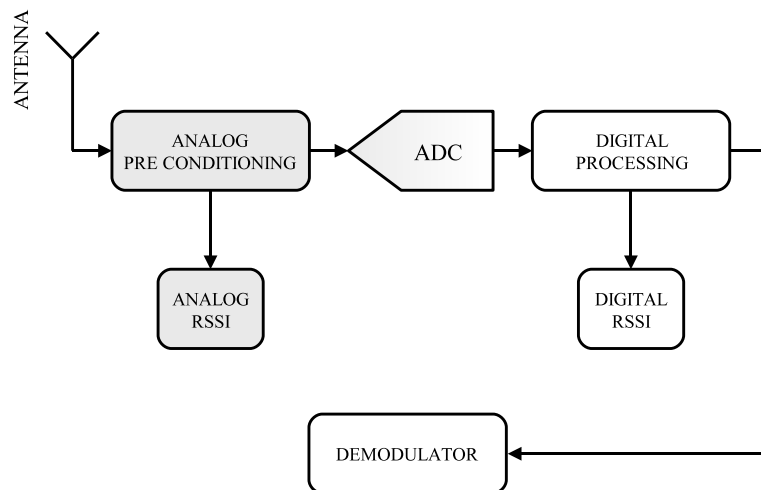


Figure 8.2: Architecture of a wireless receiver with RSSI stage.

$\hat{p}(k)$  is typically expressed in the logarithmic scale, in dBm units, and assuming a  $50 \Omega$  input impedance (using the weighting constant  $A$  to arrange the formula).

### 8.1.3 RSSI common applications

The RSSI function is widely employed in modern analog and digital communication receivers to accomplish one or few of the following specific tasks.

- **Pre-conditioning:** a common application of RSSI is the automatic gain control in the pre-conditioning stage of the receiver. Depending on the estimated  $\hat{p}(k)$ , the gain of the pre-conditioning stage is adapted and the incoming signal level amplitude varied in order to suitably match the ADC input range.
- **Transmission power control:** in battery-powered wireless devices, such as mobile phones of second- and third-generation, the information obtained by the RSSI function at the receiver side is used to optimize the transmission power and thus to save energy.
- **Channel access:** in non-scheduled wireless networks based on the CS-MA/CA protocol, wireless nodes use RSSI to evaluate the status (free or busy) of the communication channel. The transmission can start only if the measured power  $\hat{p}(k)$  is below a given threshold. Both IEEE 802.15.4 and 802.11 standards, being designed even for non-scheduled communications, use this kind of channel access method.

- **Rate adaptation:** in many communication standards, such as IEEE 802.11, RSSI is used to determine the signal to noise ratio at the receiver input connector and, accordingly, set the most appropriate modulation scheme and data rate.

Besides these common applications, RSSI can also advantageously be used in many other contexts, even if less conventional, such as for instance the following two ones.

- **Localization:** wireless nodes with RSSI are used to determine the position of objects or people in a given monitored area. The RSSI estimated power levels,  $\hat{p}(k)$ , measured by the node, are processed to infer the distance between the node and a set of suitably positioned and configured transmitters [42].
- **EM field monitoring:** wireless nodes with RSSI can also be used for distributed measurements of the electromagnetic field intensity in a given monitored area, such as industries, buildings, roads, or campuses.

It should be noted that in the former tasks, rough estimates of RSSI are quite always sufficient to adequately accomplish the respective targets. On the contrary, for the latter two tasks, an high accuracy level of RSSI estimates is quite always required. In this case, the accuracy of the RSSI estimates is a parameter to be carefully assessed and optimized.

## 8.2 Out-of-channel interference

In this section a simplified description of a IEEE 802.15.4 compliant Low-IF receiver is introduced to better understand the effects of interference on these kind of devices. After that a theoretical and simulative analysis of such a receivers is faced and an experimental validation is offered.

### 8.2.1 IEEE 802.15.4 receiver architecture

IEEE 802.15.4 radio transceivers are commonly integrated on a single chip [7]. A simplified schematic representation of a typical radio receiver chip in a wireless network node is depicted in Figure 8.3. In the figure, a first analog conditioning stage performs the following tasks: (1) low noise amplification, (2) down conversion to a low intermediate frequency (IF), i.e. typically once or twice the signal bandwidth, (3) automatic gain control. The signal is then digitally converted by the ADC, which gives a sequence of samples  $x(k)$ ,  $k \in \mathbb{Z}$ .



A digital band pass filter with impulse response  $h(k)$ , is then used to limit the spectrum of  $x(k)$  to the desired receiver channel bandwidth. When the node is receiving, the output signal  $y(k) = h * x(k)$ , where  $*$  denotes the convolution, is provided to the demodulator stage. This latter block executes the following tasks: matched filtering, chips flow decoding, despreading, etc. In the figure, the matched filter is represented by the block with impulse response  $g(k)$ , and its output signal is  $z(k) = g * y(k)$ . When the node is transmitting, the output  $y(k)$  is instead provided to the RSSI, which performs the measurement of the received in-channel power, as required by the CCA protocol (energy detection mode).

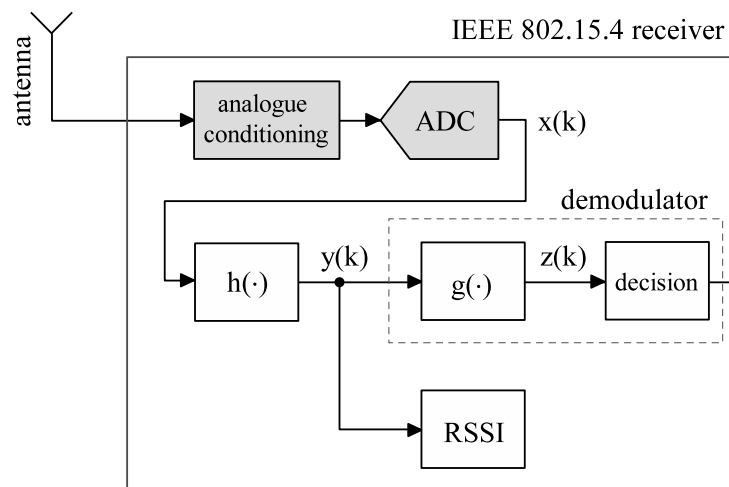
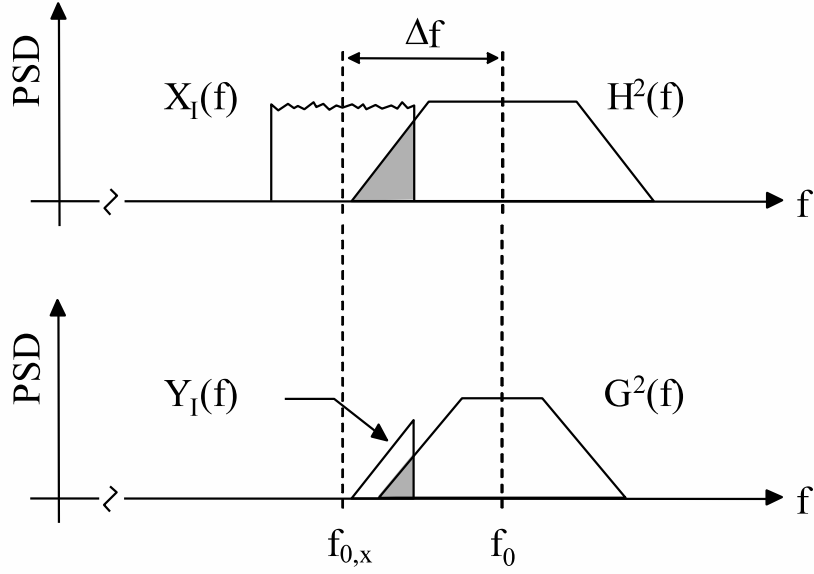


Figure 8.3: Simplified architecture of a IEEE 802.15.4 receiver.

### 8.2.2 Receiver frequency response

Out-of-channel interference occurs when part of the spectral content of a signal centered in an external channel overlaps the channel of interest. A qualitative example of out-of-channel interference is shown in Figure 8.4. In the figure,  $X_I(f)$  and  $Y_I(f)$  represent the Power Spectral Density (PSD) of the signals  $x(k)$  and  $y(k)$  of Figure 8.2, respectively, in the presence of the only interference at the antenna output connector, while  $|H(f)|^2$  and  $|G(f)|^2$  represent the square of the frequency responses of the  $h(k)$  and  $g(k)$  filters, respectively.  $f_{0,x}$  and  $f_0$  are instead the center frequencies of  $X_I(f)$  and  $H(f)$ , respectively, separated with each other by a frequency offset  $\Delta f$ . The effect of the interference on the victim system performance strictly depends on the power level assumed by the two signals  $y(k)$  and  $z(k)$  of Figure 8.2, here denoted as  $P_{I_y}$  and  $P_{I_z}$ , respectively, for the only interference contribution.



**Figure 8.4:** Influence of out-of-channel interference within the WSN bandwidth.

In the plot of figure, such two quantities are represented by the intersection areas (gray) between the two regions delimited by the PSD functions  $X_I(f)$  and  $Y_I(f)$  and the corresponding  $|H(f)|^2$  and  $|G(f)|^2$  curves. They can be determined as follows:

$$P_{I_y} = \int_{-\infty}^{+\infty} |H(f)|^2 X_I(f) df \quad (8.2)$$

$$\begin{aligned} P_{I_z} &= \int_{-\infty}^{+\infty} |G(f)|^2 |H(f)|^2 X_I(f) df \\ &\approx \int_{-\infty}^{+\infty} |G(f)|^2 X_I(f) df \end{aligned} \quad (8.3)$$

where, in (8.3), it is implicitly assumed a  $g(k)$  filter much more selective than  $h(k)$ , i.e.:  $G(f) \cdot H(f) \approx G(f)$ . It is also observed that the WSN performance are strictly dependent on both  $P_{I_y}$  and  $P_{I_z}$ , even if at two different levels. Specifically,  $P_{I_y}$  acts at MAC layer, when the node is transmitting, and affects the PER according to:

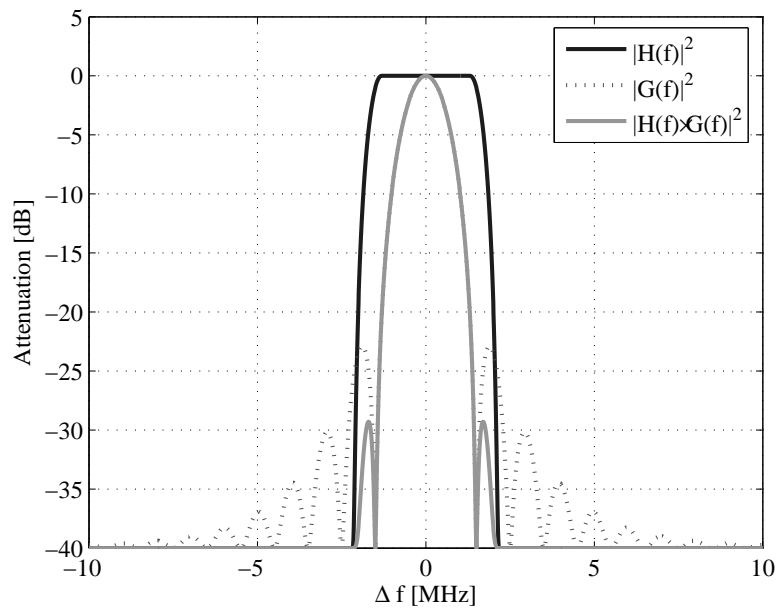
$$PER(P_{I_y}, P_{th}) = H(P_{I_y} - P_{th}), \quad (8.4)$$

where  $H(\cdot)$  is the Heaviside step function (see [34]) and  $P_{th}$  is the user selectable CCA threshold. In fact, as soon as  $P_{I_y}$  overpasses  $P_{th}$ , the channel is assumed busy, and no more transmissions are allowed.

This abruptly reduces the number of received packets and consequently increases the PER to 1.  $P_{I_z}$  instead acts at PHY layer, when the node is receiving, and affects the PER in terms of signal quality degradation. In particular,  $P_{I_z}$  degrades the SIR, defined as  $SIR = P_{S_z}/P_{I_z}$ , where  $P_{S_z}$  is the received useful power at the output of the  $g(k)$  filter, and consequently the PER. In this case, estimates of PER, for a given SIR, can easily be obtained by using theoretical or experimental curves like those shown in Figure 7.2.

