Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

Industrial Wireless Sensor Networks

Simulation and measurement in an interfering environment

Ph.D. candidate
Federico Tramarin

Advisor
Prof. Matteo Bertocco

Ph.D. School in
Information Engineering

Ph.D. School Head
Prof. Matteo Bertocco

XXIV cycle
2009-2011

To
Anna and Agata

Contents

List of Figures

List of Tables

ACRONYMS

AARF	Adaptive ARF
ACK	Acknowledgment
AES	Advanced Encryption Standard
AIFS	Arbitration IFS
AP	Access Point
API	Application Program interface
AQR	Adaptive QR
ARF	Automatic Rate Fallback
ARQ	Automatic Retransmission Query
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BIR	Bounded Immediate Retransmission
BPSK	Binary Phase Shift-Keying
BSS	Basic Service Set
BSSID	Basic Service Set IDentification
CAP	Contention Access Period
CCA	Clear Channel Assessment
CCK	Complementary Code Keying
CDF	Cumulative Distribution Function

CF	Coordination Function
CFP	Contention Free Period
CP	Contention Period
CPU	Central Processing Unit
CRC	Cyclic Redundancy Code
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
CTS	Clear To Send
CW	Contention Window
DAC	Digital to Analog Converter
DBPSK	Differential Binary Phase Shift Keying
DCF	Distributed Coordination Function
DIFS	Distributed Coordination Function Distributed Coordination Function DIFS (DIFS)
DQPSK	Differential Quadrature Phase Shift Keying
DS	Distribution System
DSM	Distribution System Medium
DSS	Distribution System Services
DSSS	Direct–Sequence Spread Spectrum
DUT	Device Under Test
ED	Energy Detection
EDCA	Enhanced Distributed Channel Access
EIFS	Extended IFS
EIRP	Equivalent Isotropically Radiated Power
EPDF	Empirical Probability Density Function
ERP	Extended-Rate PHY
ERP–OFDM	Extended-Rate PHY–Orthogonal Frequency Division Multiplexing

ESS	Extended Service Set
EVM	Error Vector Magnitude
FARF	Fast rate reduction ARF
FCC	Federal Communication Commission
FCF	Frame Control Field
FCS	Frame Check Sequence
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FFD	Full–Function Device
FHSS	Frequency Hopping Spread Spectrum
FIFO	First In–First Out
FN	Fragment Number
FPGA	Field Programmable Gate Array
GFSK	Gaussian shaped Frequency Shift Keying
GTS	Guaranteed Time Slots
HCCA	Hybrid Coordination Function (CF) (HCF) Controlled Channel Access
HCF	Hybrid CF
HR	High Rate
HR/DSSS	HR/DSSS
HR/DSSS/PBCC	HR/DSSS (HR/DSSS)/Packet Binary Convolutional Coding (PBCC)
IBSS	Independent Basic Service Set (BSS)
IEEE	Institute of Electrical & Electronics Engineers
IFFT	Inverse Fast-Fourier Transform
IFS	Inter Frame Space
IP	Internet Protocol
IR	Immediate Retransmission

ISM	Industrial, Scientific and Medical
ISO	International Standard Organization
ISO/OSI	ISO/Open System Interchange
ITU–T	International Telecommunication Union – Telecommunication Standardization Sector
IWSN	Industrial Wireless Sensor Network
LAN	Local Area Network
LLC	Link Layer Control
LOS	Line of Sight
LQI	Link Quality Indication
LR–WPAN	Low-Rate Wireless Personal Area Network
LSB	Least Significant Bit
MAC	Medium Access Control
MHR	MAC Header
MIMO	Multiple Input–Multiple Output
MLME	MAC Layer Management Entity
MMPDU	MAC Management Protocol Data Unit
MPDU	MAC PDU
MSDU	MAC Service Data Unit
NAV	Network Allocation Vector
NET	Network Layer
Nic	Network Interface Card
OFDM	Orthogonal Frequency Division Multiplexing
O–QPSK	Offset Quadrature Phase Shift-Keying (PSK) (QPSK)
OS	Operating System
PAN	Personal Area Network
PBCC	Packet Binary Convolutional Coding

PC	Point Coordinator
PCF	Point Coordination Function
PDU	Protocol Data Unit
PER	Packet Error Rate
PHR	Physical Header
PHY	Physical Layer
PIFS	Point Coordination Function IFS
PLCP	Physical Layer Convergence Procedure
PLME	Physical Layer Management Entity
PLR	Packet Loss Rate
PMD	Physical Media Dependent
PN	Pseudo–Noise
PPDU	Packet Protocol Data Unit
PSDU	Packet Service Data Unit
PSK	Phase Shift-Keying
PTP	Precision Time Protocol
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature PSK
QR	Queued Retransmission
RA	Rate Adaptation
RFD	Reduced–Function Device
RT	Real–time
RTS	Request To Send
RSSI	Received Strength Signal Indication
RTE	Real–Time Ethernet

SARF	Static retransmission rate ARF
SDU	Service Data Unit
SFD	Synchronization Frame Delimiter
SHR	Synchronization Header
SIFS	Short IFS
SINR	Signal To Interference Noise Ratio
SN	Sequence Number
SNR	Signal to Noise Ratio
SS	Station Services
STA	Station
TCP	Transport Control Protocol
TCP/IP	Transport Control Protocol/Internet Protocol
TDMA	Time Division Multiple Access
TSC	Time–Stamp Counter
TXOP	Transmit Opportunity
UIR	Unbounded IR
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

ABSTRACT

Recently the research community is considering with a growing interest the adoption of Industrial Wireless Sensor Networks (IWSNs) in application contexts such as real–time (industrial) communications and distributed measurement systems. These types of applications typically impose very tight requirements to the underlying communication systems and, moreover, they might have to cope with the intrinsic unreliability of wireless networks. It is hence needed an accurate characterization of these networks' behavior, from a metrological point of view. Suitable measurement systems have to be realized, and experiments performed aimed at evaluating some of the most appropriate performance indicators.

Unfortunately, despite the appealing opportunities provided by IWSNs, their adoption is just at its beginning. It is clear that a comprehensive experimental analysis of their behavior would improve theoretical analysis, simulations and design of the network, since the consequent increased accuracy of models could reduce the source of difference between real and expected behaviors.

With the work presented in this thesis the author would provide some original contribution in the field of measurements on real–time wireless networks adopted for industrial communications and distributed measurement systems.

In this context, one of the most relevant aspect to be considered is represented, as described in the literature, by interference that possibly arises from "intentional" communications taking place in external systems. In order to address such an issue, some simulation techniques have been considered. As a result, they lead to the development of a network simulator software tool that enabled a cross–layer analysis of interference.

This activity stimulated an in-depth study of the IEEE 802.15.4 and IEEE 802.11 communication protocols. Particularly, medium access techniques have been analyzed in the perspective of IWSN applications. On this basis new and effective methods for increasing the network reliability have been proposed, along with fair packet retransmission scheduling methods. Moreover, new rate adaptation algorithms for wireless networks specifically designed for real–time communication purposes, exploiting the high robustness of low transmission rates have been proposed.

Finally, since the reliability of a network strongly depends on the real behavior of the employed devices, an experimental approach for the measurement of the devices characteristics is presented, with the aim of providing suitable models and methods for designers.