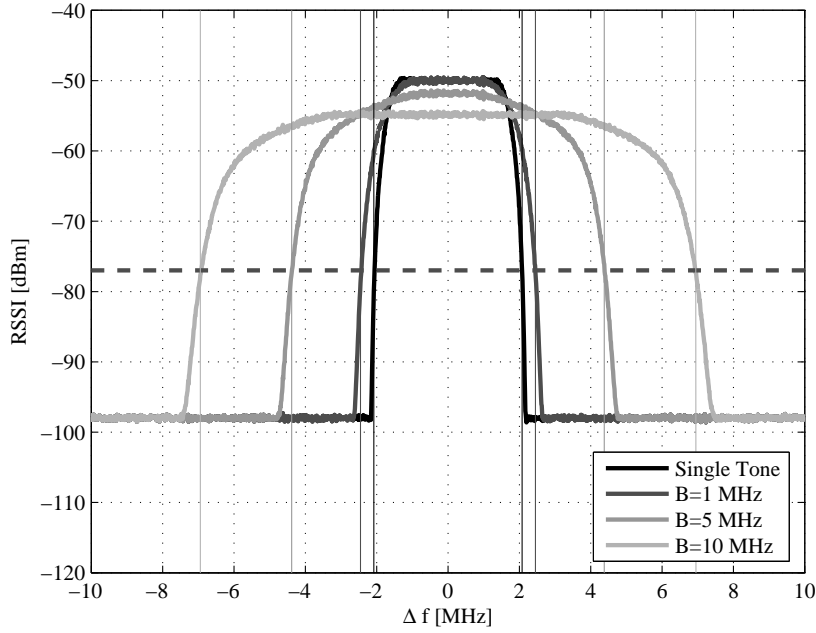
### 8.2.3 Simulation

A number of simulations have been carried out by modeling the operation of the receiver in Figure 8.2, and using parameters of a real receiver compliant with the IEEE 802.15.4 standard. One meaningful result is shown in Figure 8.5.



**Figure 8.5:** Out-of-channel interference attenuation due to receiver's filters.

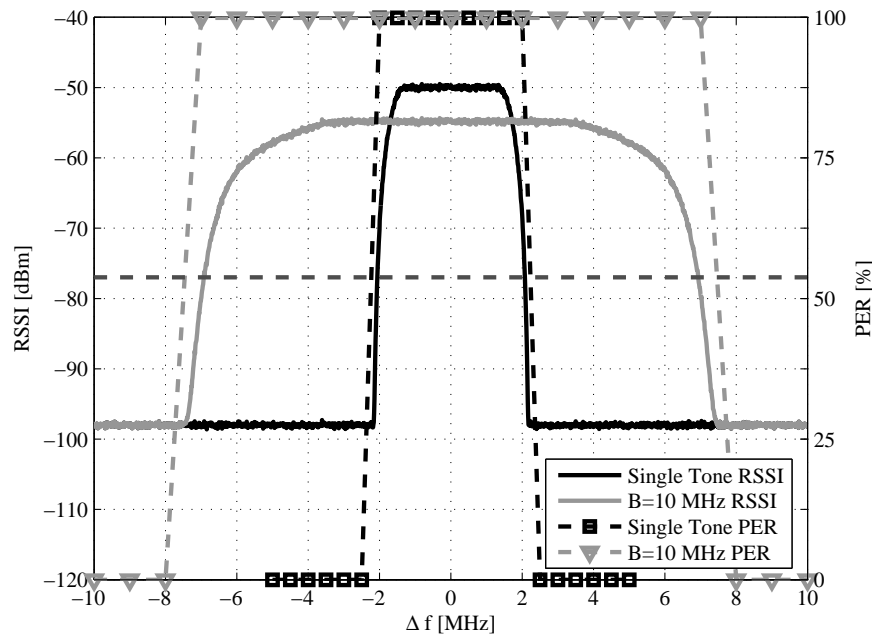
The plot provides the square magnitude of the frequency responses of the two filters  $h(k)$  and  $g(k)$ , and of their series  $h * g(k)$ , namely  $|H(f)|^2$ ,  $|G(f)|^2$  and  $|H(f) \cdot G(f)|^2$ , respectively. For more generality, the x-axis represents the frequency offset from the WSN center frequency. The figure clearly highlights that  $G(f)$  bandwidth is effectively narrower than the one of  $H(f)$ , and that  $G(f) \cdot H(f) \approx G(f)$ . This implies that the out-of-channel interference power measured at MAC layer is greater than the one acting at PHY layer, i.e.  $P_{I_z} < P_{I_y}$ .



**Figure 8.6:** RSSI estimates (simulated  $P_{I_y}$ ) vs interference offset for different interference types.

A second interesting result is given in Figure 8.6. The figure provides the RSSI estimates,  $P_{I_y}$ , resulting from the analysis of different kinds of interferer and upon the varying of the interference offset  $\Delta f$ . Two types of signals have been analyzed. The first is a single tone interference at a frequency  $f_{0,x}$ . The second is a band limited AWGN signal of bandwidth  $B$  and centered at  $f_{0,x}$ . Specifically, the following bandwidths have been investigated:  $B = 1, 5, 10$  MHz. In order to better highlight the interference effect at MAC layer, which, as above quoted, represents the case less investigated in the literature, the power of both type of interference signals,  $P_{I_x}$ , has been set equal to  $-50$  dBm, *i.e.* greater than the selected CCA threshold ( $P_{th} = -77$  dBm). The figure clearly highlights that the presence of band limited noise-like interference centered in the range  $-\frac{B}{2} - 2 \text{ MHz} < \Delta f < \frac{B}{2} + 2 \text{ MHz}$  may lead to significant levels of RSSI even if not centered in the pass-band of the system. The curves can also be used in order to efficiently investigate on the origin of a given loss of PER or to optimize the choice of the CCA threshold. In fact, once drawn  $P_{th}$ , in the figure represented by the horizontal line ( $-77$  dBm), the  $\Delta f$  domain can be easily divided into two disjoint sets: one for which  $P_{I_y} > P_{th}$ , where MAC layer interference dominates, and another, for which  $P_{I_y} < P_{th}$ , where PHY layer interference dominates.

This effect is simulated in Figure 8.7, where only the single tone and the 10 MHz bandwidth AWGN are sketched. The right vertical axis shows the PER, that follows exactly the threshold-like behavior just predicted. When the RSSI is just above the chosen CCA threshold (in this case  $-77$  dBm), no transmission is possible at all, causing a 100% packet loss.



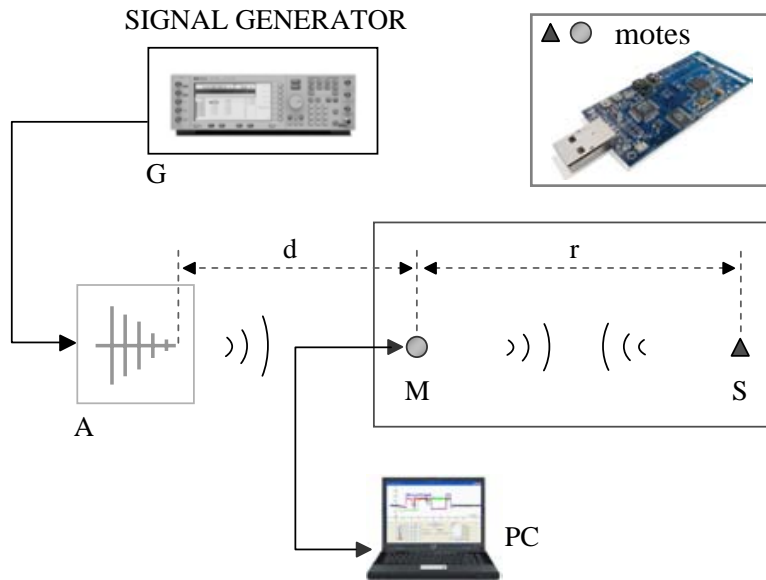
**Figure 8.7:** RSSI and PER (simulation) vs interference offset for single tone and 10 MHz-bandwidth interference.

### 8.2.4 Testbed

A number of experiments have been carried out by using a purposely developed testbed to validate the results obtained in section 8.2.3. The testbed, as visible in the sketch of Figure 8.8, consists of a simple WSN, made of only two wireless sensor nodes (motest), one acting as master, M, and the other as slave, S. Wireless nodes are Tmote Sky sensors compliant with IEEE 802.15.4, available from Moteiv [5]. The mounted radio apparatus, CC2420 from Chipcon [7], allows the choice of CCA mode and threshold and embeds a RSSI chip (as depicted in Figure 8.2), that provides an estimate of in-channel power,  $P_{I_y}$ .

A PC is connected to M via a wired link, for storing collected data and post-processing. This simple network performs two tasks. First, it senses the RSSI at the master node with a sample-time of 10 ms, downloading the acquired readings to the PC.

The values so obtained, once properly converted into dBm units, provide estimates of the power effectively measured at the antenna output connector of motes. Second, it evaluates the PER using a master-slave communication. In particular, traffic is generated and monitored through the tool WSN Factory Sniffer [16], which allows the setup of M and S and displays the data acquired from the embedded light sensors. In this case M executes a periodical polling of S (polling request), which replies sending the acquired data (lightness samples in this case). An ACK frame is used to confirm the correct reception; in case of not received ACK, polling is assumed failed and no retransmission is performed. The evaluation of PER is based on the number of correctly received pollings compared to the total number (1000) of pollings attempted.



**Figure 8.8:** Testbed of the simplified analysis scenario.

For all the performed tests, polling period has been set equal to 30 ms, which is the minimum value allowing lossless transmission in the absence of interference [16]. The channel number 26 centered at 2480 MHz has also been chosen in order to minimize frequency overlap with bands of near-operating Wi-Fi systems. Several other IEEE 802.15.4 channels have been investigated in the tests, obtaining almost identical results, here omitted for the sake of conciseness. An arbitrary radio frequency signal generator, G, Agilent Technologies AT E4433B (250 kHz–4 GHz) and a log-periodic antenna, A, EMCO 3146 (220 MHz–1 GHz), oriented toward M, have been used to emulate interference. Despite the limited range of frequencies of A, a good level of interference power has been obtained at the WSN position also at 2.480 GHz.

In particular the same interference power level of -50 dBm at the master output connector considered in the simulations of Section 8.2.3 has been obtained. Two interference signals have been considered in the tests: a single tone signal and a AWGN interference of variable frequency (central frequency)  $f_{0,x} + \Delta f$ , with  $f_{0,x}$  and  $\Delta f$  having the same meaning of Section 8.2.2. The AWGN solicitation presents a spectral content rather similar to which of some typical interfering contributions of ISM band, like for instance Wi-Fi signals [1]. The power on the generator G has been set in such a way to achieve -50 dBm at the mote's receiver with a single tone. The same power has been used for the AWGN signals. Testbed has been physically positioned as shown in figure, above wood tables at one meter from ground floor, and with  $d = 1$  m, and  $r$  variable. Tests have been performed in open-site conditions (but not in an anechoic chamber), sufficiently far way from external interfering sources. In the absence of intentional interference, i.e. disabling G, a null PER has been in fact assessed.

### 8.2.5 Experimental results

The above described testbed and setup have been adopted in a number of experiments aimed at investigating on the real effects of out-of-channel interference on a WSN, with special regard to MAC layer interference. In Figure 8.9, a first set of experimental results are shown in terms of RSSI readings versus  $\Delta f$ , i.e. the frequency offset defined in Section 8.2.2. As can be seen, the obtained results of Figure 8.9 agree very much with the RSSI values of Figure 8.6, obtained through simulations at the same setup conditions adopted for measurements. For example, in the case of a 10 MHz bandwidth AWGN interference and with  $\Delta f = -5$  MHz, the estimated RSSI is -57 dBm, from simulations, and -58 dBm, from measurements. This confirms that the provided simplified approach is really accurate in order to simulate the behavior of a IEEE 802.15.4 receiver with respect to the interference at MAC layer.

The figure also highlights the appearance of undershoots at  $\Delta f = \pm 2$  MHz, which propagate even below the noise floor (typically -98 dBm). Such an unusual fact can be justified by the occurrence of a saturation phenomenon in the ADC or in the low-IF filtering stages of the wireless node. This saturation is probably due to the high energy concentration of the single-tone interferer especially when centered at the local oscillator frequency, i.e. for  $\Delta f = \pm 2$  MHz. In fact, the phenomenon disappears for AWGN signals, which power is not concentrated at a unique frequency, but is spread over a large band.

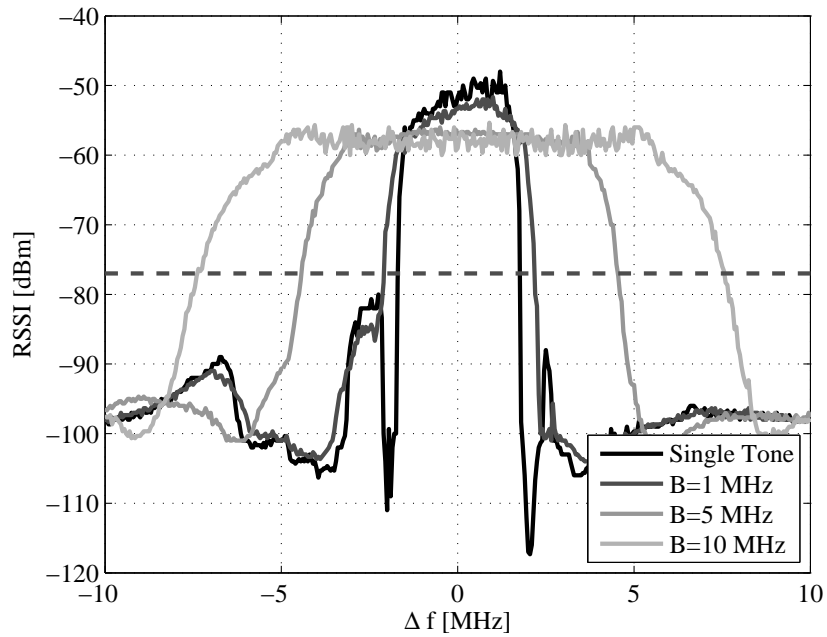


Figure 8.9: RSSI estimates (measured) vs interference offset and for different interference types.

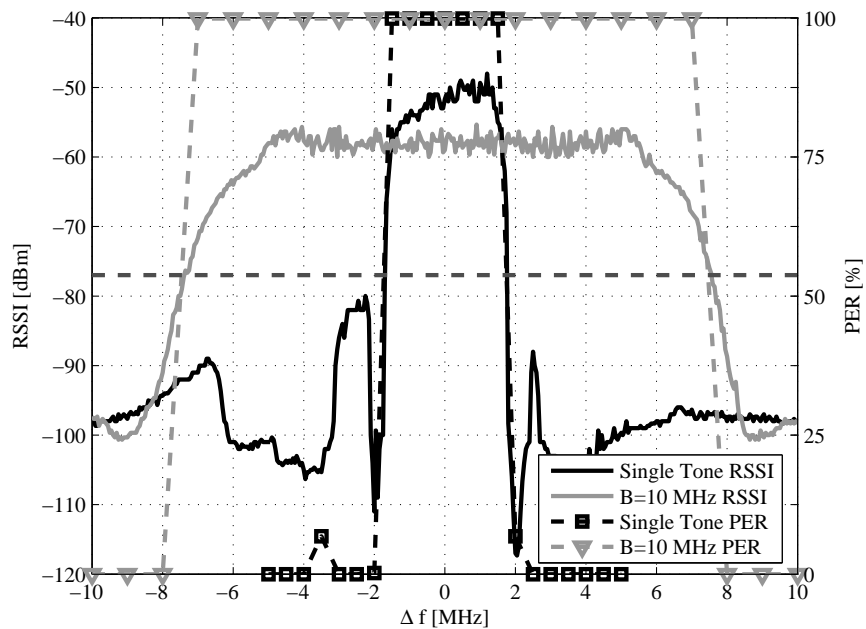


Figure 8.10: RSSI and PER measurement vs interference offset for single tone and 10 MHz-bandwidth interference.



It is then observed that for AWGN interference with bandwidth,  $B$ , larger than the channel filter pass-band, the values of RSSI for  $|\Delta f| < 2$  MHz may be lower than the overall interference power (-50 dBm) at the WSN node antenna output connector. In other words, only a fraction of the interference power is sensed by the RSSI device, leading to RSSI estimates lower than -50 dBm, *e.g.* nearly equal to -57 and -58 dBm for  $B = 5$  and 10 MHz respectively. The presence of AWGN interference with a large bandwidth  $B$  enlarges the region of  $\Delta f$  values for which the power is greater than the selected CCA threshold (here -77 dBm). This can be seen in Figure 8.10, where RSSI estimates (at physical layer) are shown along with the corresponding estimates of PER (at network layer).

Also in this case a strict agreement can be noticed between simulation (Figure 8.7) and measurement results (Figure 8.10). The diagram confirms that as soon as the RSSI estimate overpasses  $P_{th}$ , the PER abruptly worsens to 100 % and no packets are thus transmitted. When the power is instead below this threshold, interference effects are visible at only physical layer. In this latter case, the values of PER strictly depend on SIR, as can be seen from the results summarized in Figure 7.2 obtained upon the varying of SINR. The effect of SIR is instead less visible in Figure 8.10. In fact, it has been chosen on purpose high enough to ensure a zero packet loss and thus to better highlight the interference effect at MAC layer.

## 8.3 Impairments on RSSI

In this section an experimental analysis shows how RSSI impairments may influence the performance of a IEEE 802.15.4 wireless sensor network in the presence of interference.

### 8.3.1 Measurement setup

The testbed used to perform all the experiments is similar to the one depicted in Figure 8.8 and described in Section 8.2.4.

In the testbed the sensing device,  $M$ , is used simply as a generic in-channel power meter. In particular, the testbed is arranged in order to focus the attention on the only RSSI detector operation, upon the varying of the interference scenario. One should note that the estimates it provides, once properly converted into dBm units, represent measures of the power effectively flowing at the mote's antenna input connector. Specifically, the analysis is carried out with the following types of interference:

1. a single tone waveform of power  $P_S$  and frequency  $f_x$  defined as  $f_x = f_c + \Delta f$ , with  $\Delta f$  variable in the range  $[-10, +10]$  MHz, and  $f_c$  center frequency of the deployed IEEE 802.15.4 channel.  $P_S$  is also varied in the range  $[-140, -20]$  dBm;
2. a couple of single tone waveforms of frequency  $f_x$  and  $f_y$ , with  $f_x = f_c$  and  $f_y = f_c \pm \Delta f$ , with  $\Delta f = \pm 2$  MHz. In this case the sinusoid at frequency  $f_x$  represents the useful signal, with a variable power  $P_S$  in the range  $[-100, -50]$  dBm, while the one varying at  $f_y$  is an out-of-band interference with power  $P_I$ ;
3. a single tone and band-limited AWGN interference with bandwidth  $B = 1, 5, 10$  MHz, and centered at  $f_x = f_c + \Delta f$ , with  $\Delta f$  in the range  $[-10, +10]$  MHz.

In order to check the repeatability of the performed measurements, an high number of RSSI readings is taken (every 10 ms). Experiments are conducted in a non-anechoic room, sufficiently far away from non-controllable interfering sources. The investigated interference central frequency,  $f_c$ , is then chosen in order to avoid possible overlap and interference with other surrounding sources, e.g. the near operating Wi-Fi systems. Testbed is physically positioned above wood tables at one meter from the ground floor.

### 8.3.2 Analysis of the RSSI meter frequency response

A first set of experiments has been focused on the analysis of the RSSI meter frequency response. To this aim, the single tone waveform defined in Section 8.3.1 (interference n. 1) has been used. The obtained results in terms of RSSI readings are shown in Figure 8.11 upon the varying of the sinusoidal waveform frequency,  $f_x = f_c + \Delta f$ , in the range  $[f_c - 10, f_c + 10]$  MHz, and for three power levels of the sinusoid,  $P_S = -30, -50$  and  $-70$  dBm.

The figure clearly shows that in the center of the RSSI meter bandwidth, *i.e.*  $-2 \text{ MHz} < \Delta f < +2 \text{ MHz}$ , the three curves are shifted by 20 dB (along the RSSI axis), as expected by the fact that  $P_S$  varies in the three cases by 20 dB. The curves are very flat in the passband, even if, in the case of  $P_S = -30$  dBm, the measured values are nearly 5 dB above the real value. In the stop band, *i.e.* for  $|\Delta f| \geq 2 \text{ MHz}$ , an high attenuation is expected, although in the cases of  $P_S = -30, -50$  dBm a strongly irregular trend can be noted. In particular, non-negligible undershoots occur for  $\Delta f = \pm 2 \text{ MHz}$  and, especially for  $P_S = -30$  dBm, visible spurious images appear in the stop band.

In the case of  $P_S = -70$  dBm, the RSSI values in the stop band instead converges quite regularly to the noise floor (approximately equal to -96 dBm). The same experiments have been repeated with different devices belonging to the same family, and deploying, one by one, all the IEEE 802.15.4 channels available in the ISM band. In all the considered cases, the same undershoot phenomenon for  $P_S = -30, -50$  dBm has been observed. Such a fact suggests the presence of a sort of saturation phenomenon in the device, which depends on the values assumed by  $P_S$ .

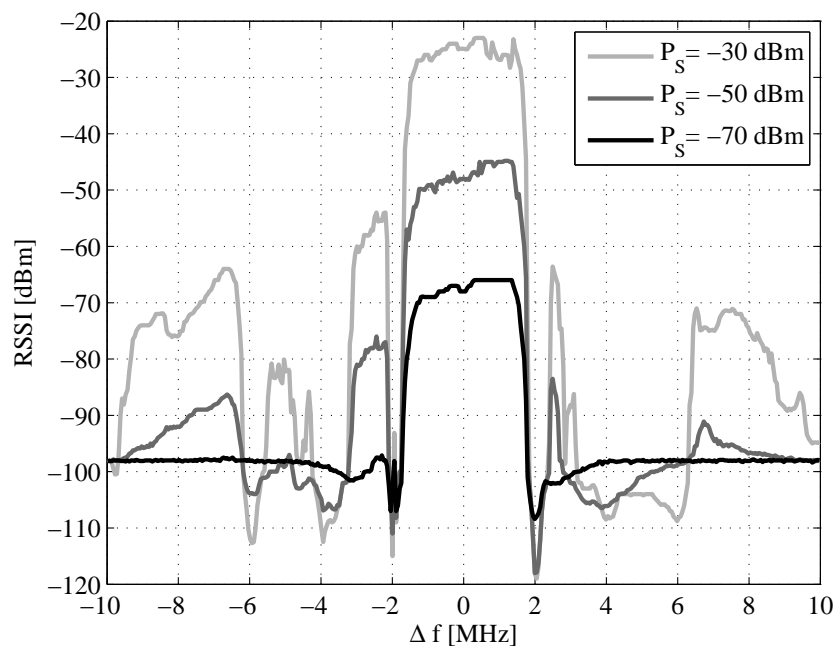
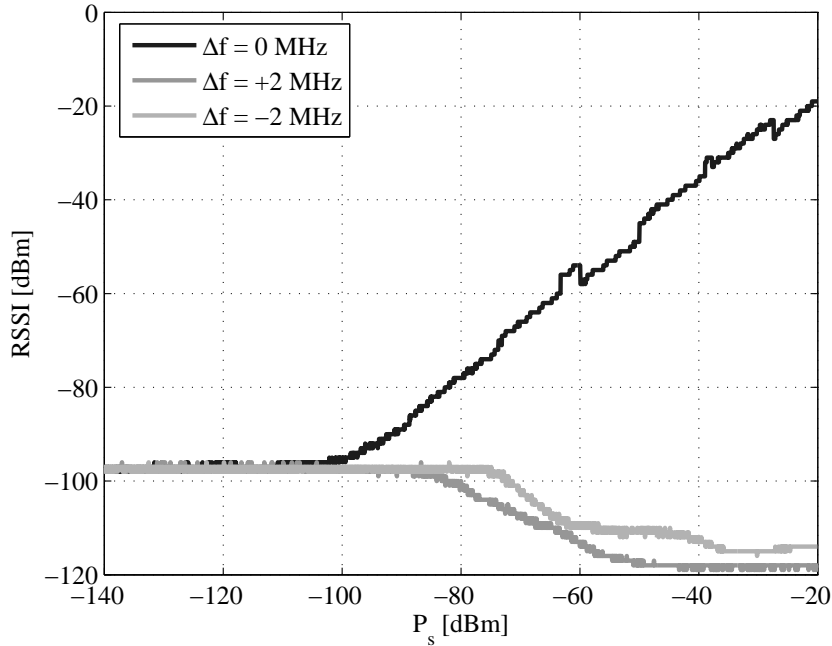


Figure 8.11: Estimated RSSI vs frequency offset  $\Delta f$ .

In Figure 8.12, the obtained RSSI estimates versus  $P_S$ , in the range  $[-140, -20]$  dBm, are summarized for three different frequency offsets:  $\Delta f = 0, +2, -2$  MHz. The figure highlights a linear trend with unitary slope between RSSI and  $P_S$ , for  $\Delta f = 0$  MHz and  $P_S \geq -96$  dBm, while below this power threshold, the only noise floor is sensed. For  $\Delta f = +2, -2$  MHz, *i.e.* with the interference sinusoid frequency just outside the RSSI bandwidth, the measured power assumes very low levels, as expected. Nevertheless, an unexpected strange trend can be observed also in this case. In particular, the values assumed by the RSSI detector are here below the noise floor. This fact could be attributed to the already cited undershoot phenomenon due to saturation at the RSSI detector.

A third set of experiments have been conducted in the presence of the above described single carrier and AWGN signal (interference n. 3).



**Figure 8.12:** Estimated RSSI vs  $P_s$ , for three different frequency offsets:  $\Delta f = 0, +2, -2$  MHz.

The results shown in Figure 8.9 provide the RSSI estimates upon the varying of the carrier center frequency  $f_c$ , or equivalently  $\Delta f$  in the range  $[-10, +10]$  MHz. Different AWGN bandwidths  $B = 1, 5, 10$  MHz have also been considered. The results highlight also in this case the presence of the above observed undershoot phenomenon, which arises only in the case of single tone signal, while for larger  $B$  the phenomenon disappears. The larger bandwidth of the RSSI response is instead due to the larger input signal bandwidth  $B$ .

### 8.3.3 Interference effects on RSSI readings

A further set of experiments has been performed to better investigate on the consequences of the saturation effect, above observed, on the RSSI performance. To this aim, the couple of sinusoids previously defined (interference n. 2) has been used. In particular, the sinusoid at frequency  $f_x$ , tuned at the RSSI meter center frequency ( $f_x = f_c$ ) emulates a typical in-bandwidth useful signal of power  $P_s$ , while the one varying at frequency  $f_y$  emulates an out-of-band interference of power  $P_I$  tuned exactly just at the undershoot frequencies of Figure 8.11,  $f_x = f_c + \Delta f$ , with  $\Delta f = \pm 2$  MHz. The results of the obtained experiments are shown in Figure 8.13, in terms of RSSI readings versus the useful power  $P_s$ , variable in the range  $[-100, -50]$  dBm, and with  $P_I = -50$  dBm.