SOMMARIO

La comunità scientifica, recentemente, sta considerando con sempre maggiore interesse l'adozione di reti di sensori wireless in contesti come le comunicazioni real–time (industriali) e i sistemi di misura distribuiti. Queste applicazioni richiedono tipicamente, al sistema di comunicazione, di soddisfare requisiti molto stringenti, considerando anche l'intrinseca inaffidabilità del canale radio. Risulta quindi necessaria un'accurata caratterizzazione, in termini metrologici, del comportamento di questa tipologia di reti, tramite sistemi di misura adatti alla valutazione dei più appropriati indici di prestazioni.

Sfortunatemente, infatti, l'impiego di questi sistemi è ancora agli inizi, nonostante le interessanti prospettive applicative fornite dalle reti wireless real–time. Appare quindi chiaro come un'accurata caratterizzazione sperimentale del loro comportamento reale migliorerebbe sensibilmente l'efficacia delle analisi teoriche, delle simulazioni e di conseguenza del progetto della rete, risultando incrementata l'accuratezza dei modelli teorici e limitate le sorgenti di deviazione tra i risultati attesi e quelli sperimentali.

Con il lavoro presentato in questa tesi, l'autore intende fornire contributi originali nel campo delle misure sulle reti wireless real–time adottate per comunicazioni industriali e sistemi di misura distribuiti.

In questo contesto, uno dei principali aspetti da considerare, come si evince dalla letteratura di compatibilità elettromagnetica, è dato dall'interferenza dovuta a comunicazioni "intenzionali" da parte di sistemi esterni. Per affrontare quest'analisi si sono inizialmente valutate alcune tecniche di simulazione. Questo ha portato allo sviluppo di un software di simulazione per reti di comunicazione specificamente progettato per l'analisi cross–layer dei fenomeni d'interferenza.

Quest'attività ha stimolato uno studio approfondito dei protocolli di comunicazione IEEE 802.15.4 and IEEE 802.11. Nell'ottica del loro impiego per reti wireless real–time, particolare enfasi è stata rivolta alle tecniche di accesso al mezzo specificate nei citati standard.

Sulla base di quest'analisi, sono stati proposti alcuni metodi originali per incrementare l'affidabilità di questi sistemi, considerando ad esempio nuove politiche di ritrasmissione per reti basate su polling ciclico. Inoltre sono stati proposti nuovi algoritmi per l'adattamento automatico del rate di trasmissione per reti IEEE 802.11, progettati per l'impiego specifico in un contesto di reti real–time.

Infine, considerando che l'affidabilità di una rete in questo contesto dipende strettamente dal comportamento fisico dei componenti impiegati, viene proposto un approccio sperimentale per la misura e caratterizzazione dei ritardi introdotti dai dispositivi di rete, allo scopo di fornire metodi e modelli adeguati in un contesto di progettazione di rete.

PREFACE

Recently, an increasing interest in the adoption of real–time wireless networks for industrial communication systems, as well as for distributed measurement systems, has been experienced, by system manufacturers, designers and researchers. Among these communication systems, Industrial Wireless Sensor Networks (IWSNs) represent a very promising opportunity. From a behavioral point of view, these networks are then required to provide very tight performance with respect to general purpose networks, in terms of, for example, tight reliability, very low jitters, deterministic traffic prioritization, etc. These performances have to cope with the intrinsic behavior of the wireless channel, which is substantially unreliable. The characterization of the achievable performances, as well as the proposal of methods for improving IWSN capabilities require, besides the execution of theoretical analyses, the deployment of suitable measurement systems capable of highlighting and modeling the sources of difference between the expected and the actual behavior of the communication network.

Deviations with respect to the theoretical expected behavior can be experienced for several reasons. Among the other, it is worth mentioning interferences between coexistent radio systems, the non–ideal operation of communication devices and protocol issues. It is worth noting that the scientific and research community primarily developed communication standards specifically tailored for “general purpose” applications (such as Bluetooth, WiFi, etc). Only during the recent times there has been an increasing interest in the realization of standards purposely developed for the deployment of real–time wireless communications, such as for example WirelessHART and ISA 100.11.

Furthermore, the scientific literature in electromagnetic compatibility recognizes that the main source of disturbances for wireless systems is typically found in an intentional radio communication in the same frequency band.

Network failures due to interferences partially justify the resistance in the adoption of wireless systems in the depicted industrial scenario. As a matter of fact, unintentional disturbance sources (like for example electric machines, welders, etc) typically produces noise components in lower frequency bands with respect the common Industrial, Scientific and Medical (ISM) 2.4 GHz band adopted by the aforementioned wireless systems.

Despite the efforts spent by the scientific community to solve the principal issues relevant to this topic, actually the adoption of such systems in real applications is not as widespread as expected. Perplexities felt by designers on the effectiveness of such systems are a limiting factor for the deployment of IWSN, even if the appealing characteristics provided by these systems could lead to new and interesting perspectives in this field.

The goal of the present work is to provide some contributions in the field of measurements for the performance analysis and the characterization of real–time wireless networks for industrial applications and distributed measurement systems. Suitable measurement setups have been realized to verify the effects of interference between wireless networks sharing the same channel, and at the same time to evaluate delays introduced by real communication components. Moreover, some original solutions for IWSNs performances' optimization are proposed, exploiting new retransmission schemes for polling-based networks and new rate adaptation techniques for hybrid RTE/IEEE 802.11 networks. To this purpose, the analysis has been conducted through both simulations and measurements campaigns. The final aim is increasing the knowledge on the behavior of real–time wireless communication systems, possibly helping designers to reduce the distrust in the adoption of wireless communication systems in industrial scenarios.

In the following, an outline of the chapter of this thesis is provided. At first, an introduction to Industrial Wireless Networks is given in **Chapter 1**, where the state of the art of the field as well as the main issues are highlighted. This will serve as a basis for the complete understanding of the rest of the work.

In **Chapter 2** and **Chapter 3** the standard IEEE 802.15.4 and IEEE 802.11 are, respectively, presented. These brief overviews would present the key aspects of these communication standards and highlight the major control knobs to increase their performances in the application field of interest. The Chapters introduce also the adopted models for the calculation of the Packet Error Rate (PER) for each of these communication standards.

In **Chapter 4** a simulation tool developed during the PhD course is presented. The tool is based on OMNeT++ [50], and has been specifically designed for the study of coexistence issues, enabling a cross–layer approach. With such a tool it has been possible to analyze interfering wireless networks in the same environment, sharing therefore the same communication medium. The experiments performed in several case studies provide also a mean for the characterization of the tool, that has been subsequently used for simulation-only analysis of new protocols. The simulator, indeed, served as a design platform for some of the original proposed solutions.

In **Chapter 5** some polling-based protocols for IEEE 802.15.4 networks are proposed. As a matter of fact, often polling procedures are exploited by a controller to exchange data with the connected sensors/actuators. In the case of interferences from other networks, as well as communication errors due to the harsh industrial electromagnetic environment, a high number of retransmissions could be experienced. Considering the fixed cycle time, and hence the limited number of transmission retries, several nodes could result unserved. Therefore, some new and effective retransmission schemes are proposed, able to smartly schedule transmission retries, sensibly decreasing the number of unserved nodes and transmission attempts, while enabling a good fairness among nodes. Simulation and a detailed experimental analysis proved the effectiveness of the proposed techniques.