The results show an interesting fact: the RSSI meter correctly works in the absence of interference, while it fails (very low RSSI levels) in the presence of out-of-band interference. In particular, in this latter case, a threshold relationship between RSSI and  $P_S$  can be noted. Specifically, if  $P_S \geq P_S^*$ , with  $P_S^*$  threshold value equal to  $-65$  and  $-70$  dBm, respectively for  $f_y = -2, +2$  MHz, the power is correctly measured, while if  $P_S < P_S^*$ , completely erroneous and underestimated values of RSSI are achieved.

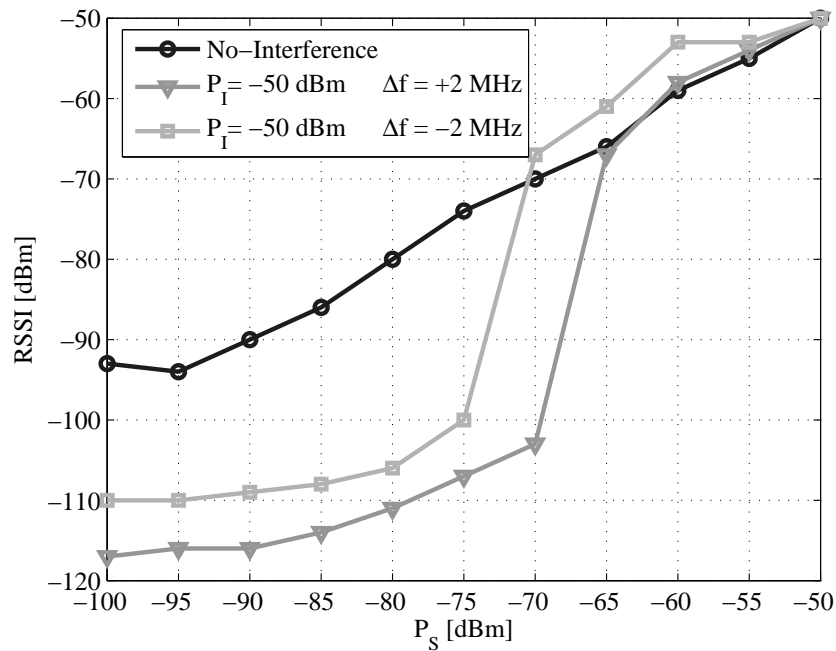
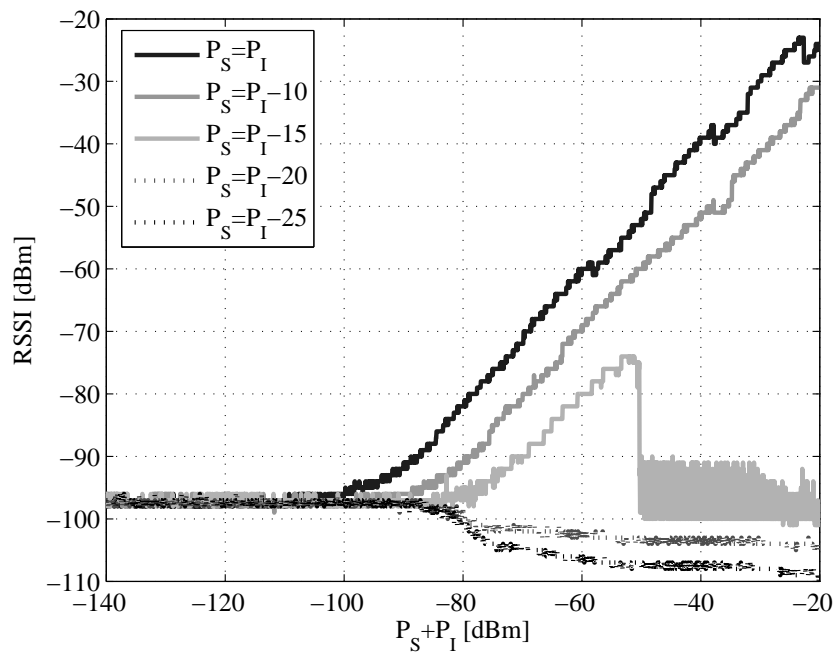


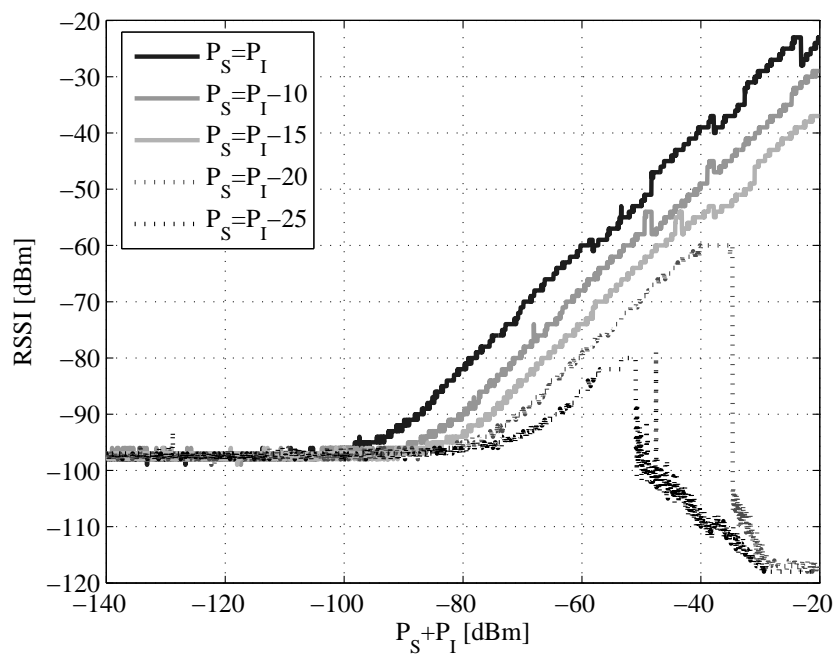
Figure 8.13: RSSI reading error due to interference.

A last set of experimental results are given in Figure 8.14 to underline the effects of the above observed phenomenon. In this case,  $P_S + P_I$  has been linearly varied in the range  $[-140, -20]$  dBm. Each curve has been obtained with a fixed attenuation,  $\Omega$ , of the useful signal  $P_S$  with respect to  $P_I$  ( $\Omega = P_I - P_S$ ).

Theoretically, each straight line should maintain a unitary slope, with a simple shift in the RSSI axis equal to the chosen  $\Omega$ . Instead, it is clear that for  $\Omega > 15$  dB and  $f_y = +2$  MHz, or  $\Omega > 20$  dB and  $f_y = -2$  MHz the presence of interference can significantly degrade the RSSI reading.



(a)



(b)

**Figure 8.14:** RSSI reading affected by interference for: (a)  $\Delta f = +2$  MHz  
(b)  $\Delta f = -2$  MHz.

## 8.4 Conclusions

The behavior of a IEEE 802.15.4 WSN with CSMA/CA may be seriously affected by out-of-channel interference, in terms of parameters such as the packet error rate. In this paper, such effects have been investigated with special regard to interference at MAC layer, which is traditionally less considered in a design stage with respect to the well-known PHY layer counterpart. With respect to single tone interference, AWGN signals may interfere significantly even when they are centered at frequencies outside the bandwidth of the WSN. In this case, the effects of interference can be accurately predicted from the analysis of diagrams like those shown in Figures 7.2, 8.9, and 8.10. In particular, from Figure 8.9 (or from its counterpart obtained from simulations, shown in Figure 8.6) one can easily estimate the range of frequency offset  $\Delta f$  where the interference effect at MAC layer dominates the PHY layer one, or *vice versa*. In the former case, the PER can be estimated by using the results of Figure 8.10, regardless the value assumed by the SIR. In the latter case, the PER can be evaluated, for a given SINR, by using the experimental curve of Figure 7.2. Such diagrams can be derived both theoretically, by simulations, and experimentally, by measurements, to be conducted as described in the paper.

A number of suitable experiments has revealed the importance of a saturation phenomenon arising in the RSSI operation under stressful conditions. The worst behavior has been observed with a narrow band or single tone interference centered at  $f_y = \pm 2$  MHz with respect to the RSSI center frequency. This problem can be attributed to the LO leakage and the DC offset phenomena acting at the IF stage of such an integrated receiver [40, 41]. This fact can have severe consequences on the operation of a wireless sensor network. In fact, in the presence of single tones or narrow band signals in the nearness of the undershoot frequencies, a wireless sensor can be unable to detect the presence of other transmitting devices, with consequences in terms of possible collisions.





# Chapter 9

## *RSSI based RF power measurement*

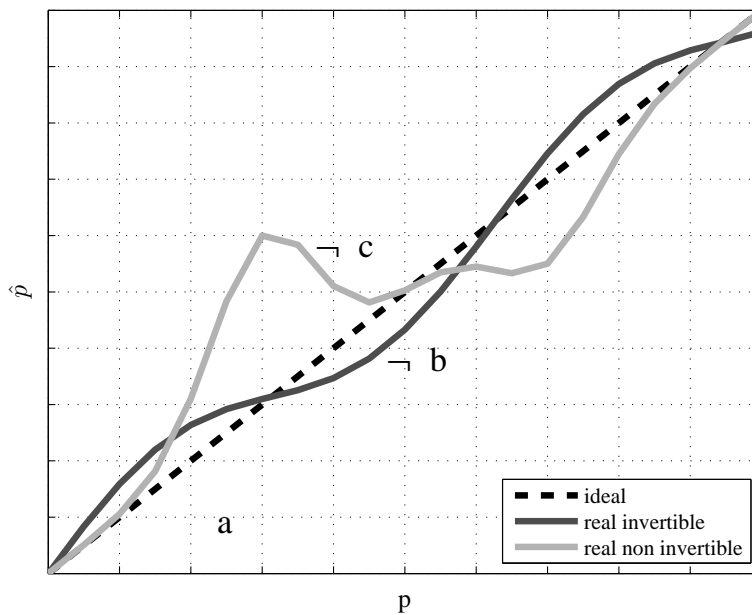
**T**HE use of the RSSI function for radio frequency power measurements is investigated. The analysis is carried out both conceptually, recalling fundamental details about the operation of a RSSI device, and experimentally. A procedure is proposed for improving the use of RSSI for power measurement tasks, especially in terms of non-linearity errors and accuracy. The ultimate purpose is to show how the RSSI function can even be deployed for accurate power measurement purposes. To this aim, a suitable testbed is arranged and a measurement campaign conducted by using a suitably developed semi-automatic testbed based on IEEE 802.15.4 compliant wireless network nodes. Experimental results are finally presented proving the effectiveness of the proposed procedure. RSSI basics have been discussed in Chapter 8. All the material discussed in this chapter has been taken from [39].

### **9.1 Single input approach**

A procedure is proposed aimed at improving the accuracy of a wireless communication node when deployed for RSSI measurements. The procedure is based on a compensation of the ADC non-linearity, which is performed by adding to the obtained estimates  $\hat{p}(k)$  a given correction offset to be *a-priori* determined by means of a preliminary calibration procedure. The overall procedure can be thus subdivided into two steps: (A) calibration and (B) correction step. This is a single input approach (punctual) trying to estimate the “real” input power with a single measure.

### 9.1.1 Calibration step

In Figure 9.1, three examples of possible RSSI transcharacteristics are reported, where the axes units are omitted for more clarity. The (a) curve represents the ideal case, in which the relationship between the measurand  $p(k)$  and the associated estimate  $\hat{p}(k)$ , hereinafter more simply referred to as  $p$  and  $\hat{p}$  respectively, is perfectly linear. The calibration procedure, in this case, is very simple: only the slope and offset of the line must be estimated.



**Figure 9.1:** Example of RSSI transcharacteristic: a) ideal, b) non-ideal invertible, c) non-ideal non-invertible.

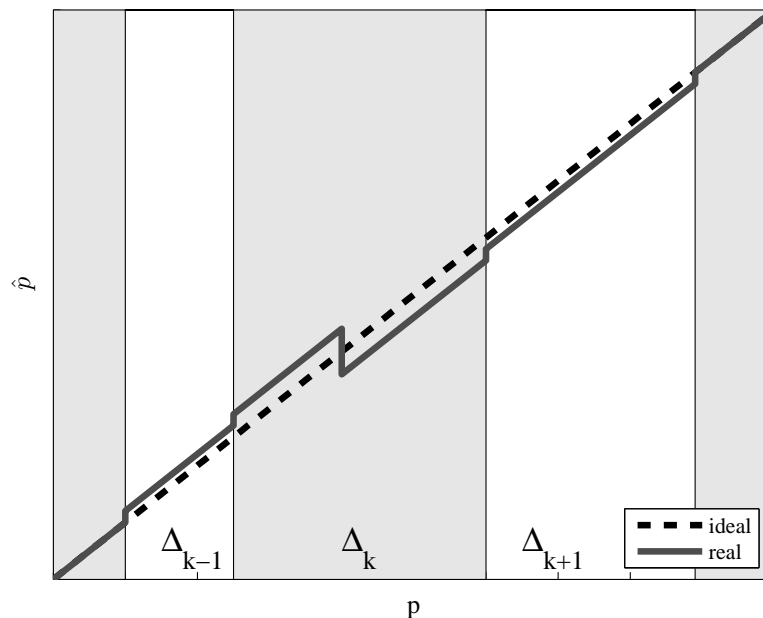
The (b) curve represents the case in which the RSSI transcharacteristic is not linear but still invertible. In this case, the curve can be fitted to a  $n$ -degree polynomial or a sum of basic functions. The calibration procedure simply consists of estimating the coefficients of the polynomial fit or of the sum. This procedure is simple in the case of quite regular transcharacteristics, but very complex in the case of ADC characterized by a strong non-linear behavior. In this latter situation, a point-to-point measurement of the transcharacteristic is needed on a number of points,  $M$ , to be chosen according to desired accuracy level. The (c) curve denotes the case of a non-linear and non-monotonic transcharacteristic. Also in this circumstance, a point-to-point calibration approach is needed and, moreover, there exist different values of  $p$  mapped into the same  $\hat{p}$ .

In this case, the correction of the estimates  $\hat{p}$  has to take into account the presence of all these input values  $p$  associated to  $\hat{p}$ . The starting point for the error minimization is the input-output transcharacteristic.

The model assumes the following relationship:

$$\hat{p}(p) = p + e(p), \quad (9.1)$$

where  $\hat{p}(p)$  is the estimated power, as a function of the real power at the input connector,  $p$ . This extremely simple model considers a linear relationship: all the non-linearities and gain imbalances are collected by the error term,  $e(p)$ . If the estimation model is improved, the error term is expected to decrease. The calibration step here proposed is based on a point-to-point measurement approach: for each measurand a different offset is to be applied. The output of the calibration algorithm is a look-up-table mapping each raw RSSI estimate into a precise offset. In the worst case, the number of the matrix entries is equal to the total number  $M$  of points on which the calibration is made. If possible, the transcharacteristic can be modeled as a piece-wise linear function, limiting the number of points of the look-up-table. In the following, without loss of generality, a piece-wise linear transcharacteristic is assumed.



**Figure 9.2:** Piece-wise-linear transcharacteristic.

In such a direction, a more efficient and accurate compensation procedure is possible. It consists of the following steps:

1. Acquire the transcharacteristic with a chosen number of points  $M$ .
2. Subdivide the  $M$  input values into  $K$  ordered sub-set (intervals), with  $K \leq M$ , as depicted in Figure 9.2 (axis units omitted for simplicity). Each input value  $p_m$  belongs exactly to one interval  $\Delta_k$  with  $1 \leq k \leq K$ . Intervals can be easily determined analyzing the jumps in the discrete approximation of the derivative of the transcharacteristic,  $\frac{\Delta \hat{p}}{\Delta p}$ , define as follows:

$$\frac{\Delta \hat{p}(m)}{\Delta p(m)} = \frac{\hat{p}_m - \hat{p}_{m-1}}{p_m - p_{m-1}}. \quad (9.2)$$

Interval bounds are determined in  $m_k$  values, with  $m_k \in \{1, \dots, M\}$  for which  $\frac{\Delta \hat{p}(m_k)}{\Delta p(m_k)} > 1 + \epsilon$ , where  $\epsilon$  is a positive correction value to be properly tuned based on the maximum resolution of the RSSI. It is worth noting that negative jumps are not to be considered, since the transcharacteristic must be a wide-sense increasing function and negative jumps are associated to non monotonic and, hence, non invertible intervals.

3. In each interval  $\Delta_k$ , a least square fit [33] is performed based on all the samples of  $\hat{p}$  and of  $p$  belonging to that interval. The reference function  $I_k$  is a straight line, i.e.  $I_k(p) = a_k \cdot p + b_k$ , where the slope  $a_k$  is kept fixed to 1 and the offset  $b_k$  is evaluated.
4. Considering the interval  $\Delta_k$ , apply to the obtained estimation  $\hat{p}$  a proper compensation contribution  $o_k = -b_k$  leading to a corrected version of  $\hat{p}$ ,  $\hat{p}_c$ , according to the expression:

$$\hat{p}_c(p) = \hat{p}(p) + o_k = p + \underbrace{e(p) + o_k}_{e_k(p)}, \quad (9.3)$$

where  $e_k(p)$  is the overall error term.

This procedure can be viewed as an offset compensation carried out at each singular  $\Delta_k$  interval.

### 9.1.2 Correction step

The above proposed algorithm uses as input data the real power samples, i.e. the measurand, while outputs the measures according to the transcharacteristic of the RSSI. This is the case of a calibration stage, where the measurand is perfectly known.

The measurement stage, instead, faces the inverse problem. The estimates (measures) are known, and the physical quantity (measurand) stimulating the system is unknown. The role of the calibration stage is to find a unique function or map among input and output quantities (the transcharacteristic). Once known, the transcharacteristic should be inverted to determine, starting from the measurements, the “real” value. The mathematical formula is, hence, the following:

$$f(\cdot) : p = f^{-1}(\hat{p}) \quad (9.4)$$

The considered transcharacteristic is non monotonic and so non invertible (Figure 9.4). However, Figure 9.4 depicts a property of piecewise regularity that can help to overcome the highlighted problem. The compensation procedure can be divided into two steps:

1. Map the  $\Delta_k$  bounds  $p_k$  through the transcharacteristic and find  $\hat{p}_k$  (bounds in terms of estimates).
2. For each input  $\hat{p}$  find the proper  $k$ , i.e.  $\hat{p}_k \leq \hat{p} \leq \hat{p}_{k+1}$  and apply to  $\hat{p}$  the  $k$ -th offset  $o_k$ .

Obviously the non-invertible regions can not be eliminated, but the new transcharacteristic is far more adherent to the ideal identity function. The non invertibility is translated into a uncertainty in the input power.

## 9.2 Experimental verification

### 9.2.1 Adopted testbed

A sketch of the testbed adopted in the performed experiments is shown in Figure 9.3. It consists of a IEEE 802.15.4 compliant wireless node (sensing node) available from Tmote Sky and equipped with a CC2420 radio[7], providing a rough power estimate as a signed 8-bit integer.

A constant -45 offset has to be applied to convert it to dBm units. An arbitrary waveform generator, AWG, Agilent Technologies E4420B ESG (250 kHz - 6 GHz frequency range), is used to generate the test signal with power  $p$  and a trigger command for enabling the sensing node. The trigger command is a digital TTL signal that, applied via a cabled connection to the sensing node, causes an interrupt to be caught. The PC is used to control AWG, through a General Purpose Interface Bus (GPIB) connection, and to acquire the readings from the node through a USB 2.0 link. The internal antenna printed on the node board is disabled while an external SMA-type socket is soldered to the transceiver’s RF pins allowing the use of an external antenna [5].

In order to reduce power estimation uncertainty due to the radio link, the generator AWG is cable-connected to the above mentioned SMA-socket. An SMA-to-SMA-type coaxial cable with an additional N-to-SMA-type adapter are used to connect the N-type output of AWG to both the node and an accurate power meter, a Hewlett Packard (HP) 437B equipped with a HP 8482H power sensor used as reference. This power meter is used to measure the cable loss that is accounted for and compensated in the final calibration procedure. The calibration stage is performed considering a sine wave test signal with a fixed frequency  $f_0$ ; the signal power  $p$  is linearly varied in the range  $[-90, 0]$  dBm, with steps of 0.2 dB (maximum AWG resolution), for a total of  $M$  steps. Test frequency  $f_0$  is chosen in the middle of the communication channel deployed by the sensing node, i.e.  $f_0 = 2.480$  GHz (channel 26 of the IEEE 802.15.4 standard).

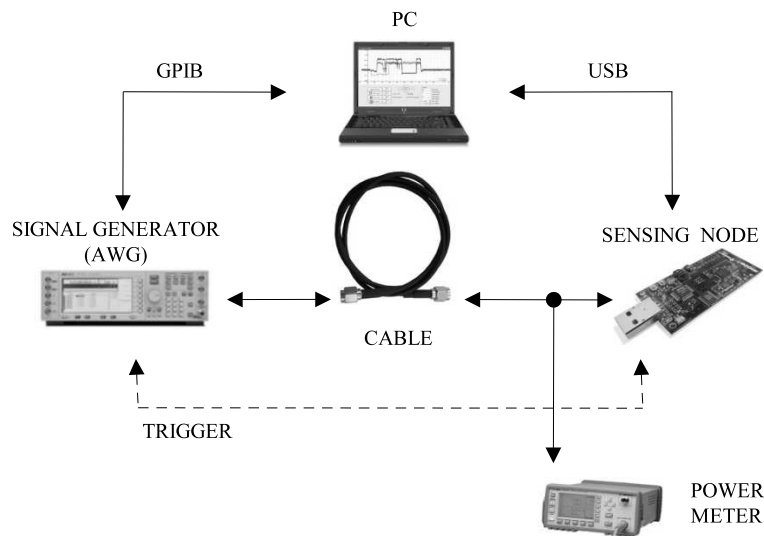


Figure 9.3: Adopted testbed.

For each of the  $M$  input power levels,  $p_m$ , with  $1 \leq m \leq M$ , the following tasks are performed:

1. the PC communicates to the AWG the actual value of  $p_m$ ;
2. the AWG generates the sine-wave and the trigger signal for the sensor node. Each configuration is held for a 100 ms interval to guarantee a stationary signal to the sensing node;
3. at the reception of the trigger, the sensor node acquires  $N = 100$  power estimates from the RSSI pin at the maximum available data rate;

4. the wireless node sends the acquired samples to the PC, and the data are displayed and stored in a  $M \times N$  size matrix  $\mathcal{R}$  for post-processing.
5. Once acquired, the matrix  $\mathcal{R}$  is processed through an automatic algorithm whose output result is a correction look-up-table to be applied to the RSSI readings.

### 9.2.2 Experimental results

A set of experimental results obtained by using the described testbed are here summarized. In Figure 9.4 the transfer characteristic  $f(\cdot)$  of the analyzed wireless RSSI device is presented. The ideal function is the identity  $\hat{p}(p) = p$ . The dark trace plots the raw estimate evaluated by the RSSI circuitry,  $\hat{p}$ , while the light trace shows the RSSI estimation after the offset compensation,  $\hat{p}_c$ , according to the algorithm described in Section 9.1. It is worth noting that the lower limit of the trace is truncated to -90 dBm: in fact, for values lower than -95 dBm, the effect of the receiver noise floor is dominant. From the uncompensated data, a few interesting results can be observed:

- Different values of  $\hat{p}$  are mapped into the same values of  $p$ . Such an effect may lead to inaccuracy in power measurements in the order of 6 dB (as can be seen in the diagram), as stated in the transceiver data-sheet [7].

This effect has a “systematic” component, because the maximum resolution of the RSSI is 1 dB, so even in the case of perfect calibration, input power  $p \in [p - \rho/2, p + \rho/2)$  (where  $\rho$  is the resolution step) are mapped into the same value. In the compensation phase this effect leads to a minimum uncertainty of  $\pm 0.5$  dB.

- Some values of the output  $\hat{p}$  can never be obtained (missing levels).
- $f(\cdot)$  is not invertible: there are values that greatly differs from the ideal curve, up to 6 dB.

A detailed view of the differences between the two transcharacteristics and the ideal curve is sketched in Figure 9.5. As can be seen, the proposed procedure significantly minimizes the errors: the corrected error is an almost zero-mean signal, with a “granular error” behavior typical of quantized input-output relationship.

A representation of the inverted traces of Figure 9.4 is shown in Figure 9.6. The diagram provides for each estimate  $\hat{p}$  the corresponding  $p = f^{-1}(\hat{p})$ .

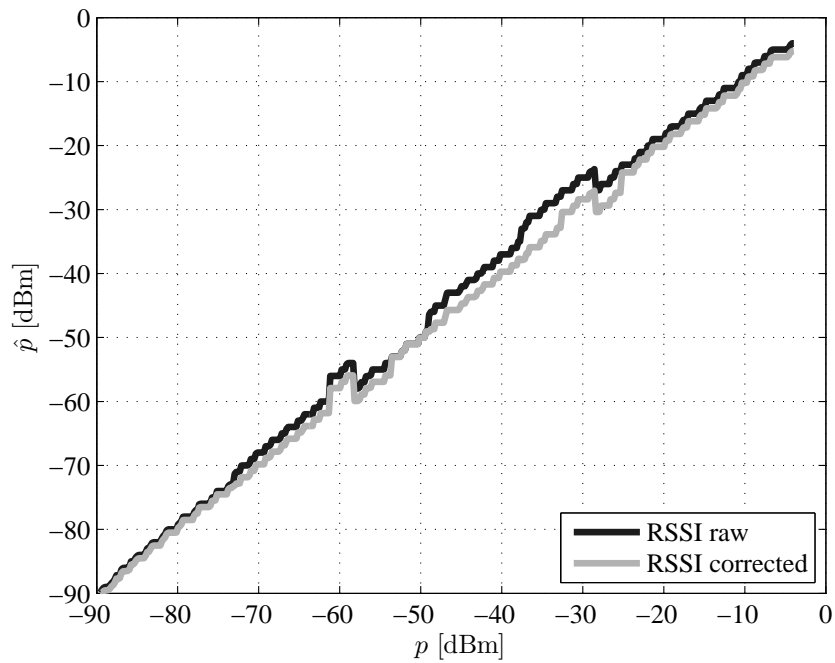


Figure 9.4:  $\hat{p}$  vs  $p$  before and after correction.