The differences between theoretical, simulative studies and experiments outcomes revealed how the real component behavior impacts on the overall performances of an real–time communication system. In this perspective, a comprehensive experimental measurement session will be presented in **Chapter 6**, relevant to the case of a IEEE 802.11 wireless extension of a Real–Time Ethernet (RTE) network. The delay introduced by some real Access Points (APs) is measured through an effective measurement system. Models are then derived, and an original method, based on deconvolution and measure by substitution methods, is presented, allowing the characterization of such components starting from simple measurements on the traffic flow.

In the same scenario of IEEE 802.11-based networks, one of the major control knob for this kind of communication standard is the presence of several modulation schemes, *i.e.* transmission rates for packet delivery. It is a common practice to adapt the transmission rate to the channel conditions, in order to avoid packet losses while maintaining the higher possible throughput. This technique is known as rate adaptation. In **Chapter 7** an in-depth analysis is presented, from a measurement point of view. Moreover, two originally developed rate adaptation algorithms, specifically conceived for real–time communication are presented. Some extensive simulation tests confirmed their higher effectiveness with respect to the classic, and commonly implemented ARF algorithm, in the case of typical field level industrial traffic. The forthcoming need for real-time multimedia traffic in industrial networks suggested also some additional experiments, where these

algorithms again showed good performances.

Chapter 1

Real–Time Wireless Networks

Real –time communication networks[1] are nowadays widely adopted at the lowest levels of industrial processes as well as by many distributed measurement systems [83], [57]. In these contexts, traditionally, the network traffic is characterized, mainly, by the transfer of small amounts of data with tight timing constraints even with, during the years, the handling of multimedia real–time (RT) traffic [59, 63], has emerged as a further possible requirement.

In the above scenarios, fieldbuses [67] have been initially the most appealing solutions since the end of '80s when they have started to be deployed. Subsequently, roughly at the end of the '90s, field networks based on the well known Ethernet technology have begun to be introduced. These ones are often referred as either Real–Time Ethernet (RTE) or Industrial Ethernet networks [14, 15]. They are characterized by strong performance figures, in that they are able to provide high transmission rates (typically up to 100 Mbit/s), very limited and predictable transfer times, high determinism, low jitters.

Finally, in the last years, wireless networks have started to be considered as an interesting solution for RT communication as well [76]. Some of the reasons supporting this interest are the simplification of cabling, the increased degree of mobility of devices, the possibility of adopting simpler configuration schemes, the ease of integration with handheld devices, etc. Nonetheless, some drawbacks may make sometimes critical the adoption of wireless systems for Real–time (RT) communications, mainly due to detrimental interactions among devices at the physical layer, resulting in corruption of packets, increased retransmission attempts, and hence in longer and random transmission delays.

Moreover, differently from the other types of industrial networks, at least at the very beginning of the employment of wireless systems for RT communications, no specific protocols have been developed. Thus, most of the analysis carried out in the past years were aimed at evaluating the suitability of commonly available wireless networks (for example IEEE 802.11, IEEE 802.15.4 and Bluetooth) for RT applications (see [78, 76] and the references therein for further details). In this context, recently, IWSNs [27] have started being

[1] For further reference, consider also [62] for an in-depth overview on the topic of the performance of real–time networks in real applications.

considered as an interesting opportunity. Purposely developed protocols like ZigBee [82], WirelessHART [75] and ISA 100.11 [35], to mention some, are already available in this field. Moreover, specific case studies of IWSNs applications have already appeared in the scientific literature such as, for example, the intrusion detection system described in [64] and the analysis of the temperature effects on outdoor sensor nodes given in [9]. It is worth observing that the availability of all the aforementioned kinds of communication systems allows for the effective implementation of mixed configurations using bridge/-gateway devices. In particular, hybrid wireless/wired networks, as described in both [12] and [65], represent truly promising solutions in the context of industrial communication systems.

In order to comprehensively discuss about the implementation of hybrid wireless/wired networks in RT applications, and specifically on the topic of measurements and simulations on this kind of communication systems, some considerations about the performances required to these networks are necessary, as well as the main issues degrading performance indexes of interest are to be identified. As a matter of fact, RT communication networks are often required to provide very tight performance figures in terms of both real–time and determinism [83, 3, 56, 20, 18].

Performance indicators

Differently from traditional telecommunication systems, whose performance are usually evaluated on the basis of some "macro" indicators like transmission speed, overall throughput, maximum number of nodes, etc., RT networks are much more concerned with specific metrics imposed by the applications they are conceived for. Thus it may be requested, for example, that a station connected to an industrial network is able to periodically send short messages with very low jitter and/or that, at the same time, the network protocol has to grant the medium access to any station of the network within a specific time slot in order to timely deliver (possibly) critical messages.

A valuable contribution in the context of performance evaluation of RT networks is represented by the IEC 61784–2 International Standard [34] which defines a set of performance indicators. The standard has been actually conceived for a specific type of communication systems, namely real–time Ethernet networks. Nonetheless, most of the indicators it defines may be effectively employed to characterize the performance of wireless RT network as well, as described in [22]. The more relevant performance indicators are here briefly described.

Delivery time it represents the time needed to convey in real-time an application packet containing a relevant payload from a source node to a destination. The standard specifies that the measurement should be performed in the case of a successful delivery at the first attempt, and also in the case one frame is lost, with the consequent retransmission. All the times involved in this process have to be included in the measurement. This requires, in the case of wireless communications, to consider times that are not specifically related to the actual frame transmission, such as, for example, interframe spaces, backoff periods, time-outs etc.

Cycle Time it is typical that a considerable amount of the traffic on a RT network is represented by data periodically exchanged. In this case, the cycle time is therefore a key parameter, that characterizes the overall behavior of the network. It is lower bounded by the *Minimum Cycle Time* (MCT) represented by the time necessary to poll all the slaves attached to a controller in sequence, without introducing any idle time. Given the randomness of the internal behavior of components [60], and the MAC specifications of the given communication standard, the MCT value is also subject to variations.

Real Time Throughout it represents the total amount of application level data, in Bytes, conveyed on one link per second, exclusively relevant to real–time traffic. Since typically the amount of data a sensor/actuator exchange with the master node is fixed on a per–link basis, the RTT is inversely proportional to the cycle time. Given that the MCT, while a certain degree of variability is present, is in general lower bounded, the RTT present also an higher bound.

Non–Real Time bandwidth It is the percentage of bandwidth which non real–time communications can use on a specific link. It measures of the ability of a network of simultaneously serving real–time (critical) traffic as well as general purpose one.

As a matter of fact, a clear assessment of the performance of RT communication system, in particular for wireless systems, represents a key aspect that needs to be adequately investigated. Most of the literature works deal with protocol analysis, providing results mainly derived from theoretical models as well as numerical simulations, while only few contributions are concerned with practical measurements. Typical reasons behind this lack of practical results are found in the efforts and competencies required for the execution of experimental measurements (both in terms of personnel and instrumentation), in the relevant time needed to perform such experimental campaigns, and in the unavailability of the real plants, since they are mostly dedicated to their current activities.

In the case experimental sessions have been conducted, it often happens that a comparison between the results obtained from practical tests and those deriving from theoretical/simulative analysis, reveals considerable differences, where experimentally obtained performance figures are typically worse than those expected. Such an aspect represents a relevant issue, since it makes difficult to precisely estimate in advance the actual behavior of RT communication systems once they will be deployed in practical applications. Consequently, it may happen that the design of the whole system that employs the network needs to be reconsidered in the light of the actually achievable performance (some expected goals might be either withdrawn or considerably reduced) leading to longer re–commissioning phases as well as to possible limitations of the production sessions.

Differences between expected and measured network behavior

There are several causes, addressed briefly in the following, explaining such a discrepancy. However, even a simple introductory example could provide some useful insights. Indeed, let us suppose that in a networked control system a set–point has to be delivered by a controller to some actuator(s) with a period of 100 ms and a jitter below 10%. A simple analysis of the available solutions shows that these performance figures are easily achievable by most of the commercially available industrial communication systems. However, if during operation the Bit Error Rate (BER) of the network suddenly increases, as it often happens in industrial sites because of a burst of noise, there will be a non-negligible amount of frame retransmissions that, consequently, increase the jitter on the set–point delivery. On the other hand, if some of the used components introduce unpredictable, even constant, delays that were not taken into consideration in the analysis, then the actual cycle time of the network could overcome 100 ms, compromising the timely set–point transmission. In both these situations, the networked control system does not behave as requested, compelling to review its design. Furthermore, performance degradation may lead to (possibly temporary and/or partial) network unavailability. In fact, the increase of the BER may cause the temporary exclusion of some devices from a network since they are not able to communicate either with each other or with other devices (this situation may occur frequently in wireless communication systems that are implicitly very sensitive to Signal to Noise Ratio (SNR) fluctuations). Furthermore, the re-introduction of these devices in the network (which can only take place when the BER returns to an acceptable value) may require additional time before they re-enter the full operational state. Similar effects (i.e. temporary network unavailability) may be experienced by a slave device that, due to some internal elaboration delays, does not respond timely to the queries of a master so that it is considered faulty and, as such, excluded from the network. This is typical of tightly synchronized networks, as described in [36], where the limitation as well as the precise estimation of the communication latencies, represent a mandatory goal.

In general, it may be concluded that, since the performance required to RT communication systems may be often critical, the effects of even minimal discrepancies may reveal very dangerous. Consequently, the factors that potentially lead to practical performance degradation have to be carefully investigated and, if possible, corrective actions have to be undertaken.

The main causes that impact on the performance of IWSNs and, in particular, of their theoretical and simulative analyses may be roughly divided in three major classes, namely:

- ✓ internal behaviors of the components
- ✓ communication errors
- ✓ hardware failures.

The latter ones are regarded as the failures of the components employed by the communication systems. Examples are cable and/or transceiver breaks, faulty network adapters, defective hub/switches, etc. Although the effects of such failures may be severe, if they are permanent, as it often happens, then performance degradation of the network occurs only for limited periods of time. Indeed, most of the industrial networks have built–in diagnostic facilities that allow to precisely detect defective components and, at the same time, to undertake the corrective actions. Clearly, this may lead to some network inactivity periods. Nonetheless, they are very sporadic and limited to the time necessary to replace the defective component(s). This type of problems, however, may have more dangerous effects if the failures are intermittent. In these cases the error detection is very difficult leading to frequent undetected malfunctions with the consequent performance degradation.

Conversely, both the internal behaviors of the components and the communication errors represent very challenging and yet quite unexplored causes of divergence between simulations and experimental results. The implementation of RT communication networks requires the employment of several types of devices such as sensors/actuators, I/O modules, controllers (PLCs, PCs, embedded boards) and network components (hubs, switches, transceivers, etc.) to mention the most common ones. These components are usually designed for industrial applications in the sense that they present a noticeable electrical/mechanical robustness (as it is often required by International Standards). Nonetheless, from the communication point of view, they may introduce non negligible delays. Some examples could clarify this concept.

(*i*) The filtering operations carried out by I/O modules can often require execution times in the order of some milliseconds, i.e. they may be comparable with the cycle times that RT networks might have to provide;

(*ii*) the elaboration times of the "intelligent" devices may be relevant. In some cases they could be predicted (if the devices use real–time operating systems, for example), but for some components this might be not possible;

(*iii*) the buffering function typical of switch devices, as described in [42], represents one of the most relevant causes of communication latencies for RTE networks.

These delays are difficult to know *a priori* since the manufacturers often do not provide a satisfactory characterization of their devices in this direction. More importantly, the delays may be of random nature.

In the context of RT networks, an explanatory yet realistic example is represented by AP devices that may be used in industrial applications to implement wireless extensions of already deployed wired networks. These devices serve as bridges between Ethernet and IEEE 802.11 Wireless LAN (WLAN) segments converting frame formats of one network to the other and *viceversa*. As pointed out in [43], an access point may introduce two types of delay due to, respectively, internal latencies and queuing of frames. This latter delay is mainly related to the size of the transmitted frames and, as such, it may be calculated analytically. Conversely, the latency introduced by an AP relies on its internal structure and it can be derived only via experimental measurements.

An example is reported in the following. Two different, general purpose, commercial components have revealed a random latency characterized by a mean value of some hundreds of microseconds with a relevant variance. Analogous measurements executed on a Siemens SCALANCE W784-1 access point, a device specifically tailored for industrial applications, showed a similar behavior, as can be seen in Fig. 1.1 on the facing page, that reports the EPDF of the latency actually introduced by the AP. The measurements are relevant to a streaming of 1000 minimum size Ethernet packets sent with a period (indicated as T_P in Fig. 1.1 on the next page) of 3 ms. It may be noticed that the delay of the access point, D_{AP} spans roughly in the range 140-200 µs. A final relevant consideration about the internal component behaviors is concerned with the actual conformity of the components themselves with the standards they (claim to) comply with. As a matter of fact, it may happen in practice that the devices do not adhere exactly to the official documents emanated by the competent Standardization Bodies, as it is discussed, for example, in [17] for IEEE 802.11 cards. Such a problem is mainly due to the missed execution of an exhaustive

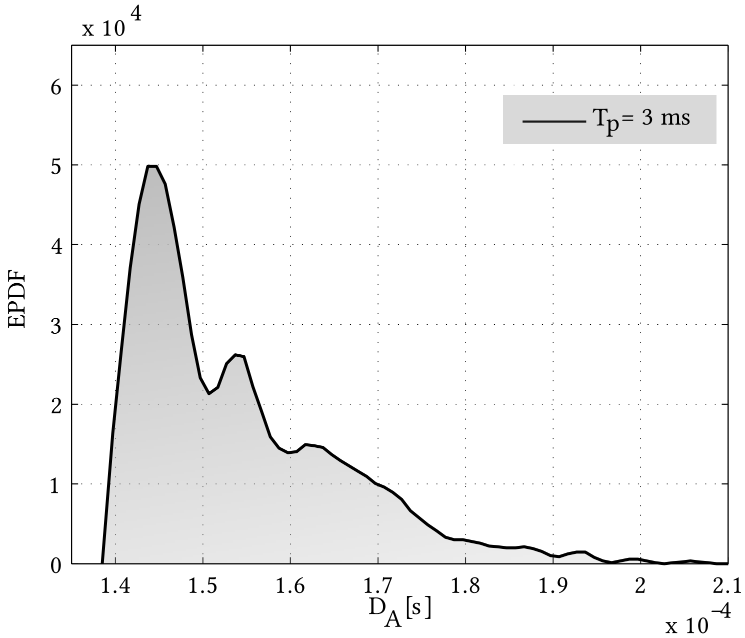


Figure 1.1 – EPDF of the latency introduced by a Siemens SCALANCE AP

session of compliance tests. In other words, it often happens that the components do not undergo an adequate set of tests aimed at verifying their compliance with the standards. A meaningful example is relative to Profibus devices: while a lot of devices are available (as can be verified at: `http://www.profibus.com/products`), only a few of them are certified by the Profibus competence centers. In this context, the paper [38] provides an interesting assessment on some Profibus master boards that do not strictly comply with the specifications leading, in this case, to possible coexistence problems in multi-master configurations. In general, it may be stated that, the ultimate effect is that the behavior of non certified devices may be unpredictable, possibly leading to unexpected results in practical implementations.