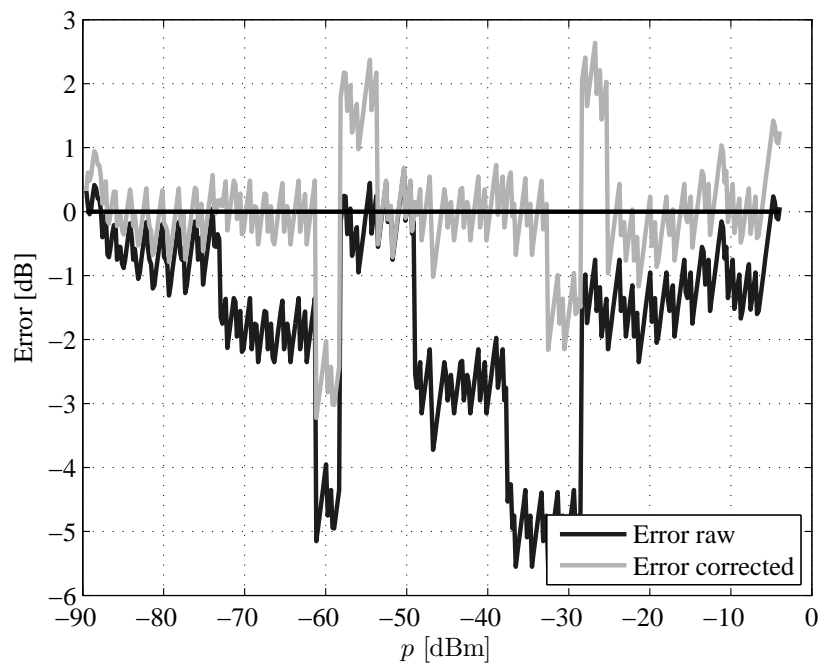


Figure 9.5: Error trend before and after correction.



As can be seen in the figure and also reported in Table 9.1, the proposed procedure greatly improves the accuracy performance of the power measurements: for example, in the range  $[-27,-24]$  dBm, the error  $e$  (5.6 dB) without the adopted compensation changes into  $e_k$  (2 dB). In the table, the values of  $o_k$  are also reported for each of the considered intervals of  $\hat{p}$ .

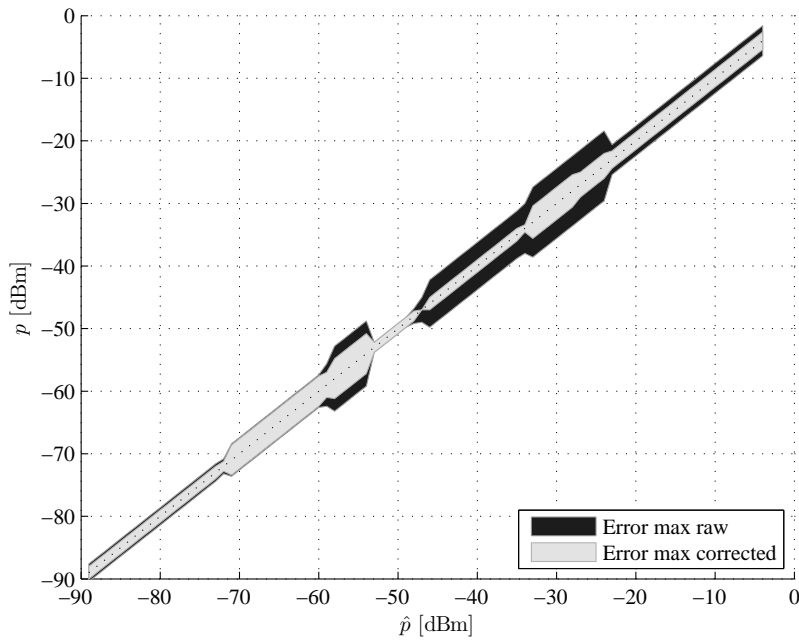


Figure 9.6:  $p$  vs  $\hat{p}$  with error bars, before and after the adopted compensation.

Table 9.1: RSSI domain partition and error correction.

$\hat{p}$ [dBm]	$o_k$ [dB]	$e$ [dB]	$e_k$ [dB]
$[-90,-73]$	-0.5	1.4	1.2
-72	-0.8	1.2	0.8
$[-71,-60]$	-1.8	2.6	2.4
-59	-0.9	3.4	2
$[-58,-54]$	-1.9	5.2	3.2
$[-53,-49]$	0	0.8	0.8
-48	-0.7	1.2	0.8
-47	-1.7	2	0
$[-46,-35]$	-2.7	3.8	1
-34	-3.9	4	0.6
$[-33,-28]$	-4.9	5.6	2.6
$[-27,-24]$	-3.4	5.6	2
$[-23,-4]$	-1.2	2.4	1.4

### 9.3 Conclusions

In this chapter a low cost wireless sensor is considered. A theoretical analysis of non-idealities in RSSI transcharacteristic is provided and a specific case study is analyzed. A simple compact RF transceiver operating in the IEEE 802.15.4 standard is used to perform power evaluations. The raw collected data are processed with a simple and powerful algorithm in order to reduce the measurement uncertainty.

The proposed approach is more accurate with respect to methods commonly performed in practice, that are commonly based on a rough offset compensation based on a mean fit over the entire transfer characteristic. Furthermore, the proposed approach is based on a simple algorithm with low computational complexity: in fact many steps are weighted summations or matrix computations.

The presented data acquisition procedure is a fully automatic and general approach. Once changed the sensing module, this test-bed can be used to acquire the transcharacteristic of a generic RSSI chip. Only minor modifications to the system software are needed, such as the adaptation of the power and frequency ranges or the change of polarity of the trigger signal. The calibration procedure is, instead, dependent on the kind of impairments affecting the transcharacteristic.

The obtained results suggest that this approach could be an effective and cheap solution for radio frequency power measurements purposes, as previously introduced, e.g. EM pollution monitoring or localization, along with proper algorithm to be implemented. This could be an effective starting point for the creation of low-cost measuring systems exploiting “side effects” of commonly used transceivers. The RSSI-equipped nodes have proven to be a viable technology for large scale measurement scenarios.

### 9.4 Multiple input improvement

The above described power measurement method assumes that only a sample of RSSI is available for the input power estimation. This leads to a complicated algorithm that is not able to resolve uncertainty due to non monotonicity of the transcharacteristic.

In the following section a completely different approach exploiting a multiple input estimation is analyzed. This is a general approach (as the single input one) but it will be tailored for a power measurement application. In this section only a theoretical analysis and first simulation results are provided.

A testbed is, at the moment, under development. The practical results will be published as an extension of [39].

### 9.4.1 Proposed approach

In Figure 9.7 a sketch of the rationale behind this approach is depicted. Given  $p$  the input power and  $\hat{p}$  the output power (RSSI), the transcharacteristic can considerably differ from the ideal straight line.

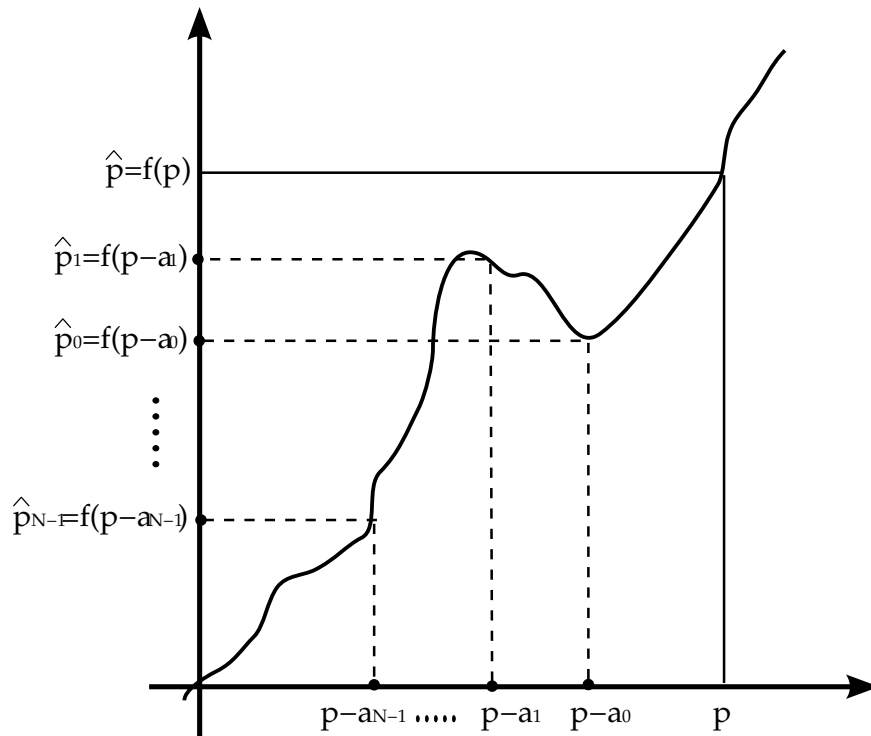


Figure 9.7: Multiple input approach.

In the following a logarithmic scale is assumed, i.e. both  $p$  and  $\hat{p}$  are measured in dBm unit.

As for the single input approach, the first step is the acquisition of the transcharacteristic with a given resolution. For the sake of simplicity in the following a 1 dB resolution is used. This data can be stored in a simple matrix associating for each input power an output power.

The measurement phase is heavily modified if compared to the original single input one. The core device is a variable attenuator to be inserted between the antenna connector and the RSSI chip. This attenuator must have an external control signal (analog or digital), allowing an external device to set a well defined attenuation. A  $N$ -tuple of  $N$  attenuations, namely  $[a_0, a_1, \dots, a_{N-1}]$ , is chosen and remains fixed for the whole algorithm.

In this first draft of algorithm attenuations are not optimized, but chosen on a rule-of-thumb basis: at least one attenuated version of the measurand must fall outside a non-invertible region of the transcharacteristic. An important future topic could be: once given the transcharacteristic, find the optimal N and the optimal attenuation set.

The algorithm runs in three steps:

1. **Transcharacteristic mapping:** this step is a calibration phase, done once with a well known power source (the AWG in Figure 9.3). This step is used to create a look-up-table storing function  $f : \hat{p} = f(p)$ .
2. **Measurement phase:** this step consist of measuring N attenuated version of the unknown input power  $p$  (using in order all the attenuations  $\mathbf{a} = [a_0, a_1, \dots, a_{N-1}]$ ). The output is the N-tuple  $\bar{\mathbf{p}} = [\bar{p}_0, \bar{p}_1, \dots, \bar{p}_{N-1}]$ . The output is  $\bar{\mathbf{p}} = f(p^* - \mathbf{a})$ , or:

$$\begin{bmatrix} \bar{p}_0 \\ \bar{p}_1 \\ \vdots \\ \bar{p}_{N-1} \end{bmatrix} = \begin{bmatrix} f(p^* - a_0) \\ f(p^* - a_1) \\ \vdots \\ f(p^* - a_{N-1}) \end{bmatrix} \quad (9.5)$$

With  $p^*$  the unknown input power to be estimated.

3. **Estimation phase:** this step estimates the most probable input power given the N-tuple  $\bar{\mathbf{p}}$ . This phase uses the N-tuple and the stored transcharacteristic to find a cost function upon the varying of the input power. The cost function  $\lambda(p)$  is calculated as described in the following formula:

$$\lambda(p) = \|\bar{\mathbf{p}} - f(p - \mathbf{a})\| = \sqrt{\sum_{i=0}^{N-1} |\bar{p}_i - f(p - a_i)|^2} \quad (9.6)$$

This approach calculates the “distance” of the measurement vector from a sliding vector taken from the transcharacteristic. The algorithm works if the cost function shows a global minimum. The most probable input power is the inverse image of the minimum:

$$p^* = \underset{p}{\operatorname{argmin}} \lambda(p) = \{p^* \mid \forall p : \lambda(p^*) \leq \lambda(p)\} \quad (9.7)$$

This approach spans all the transcharacteristic. A modified version can limit the distance estimation to the most probable points in the neighbourhood of the zero attenuated value.

### 9.4.2 Simulation results

In this section a Matlab simulation is presented to prove the effectiveness of the proposed approach. In Figures 9.8(a) and 9.8(b) a comparison of a single input and a multiple input approach is presented. The same transcharacteristic with a non-monotonic behavior is depicted.

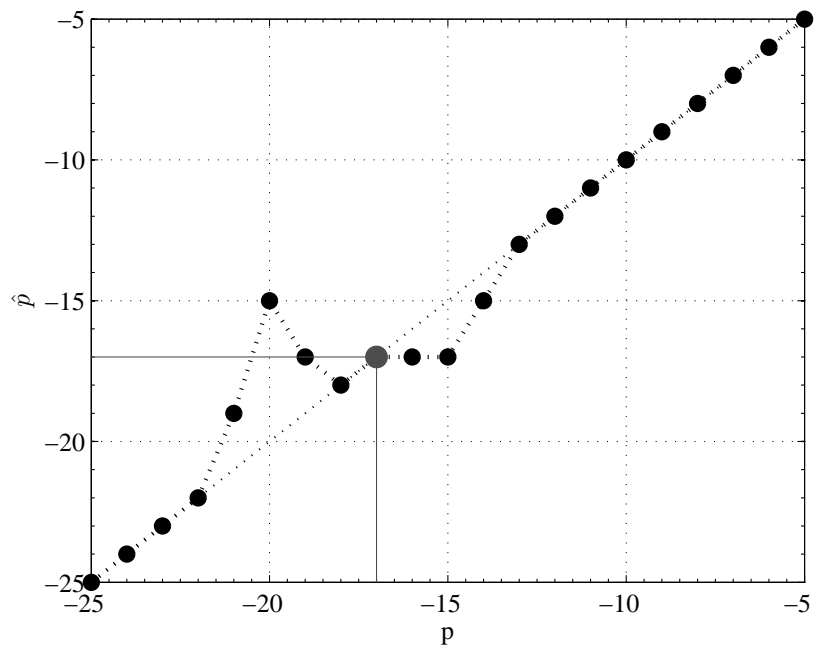
For example, in the single input approach a -17 dBm input is mapped into a -17 dBm value. In the multiple input approach a attenuation vector  $\mathbf{a} = [0, 2, 4]$  is used, so the output vector is  $\bar{\mathbf{p}} = [-17, -17, -19]$ . As clearly visible, the -17 dBm output can not be univocally associated to an input power, since there are four values (-15,-16,-17,-19) that are mapped into -17.