Communication errors, finally, represent perhaps the main source of errors in the modeling of IWSNs behavior. These refer to the corruption of bits within the frames transmitted over the networks. Such a phenomenon may be of particular relevance in industrial environments, that are frequently subjected to severe levels of electromagnetic noise, resulting in non–negligible BERs. It is worth noting that the problem represented by communication errors is commonly thought as exclusively pertinent to wireless systems, since these are inherently error prone. Conversely, this is not true, since wired systems deployed in industrial sites may be characterized by quite high BERs as well. This results necessarily in a performance degradation, as described for example in [79], [19] and [71] for the Profibus fieldbus.

Moreover, when wireless networks are employed, an important source of communication errors has to be considered. It is represented by the interference, deriving from the presence of other communication systems transmitting in the same band. This problem is particularly evident for the 2.4 GHz ISM band which is nowadays very crowded. Interesting contributions relevant to the interference effects in networked control systems are given in [5, 13, 25] for the case of IEEE 802.15.4, while for example [9, 60] are focused on networks implemented with IEEE 802.11 devices.

Wireless communication and Soft Real–Time systems

For the outlined reasons, wireless networks are often regarded as not suitable for the implementation of real–time distributed systems. Actually, real–time systems can be classified in hard and soft RT. The former class is defined as that in which it is imperative that reactions occur within the specified deadline, because missing a deadline would lead to a catastrophic behavior. Soft RT systems, instead, could accept an occasional missing of a deadline, even if it remains important that the response time of the system is under control. It is noteworthy that in soft RT systems, it is possible to define a probability threshold, under which a missed deadline is acceptable, whereas in hard RT that probability is definitely zero.

The interest in wireless networks has grown on the basis of some interesting reasons. Firstly, the availability of communication standards able to provide very high packet transmission rates and small timings, exploiting complex and robust modulations and Forward Error Correction (FEC) algorithms. It is the case, for example, of IEEE 802.11 WLAN [32], that provides performance figures capable of coping with the severe requirements typical of the industrial fields. Several transmission rates are available for IEEE 802.11g networks, ranging from 6 to 54 Mbps. They use different modulation schemes that, consequently, provide different level of robustness. The automatic rate selection on the basis of the actual channel state is a well-known topic in the telecommunication literature. However, often the proposed scheme are not conceived for real–time communications needs. The design of specific algorithms for automatic rate adaptation could represent an important field for the optimization of the performances of wireless systems in soft RT scenarios (see the following Chapter 7 for an in–depth analysis of this topic). Moreover, an effective and efficient use of the wireless channel characteristics. For example, problems due to multipath propagation and obstacles (to cite some) can be tackled through Multiple Input–Multiple Output (MIMO) techniques, as for example in the standard IEEE 802.11n [32, Amendment 5]. Finally, even if the behavior of the wireless channel, and the delays introduced by network components often exclude the possibility to realize a hard RT system, soft RT systems could be obtained adopting wireless networks.

From the above considerations it follows that the provision of experimental data relevant to the performance of RT wireless communication systems is an issue of even more growing interest. However, in order to obtain the best exploitation of these data they should be, in a certain way, as much general as possible. In practice, the experimental data should be usable not only by the specific application they are relevant to but, more importantly, they should be employed to precisely predict the behaviors of other systems. For example, the complete assessment of the delays introduced by a specific commercial component could lead to the implementation of a much more realistic simulation model of the component itself. In the same way, the exact knowledge of these delays may greatly improve the correctness of the theoretical models of networks which employ that component.

Accurate measurements are hence needed in order to thoroughly assess the actual behavior of such systems as a whole, and that of specific components in particular. To this purpose, particular attention should be given to an intrinsic difficulty arising when measurements on real devices are considered. As a matter of fact, often upgrading off-the-shelf hardware components, to include the needed improvements is very difficult, since the manufacturer does not make available the information about the software (firmware)version implemented on the specific device. As a consequence, experiments become rather expensive even when mild modifications to devices should be made.

Chapter 2

An introduction to the IEEE 802.15.4 standard

The term ZigBee® is often used to refer to networks based on the standard IEEE 802.15.4, while there is a clear difference among these two terms. ZigBee® Alliance is, in fact, a non-profit association, founded in 2002, with the aim of supplying low-cost, low-power, wireless mesh networking standards focusing on monitoring, control and sensor applications. On the other hand, task group 4 of the Institute of Electrical & Electronics Engineers (IEEE) 802.15 committee realized, and released a standard, the IEEE 802.15.4, on may 2003, which encompasses specifications for the MAC and PHY layers tailored for Low-Rate Wireless Personal Area Network (LR–WPAN).

Actually, ZigBee® supplies a suite of high-level protocol layer (from Network Layer (NET) up to Application layer) to be used in conjunction with IEEE 802.15.4 based radios. This type of communication network would place in a different segment with respect to other common solutions, as for example Bluetooth and WiFi ones. The low-power consumption makes devices last for years without changing the batteries. The low transmission power gives rise to a communication range that spans from 10 to 75 meters, depending on the environment. Multi-hop techniques enable coverage of greater distances, *i.e.* several nodes serve as *relays* for other ones, collecting their frames and forwarding them through specific links, for example in a cluster-tree topology.

In the following, only a brief overview of the low-level IEEE 802.15.4 standard will be discussed, presenting the main features of interest, such as modulation scheme, frame formats, bit error probabilities, etc.

The type of devices specified by the standard allows for a long operational life, a low transmission power, and a consequent low transmission rate. This characteristics fit well in a large set of applications, such as domotics, home and building automation, energy monitoring, factory automation systems. It is worth to note that all these applications do not require an high throughput[1], nor a wide coverage area. More-

[1] Net quantity of data transmitted over the network, without considering protocol overheads, in the unit time

over, they do not require the usage of common protocols stack like Transport Control Protocol/Internet Protocol (TCP/IP), due to their heavy needs in terms of memory and energy (an exception coming from [47]).

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

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 foreseen by the standard are sketched in Fig. 2.1.

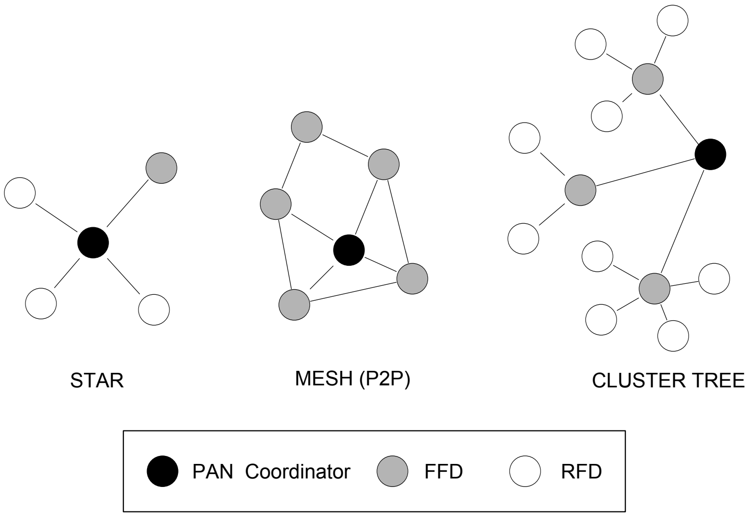


Figure 2.1 – ZigBee network topologies.

2.1 MAC Layer description

In this section only some key points of this critical protocol layer will be discussed. Indeed, the MAC represents a fundamental level in the ISO/OSI stack, and seriously impacts on the communication performances. It appears clear, therefore, that the complexity of the operations, already described in [31], is not suited to be detailed here.

The standard provides two different types of MAC services:

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).

which enables, among the other, the following features:

✓ *beacon* management;

✓ channel access;

✓ *GTS* management;

✓ sending of Acknowledgment (ACK) packets;

✓ association to a PAN.

Moreover, the standard also enables services for authentication and ciphering: communications can be based on an access list (without security), or managed with authentication procedures and Advanced Encryption Standard (AES) cryptography with 32 or 128 bits.

Finally, the transmission of ACK packets may be requested by the transmitter to ensure a reliable communication. This allows the adoption of error correction techniques. Nonetheless, FEC algorithms are not supported because of the high redundancy they introduce, decreasing the throughput to unacceptable levels. Only Automatic Retransmission Query (ARQ) error correction algorithms fit well with the low bandwidth system described by this standard.

2.1.1 The CSMA/CA algorithm

The wireless medium provides a powerful communication channels, in which transmissions are broadcast to all the radios in the coverage area of the transmitter. It appears clear, therefore, that sharing the same channel could result in serious interference issues if devices does not respect a strict policy for channel arbitration.

Many channel access techniques have been discussed and actually available, based both on deterministic and stochastic approaches [52, 26, 2].

For example, deterministic medium access methods are the common Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA). In each case every node wishing to transmit can use a “dedicated” and pre-assigned radio resource. It could be a time slot in the former case, where the time is divided into a number of slots for each participating node. In an analogous way, it could be a portion of the frequency band in the latter case.

These approaches, under specific hypothesis, completely avoids collisions among transmissions. This high reliability, and the precise time schedule of messages makes often these methods the preferred ones for real-time control applications. It is worth to note that the scalability of the methods is poor, as long as mechanisms have to be provided in order to adapt to evolving network configuration. Moreover, they are not the best choice from the performance point of view. Slots are indeed assigned to each specific node, which can start to use them without any other care. In this way, it appears clear that assigned but unused resources are waste.

In stochastic methods the approach is to avoid statically assigned resources, allocating the channel, on the basis of a contention procedure, to the node winning the contention, thus optimizing resources utilization (in some way this can be thought as an higher efficiency). The contention procedure, however, has the drawback of introducing a certain, non negligible probability of collision among different nodes, asking for access to the channel (*e.g. hidden node* problem, etc). Stochastic medium access algorithms present, on average, a more flexible behavior and are profitable in cases when no central controllers or dynamic environments are present.

The standard IEEE 802.15.4 makes use of the Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) algorithm. It is one of the principal stochastic technique and, probably, the most widespread one. As an evidence, many wireless communication standards adopted the CSMA/CA algorithm, as for example IEEE 802.11.

The algorithm approach is rather simple: if a node has data to transmit it has to sense the channel for other interfering transmissions. If the channel is “free” then the node can start sending its message, otherwise the node has to wait for a random interval of time (called *backoff*) and retry to access the channel. More schematically, the method can be summarized as follows. Assuming that a node has data ready to be sent, then (see Fig. 2.2 on the next page):

- ✓ wait for a random time $\tau \in \mathbf{U}(0, 2^{BE} - 1) \cdot BP$, where $\mathbf{U}$ indicates a random variable with uniform distribution, BE is the *Backoff Exponent* and BP is a *basic period* (defined in [82]);

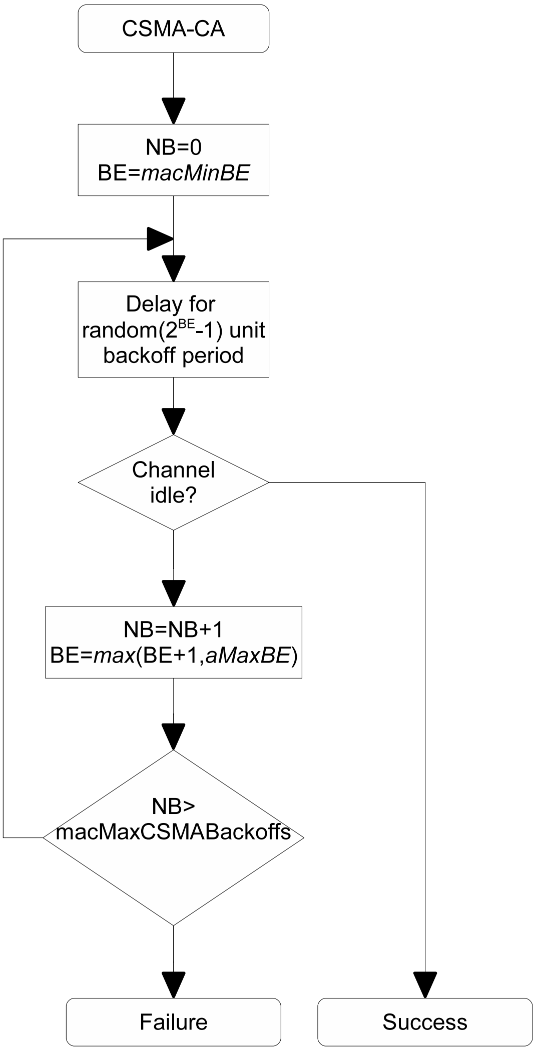


Figure 2.2
The beaconless CSMA/CA algorithm described in the IEEE 802.15.4 standard.

- ✓ perform the Clear Channel Assessment (CCA), *i.e.* sense the channel;
- ✓ if the channel is sensed free, then start the transmission;
- ✓ if the channel is sensed busy, then update $BE = \min(BE{+}1, aMaxBE)$, 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 signaled, else repeat the algorithm.

The channel access algorithm just described is adopted in the so-called *beaconless mode*.

This is in contrast with the *beacon mode*, also foreseen by the standard. In this case a *superframe* structure for transmission is provided, through which the PAN coordinator manages the access to the channel. The coordinator uses special packets, *beacons*, to synchronize all devices that are connected to the PAN, 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.