In Figures 9.9(a) and 9.9(b) the proposed cost function  $\lambda$  is used and it is plotted in logarithmic scale to catch the great dynamic range of values (minimum saturated to -20 dB). In the single input approach the function has no a single minimum, since the above described four values lead to the same value of the cost function. The multiple input approach allows a simpler detection since there is only one minimum.

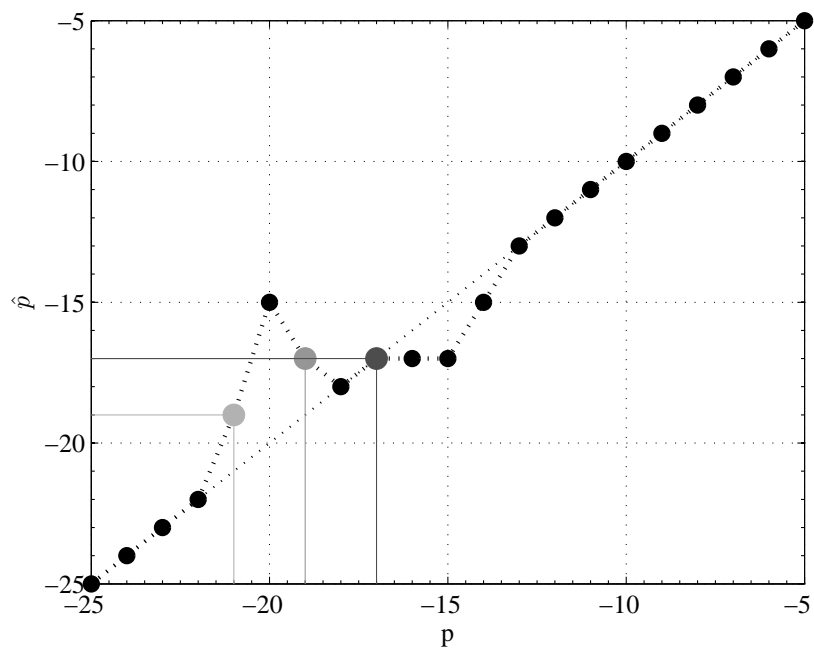
### 9.4.3 Testbed

In the practical implementation several controllable attenuators exist. An example of low cost (about ten dollars at the moment), integrated digital attenuator is the Hittite HMC273MS10G. It allows a 1 dB attenuation step, attenuation range from 1 to 31 dB, a wide span of frequency (0.7 to 3.8 GHz) and switching time less than 600 ns. A block diagram is depicted in Figure 9.10. Each digital input controls an inner attenuator: the combination of this 5 attenuators can span all the range of attenuations.

The above described procedure can be simply implemented with a minor modification to the testbed depicted in Figure 9.3. In Figure 9.11 the attenuator is simply inserted between the cable and the RF input connector of the sensing node. The chosen attenuator can be controlled with 5 digital input lines. The sensing node can hence use a digital output and a serial to parallel converter to drive the attenuator. An alternative approach (not needing any clock) is to generate an analog voltage with the node and then convert it with an ADC and take 5 bits to drive the attenuator. The major limits to this multiple input approach in a practical implementation are the switching time of the attenuator and the sampling time of the sensing node. For each sample a N-tuple of N attenuation are to be collected in a serial fashion. A fast varying RF signal can hence not be correctly measured for all the N attenuations causing the algorithm failure.

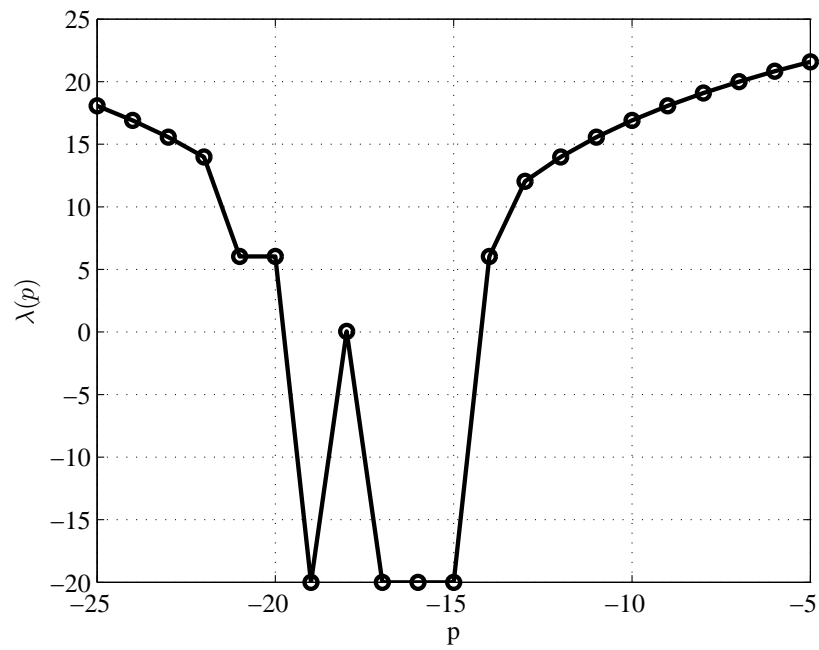


(a)

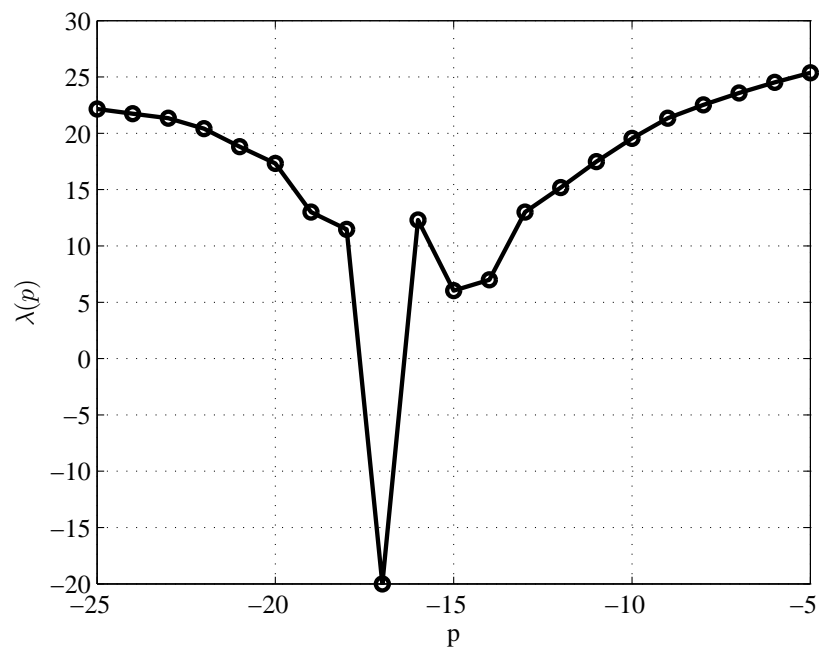


(b)

**Figure 9.8:** Example of a transcharacteristic: (a) single input approach, (b) multiple input approach.



(a)



(b)

**Figure 9.9:** Cost function,  $\lambda(p)$  (dB scale used) for: (a) single input estimation, (b) multiple input estimation.

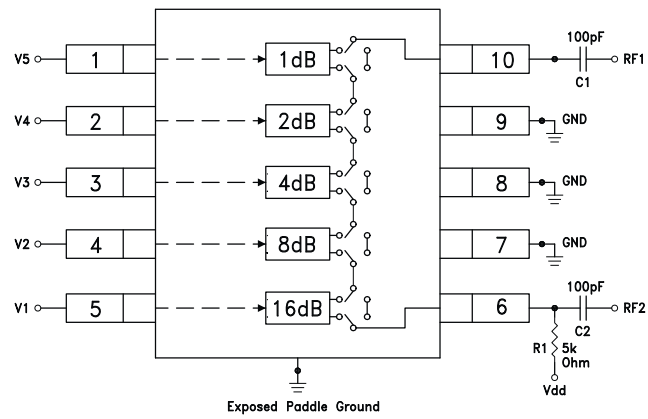


Figure 9.10: Blocks diagram of a typical digital controlled attenuator.

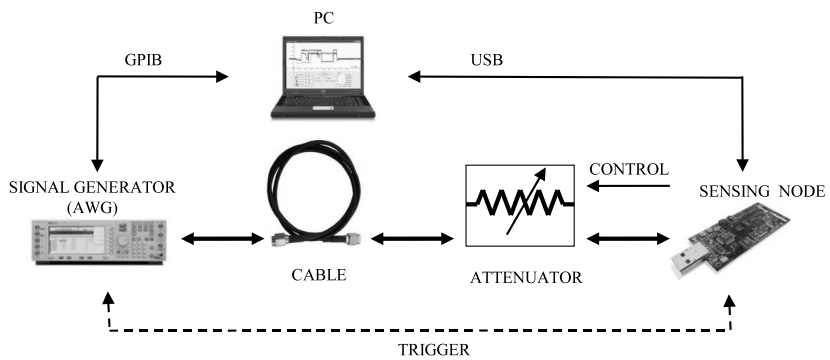



Figure 9.11: Modified testbed for multiple input estimation.



## Conclusions

 SCIENTIFIC research, as a work of art, seldom follows a straight line. The research subject sketched at the beginning of this work has been mainly a magnetic compass during this journey: helpful to see a route but not sufficient to prevent the author from wandering around in a multi-dimensional space, where neither the bounds nor the number of dimensions are known yet.

This thesis is a *summa* of a three-year research period. Probably it is not such an homogeneous work as it should be; it is a collection of different perspectives on the leading idea of investigating the effects of interference on wireless systems' performance.

The main intent of the author was to underline the original results obtained during these years: not all the publications produced have been used for this thesis and several works have been summed up. The chapters are wrote, with minor modification, in chronological order.

After few chapter (Chapter 1, 2, 3 and 4) introducing the "matter" and giving the needed background to understand the next part, the thesis faces the problem of implementing and testing a WSN-based industrial control system.

Chapter 5 and 6 show that such a system is feasible with good performance. Not all the important matters are analyzed: for example power consumption is neglected. The main focus is on how interference signals can impair such a wireless system and how the network reliability can be improved. Experimental data are interpreted according to a theoretical analysis of the performance indices and an in-depth experimental analysis helps the reader to understand how to face even real interference issues. The rationale behind this part is that CSMA/CA based systems are very prone to interference impairments and often the best choice is to disable this mechanism at all.

In Chapter 7 a cross layer modeling of interference effects on CSMA/CA based systems is extensively presented, with a theoretical formulation, a simulation approach and finally a set of experimental data.

Chapters 8 and 9 face the problem of radio frequency power estimation. The RSSI circuitry embedded in many wireless devices is presented, describing some background information and evaluating the effects of interference on such a subsystem both with simulations and experimental data. After considering the effects of interference in terms of erroneous power readings and channel assessment, the feasibility of a cheap integrated RF power meter is taken into account. Theoretical problems and practical solutions are presented. The chapter proves that even starting from low quality power meters, the accuracy can be greatly improved through a pre-conditioning and post-elaboration stage.

In conclusion, effects of interference have been thoroughly analyzed especially with an experimental approach, using simulations only joint with real testbed implementations. In the author's opinion, this is the only way interference can be recognized and, hence, properly faced.

# Appendix A

## *Wireless System for ElectroMagnetic Area Notice*<sup>1</sup>

**W**IRELESS communications are nowadays gaining a huge market especially in a metropolitan and sub-urban scenario. The “wireless choice” is, on one side, dictated by the flexibility and the mobility that only a wireless device can provide (e.g. a cellular network), and, on the other side, it is much cheaper than a cabled solution. A typical example in this sense is broadband supply: a common DSL solution is almost the best choice from a performance point of view, however it has severe cabling-costs and it is not a viable solution especially in mountain or isolated villages. To this aim, wireless results are being applied also in metropolitan area network with WiFi and recently WiMax-based solutions.

At present day, the most important drawback involving wireless communications is not a technological inferiority, but a social issue. Wireless system deployment always needs an infrastructure, that is, a set of antennas displaced around the served area: this can inoculate a sense of threat into users of the same services, because it is not still known if there could be a potential health risk deriving from the prolonged exposure to electromagnetic fields.

Moreover, European and Italian legislation (see 2004/40/CE or d.lgs. 81/08) is coping with this problem, trying to lower the EM impact, in order to guarantee the health of workers in the working site. In this context, the environment can be very hostile, due to the use of factory machinery equipped with high power EM sources (e.g. plastic and metal soldering or fusion).