A sketch of the superframe structure is found in Fig. 2.3. It is composed by two parts: an "active" part and an "inactive" one. This latter is used by devices for entering an idle state in which they do not communicate each other and have low power consumption.

The active part is subdivided in time-slots, in two portions: the Contention Access Period (CAP) period, and the Contention Free Period (CFP) period. In the CAP period, communications make use of the CSMA/CA algorithm to access each one of the 9 slots. The CFP is divided in 7 slots, called Guaranteed Time Slots (GTS), in which a node access the channel without contention.

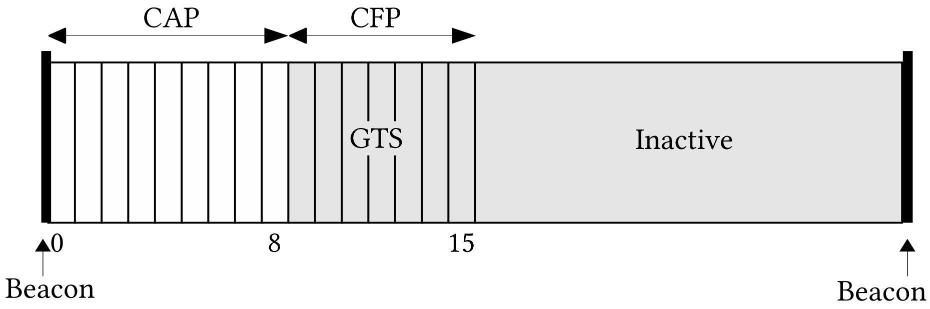


Figure 2.3 – The superFrame structure in the IEEE 802.15.4 standard.

2.1.2 MAC frame

The IEEE 802.15.4 MAC layer provides four type of datagrams:

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

Data Frame the datagram used in data transfer;

Acknowledgment 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.

The maximum length for MAC packets is limited to 127 Bytes, a small size if compared to other wireless communication standard packet length, typically about 512 Bytes. This imposes an upper bound to the header length, to provide an acceptable throughput.

The structure of datagrams is shown in Fig. 2.4. The first part of a frame is a 16 bit-long Frame Control Field (FCF), carrying various information about the packet (*e.g.* type, addressing mode, security). An 8 bits-long Sequence Number (SN) follows, which is a progressive number, identifying the packet. Finally, the MAC Header (MHR) provides four address fields, with the source and destination PAN identifiers and addresses. The addressing fields have variable length (16 or 64 bits), depending on the type of address chosen, and can be omitted in acknowledgment frames.

The payload section follows this MHR, while the Frame Check Sequence (FCS) ends the datagram. The FCS is a 16 bits-long field, containing a Cyclic Redundancy Code (CRC) for error correction, calculated on the header and the payload. The CRC polynomial is:

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

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 2.4 – The general structure of a MAC frame in the IEEE 802.15.4 standard.

Addressing

Each device has an assigned 64 bit IEEE address, while a single PAN can be identified through a 2 Bytes field. The maximum length of the addressing fields in the MAC frames is therefore 20 octets. With 64 bits addresses it is possible to identify $2^{64} \simeq 1.84 \cdot 10^{19}$ devices. Moreover, a broadcast address (0xFFFF) is provided.

The standard provides also a short address mode, which makes use of 16 bit addresses. A short address is valid only within a PAN, and is the PAN coordinator that assigns the address in the node association stage. This address space allows for 65535 devices in a network, covering almost all possible WSN networks of interest.

2.2 PHY layer

2.2.1 Overview of the main services

Two types of primitives for the physical layer are defined by the standard: the PHY *management service* and the PHY *data service*. The last one manages the exchange of frames (PPDU). The former, instead, provides some control services by realizing an interface to the Physical Layer Management Entity (PLME).

Among the services provided by the Physical Layer (PHY), the main are:

- ✓ Energy Detection (ED)
- ✓ Link Quality Indication (LQI)
- ✓ Clear Channel Assessment (CCA)
- ✓ Transceiver control (activation/deactivation)
- ✓ Packets management
- ✓ Transmission channel selection

The ED service provides an estimation of the received power within the bandwidth of radio channel. The estimation is done over 8 symbols time, and no attempt has to be made to decode signals. The outcome is a 8 bit unsigned integer (thus its value ranges from 0x00 to 0xFF). A *zero* shall indicate an energy lower than 10 dB above the sensitivity of the receiver. The standard also specifies that the dynamic range of ED estimations must be at least 40 dB. In this interval returned integer values should be converted in power levels (*i.e.* dBm) in a linear scale with an accuracy of ±6dBm. It is worth observing that typically, Received Strength Signal Indication (RSSI)[2] subsystems are often employed to provide this service. Indeed, these two acronyms are often considered synonyms.

Another useful indicator related to the "quality" of the incoming signal is represented by the LQI measurement. This is evaluated at each packet reception, and is strongly related to the BER. This one is calculated on the Synchronization Frame Delimiter (SFD), that is a set of bits known *a priori*. LQI is given with 1 byte (unsigned) precision.

Another basic functionality provided by this standard is the CCA function. Since the wireless channel is shared among many nodes, and interference should be avoided to increase reliability of the communication, a Carrier Sense Multiple Access (CSMA) technique

[2] 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 was a technology already present on IEEE 802.11 compliant devices.

is adopted. However, full-duplex capabilities are excluded, thus it is mandatory for a station wishing to transmit to listen to the channel before its transmission to check for any other transmissions. A station is cleared to send if the channel is somehow felt “free” (or “idle”), while it shall defer its activity if the channel is sensed “busy”. In order to detect the state of the channel, the IEEE 802.15.4 standard foreseen three possible CCA operating modes:

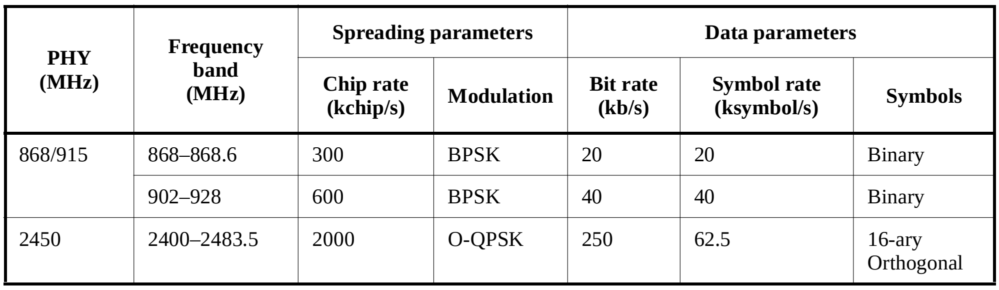
CCA Mode 1: Energy above threshold The channel is sensed busy if the in channel power overcomes a chosen threshold (CCA threshold, typically -77 dBm).

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.

The CSMA medium access method is one of the main cause of coexistence problems with other standards operating in the ISM band, as well described by [4].

The standard introduces a subdivision of band occupancy into two separate frequency bands. It uses common ISM bands, defined in ITU–T 5.138, 5.150 and 5.280 Radio Regulations, namely the 868-915 MHz and the 2.4-2.5 GHz ones. The former is adopted for lower bit-rate transmissions and is subject to regional limitations, thus of minor interest in this work. The latter, instead, provides the highest bit-rate, namely 250 kbit/s, and is located in the “crowded” 2.45 GHz band, where many other common wireless systems are possibly present. The following Table 2.5 gives a clearer picture of the situation, giving also the transmission parameters foreseen by the standard document.