To this aim, classical form of prevention are dosimeters, notifying if a worker enters in a dangerous zone, or environmental analysis, performed with both broadband (i.e. integral power PMM ) and narrowband instrumentation (i.e. spectrum analyzers).

---

<sup>1</sup>This project (WISE-MAN) participated to the grant “M31 Grant Intelligenza Coraggiosa-2008”

All these approaches are very expensive, because instrumentation costs several thousand Euros (typical 10-50 k€ for a commercial spectrum analyzer), it must be very severely calibrated and, above all, the user must be technically prepared to use it.

Modern digital wireless systems typically provide an estimation of the received signal power, and use it for “internal” purposes, i.e. to increase battery duration, to limit interference among different devices (e.g. in a cellular network) or to lower collision probability (e.g. in WiFi systems). This information, properly elaborated, can be exploited to provide an easy-to-use, cheap and ubiquitous measure of the electromagnetic impact due to the most common wireless sources in a metropolitan or sub-urban area.

This solution has a threefold impact: first, the basic idea should be a hot-topic for academic research, melting different branches of communication engineering, such as measurement, antenna, network and computer engineering. Second, once proved the real feasibility and reliability of such a solution, that is after the concrete realization end test of a prototype, this project may easily become a commercial product, a “killer” application that can be tailored to different costumers. First of all, it will be a cheap hand-held monitoring tool employed by professional engineers in preliminary investigation stages (e.g. services companies or regional protection and prevention agencies); furthermore, it should be purchased and easily used by anyone who wants to keep his house or working site monitored. At last, given such widespread applications, such a product directly aims the above mentioned social fear of electromagnetic fields, providing a everyday tool to monitor everyone’s environment.

## A.1 Proposed Approach

As above suggested, the project is based on well known technologies. Almost any digital transceiver embeds a chip that performs RSSI. This metric, usually carried out in the digital domain, or through analogical circuits in older chips, is used to improve the quality of communication. The idea beneath this project is to use a digital receiver (i.e. cellular, WiFi or ZigBee), whose typical cost rounds on a few dollars, and employ the RSSI feature embedded in to scan a well defined portion of the electromagnetic spectrum. A chip compliant to a certain standard can obviously analyze the channel of that standard only. For this reason, the target is to assemble different chips, each one able to estimate spectral power limited in defined intervals only, and fuse the data each one provide.

The basic assumption is that the major portion of electromagnetic emissions is imputable to widely used wireless systems, such as cellulars, WiFi or WiMax (for sure in immediate future); this is mainly true in the RF range. Each system can measure the EM field used by itself in the transmitting stage, with a fixed degree of precision and accuracy.

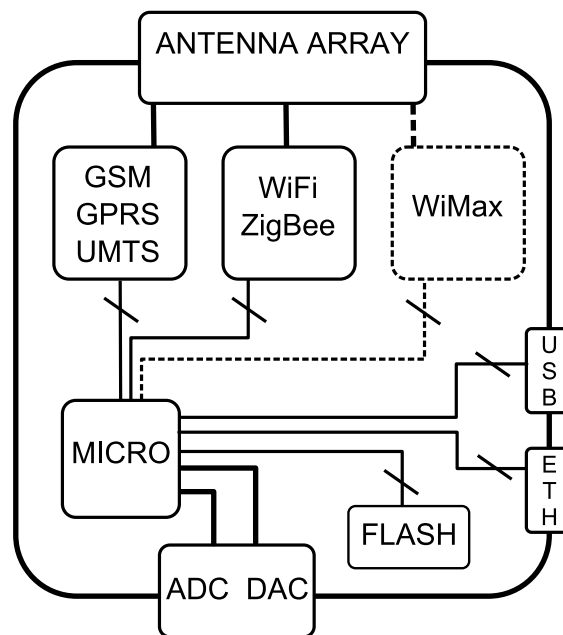


Figure A.1: Logical layout of the proposed system.

The proposed approach, as sketched in Figure A.1, employs each transceiver mounted on the board as a sensing element: obviously these chips are far under-exploited, because the RSSI feature only is used. This is a minor problem because the cost of such devices is very low. The proposed solution relies on several RF-sensitive chips (cellular, Wi-Fi, WiMax), coupled with an antenna set, where different-band elements are connected in a proper manner. Chips receive commands and send RSSI values on digital physical buses, to a low-power microcontroller. The board is also equipped with some flash memory, for data logging or calibration parameters storing. Moreover several I/O ports, ADC and DAC are connected to the proper pins of the microcontroller: they may be used to connect external actuators or sensors (temperature, light, and so on). Collected data must be driven by a PC: to this aim local interfaces such as USB and ETHERNET can be used. Remote data transfer can exploit the radio chips: if the measure phase is correctly multiplexed with a transmitting stage, the device can send data on a cellular link or on a WiFi network without interfering with the environment under test.

Classical approaches, such as broadband and narrowband power meters, are very expensive and difficult to integrate. The proposed approach tries to make the RSSI more accurate and precise with a calibration phase, and uses high quality measurements data (already available in today's radio devices) to convey an overall electromagnetic field monitoring. This approach is very cheap, i.e. the target price could be 200-300 €. A simple example of a RSSI reading, obtained with the radio chip TI CC2420 used by Tmote Sky sensor nodes, is reported in Figure 3.7: it is worth noting that such a system (about 90€), even without calibration can obtain a channel power estimation with a very good accuracy, even comparable to classical method results.

EM field monitoring in the low frequency domain is not easily suitable for this approach. It should be employed a purposely developed "low frequency receiver", capable to amplify and sample "slow" signals, and to evaluate RSSI. Nevertheless, it is important to state that fields in the Extremely Low Frequency range, such as the ones generated by 50-60 Hz power lines, are difficult to be estimated, because electric and magnetic fields must be measured separately (near-field systems). For such specific application, the proposed approach needs deep enhancement and modifications. Hence, in industrial environment, RF telecommunication signals only can be detected by the proposed system. Low frequency or very high frequency fields should be investigated with classical instrumentation or, better, with a modified version of the EM monitor proposed (to be studied).

## A.2 Implementation

Since this project idea has grown out from the team's research activities, several key steps have been already done. In particular, part of the background research to verify the feasibility to check the availability of engineered commercial products, has been part of prior works.

The project is composed of two sub-parts: the first target is the realization of a compact hardware platform that performs field measurement. This device can be used stand alone or in a networked environment, in order to allow a central monitoring system to collect data and react with the proper decision (in a manned or unmanned way). Hence, the second part of the project aims to the implementation of the described collecting network, together with the central monitoring system. The time frame envisioned for this project is the following: after the first year a prototype of the single device above described will be operating, at least with a reduced subset of features (it is worth to note that physical PCB design and production will last quite long).

## Bibliography

- [1] "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, June 12 2007.
- [2] "Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–305, 2006.
- [3] G. Mulligan, "The 6lowpan architecture," in *EmNets '07: Proceedings of the 4th workshop on Embedded networked sensors*, (New York, NY, USA), pp. 78–82, ACM, 2007.
- [4] A. Goldsmith, *Wireless Communication*. Cambridge University Press, 2005.
- [5] "Tmote Sky Datasheet." Moteiv. <http://www.moteiv.com>.
- [6] "MSP430x1xx Family User's Guide." TI Product data sheet. <http://ti.com/msp430>.
- [7] "CC2420 2.4 GHz IEEE 802.15.4 Zigbee RF Transceiver." TI Product data sheet. <http://focus.ti.com/docs/prod/folders/print/cc2420.html>.
- [8] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *ASPLOS-IX: Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*, (New York, NY, USA), pp. 93–104, ACM, 2000.
- [9] F. De Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella, "On the use of wireless networks at low level of factory automation systems," *Industrial Informatics, IEEE Transactions on*, vol. 2, pp. 129–143, May 2006.
- [10] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. Mc-Graw Hill, 3rd ed., 1991.
- [11] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," *Proceedings of the IEEE*, vol. 93, June 2005.

- [12] A. Willig, M. Kubisch, C. Hoene, and A. Wolisz, "Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer," *Industrial Electronics, IEEE Transactions on*, vol. 49, pp. 1265–1282, Dec 2002.
- [13] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, pp. 535–547, Mar 2000.
- [14] A. Flammini, D. Marioli, E. Sisinni, A. Taroni, and M. Pezzotti, "A wireless thermocouples network for temperature control in plastic machinery," in *Factory Communication Systems, 2006 IEEE International Workshop on*, pp. 219–222, 27, 2006.
- [15] A. Koubaa, M. Alves, and E. Tovar, "A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks," *Factory Communication Systems, 2006 IEEE International Workshop on*, pp. 183–192, June 27, 2006.
- [16] M. Bertocco, G. Gamba, A. Sona, and S. Vitturi, "Performance measurements of CSMA/CA-based wireless sensor networks for industrial applications," *Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE*, pp. 1–6, 1–3 May 2007.
- [17] M. Bertocco, G. Gamba, A. Sona, and S. Vitturi, "Experimental characterization of wireless sensor networks for industrial applications," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, pp. 1537–1546, Aug. 2008.
- [18] GUM, *Guide to the Expression of Uncertainty in Measurement*. Geneva: International Organisation for Standardisation, 2nd ed., 1995.
- [19] "Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)," *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–580, 2005.
- [20] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 419–420, ACM, 2006.
- [21] I. Howitt, "WLAN and WPAN coexistence in UL band," *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 4, pp. 1114–1124, Jul 2001.
- [22] J.-P. Thomesse, "Fieldbus technology in industrial automation," *Proceedings of the IEEE*, vol. 93, pp. 1073–1101, June 2005.
- [23] M. Bertocco, G. Gamba, and A. Sona, "Is CSMA/CA really efficient against interference in a wireless control system? an experimental answer," in *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pp. 885–892, Sept. 2008.



- [24] J. Howitt, I. Gutierrez, "IEEE 802.15.4 low rate - wireless personal area network coexistence issues," *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1481–1486 vol.3, 16-20 March 2003.
- [25] S. Y. Shin, H. S. Park, and W. H. Kwon, "Mutual interference analysis of ieee 802.15.4 and ieee 802.11b," *Comput. Networks*, vol. 51, no. 12, pp. 3338–3353, 2007.
- [26] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Laella, "Performance study of ieee 802.15.4 using measurements and simulations," *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, vol. 1, pp. 487–492, 2006.
- [27] N. Golmie, R. E. V. Dyck, A. Soltanian, A. Tonnerre, and O. Rébala, "Interference evaluation of bluetooth and ieee 802.11b systems," *Wirel. Netw.*, vol. 9, no. 3, pp. 201–211, 2003.
- [28] M. Bertocco, G. Gamba, and A. Sona, "Experimental optimization of cca thresholds in wireless sensor networks in the presence of interference," *Workshop on Electromagnetic Compatibility Conference Proceedings, EMC Europe 2007 IEEE*, 14-15 June 2007.
- [29] L. Angrisani, M. Bertocco, G. Gamba, and A. Sona, "Modeling the performance of CSMA-CA based wireless networks versus interference level," *Instrumentation and Measurement Technology Conference Proceedings, 2008 IEEE International*, pp. 376–381, May 2008.
- [30] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 517–526, 4-7 Oct. 2004.
- [31] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental analysis of concurrent packet transmissions in wireless sensor networks," in *Proceedings of the Fourth ACM SenSys Conference*, (Boulder, Colorado, USA), pp. 237–249, ACM, November 2006.
- [32] M. Zuniga and B. Krishnamachari, "An analysis of unreliability and asymmetry in low-power wireless links," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, p. 7, 2007.
- [33] G. Seber and C. Wild, *Nonlinear Regression*. John Wiley & Sons, 2005.
- [34] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. New York: Dover, 1965.
- [35] S. Karlin and H. Taylor, *An Introduction to Stochastic Modeling, Third Edition*. Academic Press, 1998.
- [36] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Assessing coexistence problems of IEEE 802.11b and IEEE 802.15.4 wireless networks through cross-layer measurements," in *Instrumentation and Measurement Technology Conference Proceedings, 2007 IEEE*, (Warsaw, Poland), pp. 1–6, May 2007.

- [37] M. Bertocco, G. Gamba, and A. Sona, "Assessment of out-of-channel interference effects on IEEE 802.15.4 wireless sensor networks," in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*, pp. 1712–1717, May 2008.
- [38] L. Angrisani, M. Bertocco, G. Gamba, and A. Sona, "Effects of RSSI impairments on IEEE 802.15.4 wireless devices performance susceptibility to interference," *Proc. of IEEE International Symposium on Electromagnetic Compatibility, EMC Europe 2008*, Sept. 8-12, 2008.
- [39] L. Benetazzo, M. Bertocco, A. Chiara, G. Gamba, A. Sona, and M. Ponzin, "Enhanced use of rssi-based wireless network nodes for power measurement purposes," in *Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE*, pp. 1037–1042, May 2009.
- [40] S. Mirabbasi and K. Martin, "Classical and modern receiver architectures," *Communications Magazine, IEEE*, vol. 38, no. 11, pp. 132–139, Nov 2000.
- [41] P.-I. Mak, S.-P. U, and R. P. Martins, "Transceiver architecture selection: Review, state-of-the-art survey and case study," *Circuits and Systems Magazine, IEEE*, vol. 7, no. 2, pp. 6–25, 2007.
- [42] J. Proakis, *Digital Communications*. McGraw-Hill Science Engineering Math, August 2000.