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 2.5 – IEEE 802.15.4 bands and transmission rates.

In the rest of the work, without loss of generality, we will refer only to the 2.4-2.5 GHz band, since it represents the most interesting case for control systems in an interfering scenario.

PPDU description

As in common communication systems, application layer generated data are split and encapsulated into packets crossing the ISO/OSI protocol stack down to the physical transmission medium. At each new layer, from up to down, a packet is further encapsulated adding some level-dependent information (in the form of an header), and then released. That is, in the common terminology, the packet arriving from layer $n-1$ is called, at layer n, Service Data Unit (SDU), which will be encapsulated in a layer n packet, which in turn is called Protocol Data Unit (PDU). Often, a character is added to these acronyms to indicate the packet level; for example, the MAC Service Data Unit (MSDU), PPDU, etc.

When the final packet is ready to be transmitted, the PHY layer adds its header, forming the final PPDU. This will be modulated and transmitted by the radio chip. In the IEEE 802.15.4 standard the PPDU is composed of the parts depicted in the Fig. 2.6.

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

Figure 2.6 – The PPDU structure in the IEEE 802.15.4 standard.

Each packet is transmitted and received starting from the Least Significant Bit (LSB). As shown in the Figure, the first part of the PPDU is the Synchronization Header (SHR), subdivided in the *preamble* and the SFD. This part is essential in packet-oriented communications, since packets are generated asynchronously: the receiver have therefore to synchronize with 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.

Modulation

The modulation process makes use of the DSSS spreading technique. In particular, each *symbol* (4 bits) is converted into a 32 chips long spread sequence, through a predefined look-up table. These are pseudo-noise sequences with white noise-like spectral properties [26].

This process increases the transmission bandwidth with respect to that of the modulating signal. Spreading the available transmission power on this wider band reduces emissions on that specific band, and could results in a communication system inherently less disturbing other ones. It also increases the communication robustness since symbols are "coded" into well-defined chip sequences, and help the system overcoming impulsive interference [26]

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 follow this relation with the channel number:

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

which is described in Fig. 2.7.

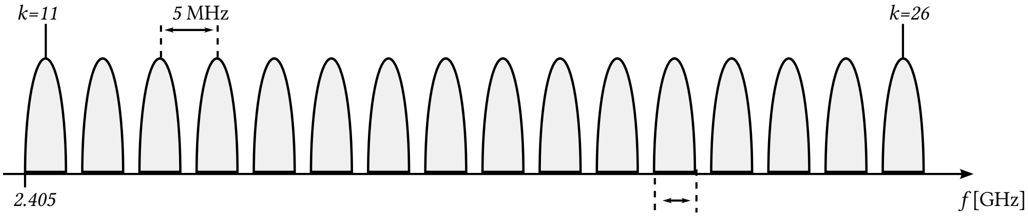


Figure 2.7 – A representation of the spectrum used in this standard in the ISM band.

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 ±40 ppm.

The sensitivity of receivers compliant to the standard shall be at least -92 dBm: to be compliant with the standard a receiver must correctly receive a signal with a power greater or equal than this threshold.

2.2.2 Transmitter functional blocks

The modulation scheme for a compliant radio can be subdivided into three blocks, see Fig. 2.8 on the next page:

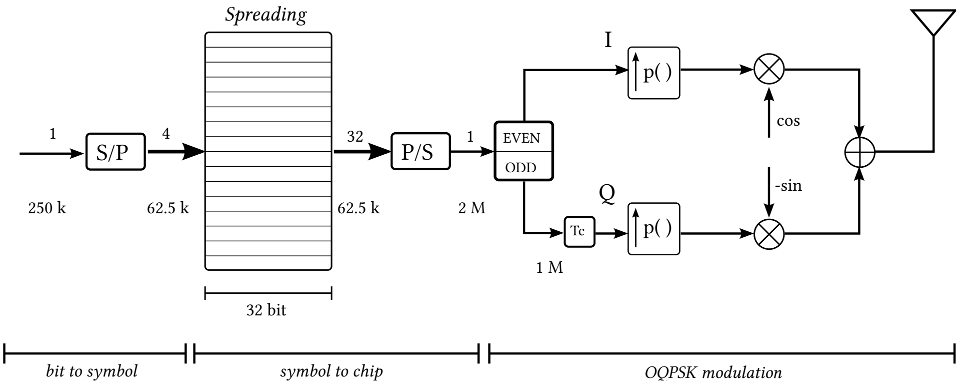


Figure 2.8 – Block diagram of IEEE 802.15.4 modulator.

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

A PPDU is processed starting from the LSB, leading to the preamble being the first processed part. Then bits are grouped in sets of four, forming symbols. Starting from the nominal bit-rate, 250 kbps, with a bit period is $T_b = \frac{1}{250\ kbps} = 4\,\mu\text{s}$, then the *baud-rate* achieved is 62.5 kbps. The symbol sequence subsequently undergoes *spreading*: for every group of 4 bits a 32 bit word (a *chip*) is selected from a standard defined table. This results in a 2 Mchip/s sequence (chip period $T_c = \frac{1}{2\ Mbps} = 500$ ns). By using this spreading process, the rate of the system and hence the occupied bandwidth is increased by the *spreading factor*, M. It represents the number of output bits from the spreading circuitry, for each input bit. In this case for each 4 input bits, a 32 bit long word is obtained, leading to a spreading factor $M = 8$.

The chip sequence represents the input for the O–QPSK modulator. The modulator derives from the chip-sequence $c_0, \dots, c_{2n+1}$ the “in phase” (I) and “in quadrature” (Q) sequences:

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

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, since in a QPSK modulation each symbol can encode two bits. Symbols are then mapped on a constellation as represented in Figure 2.9, that is

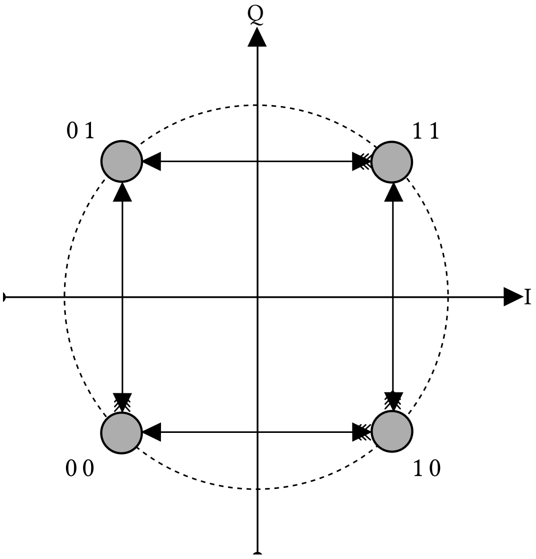


Figure 2.9 – O–QPSK constellation scheme in PHY layer of the IEEE 802.15.4 standard.

Gray coded having one bit changing during a symbol transition to one adjacent. 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 2.8. The filter shape is chosen in order to adapt the signal to the wanted bandwidth. Phase jump of ±π are avoided through the use of a Offset QPSK. In practice, looking at

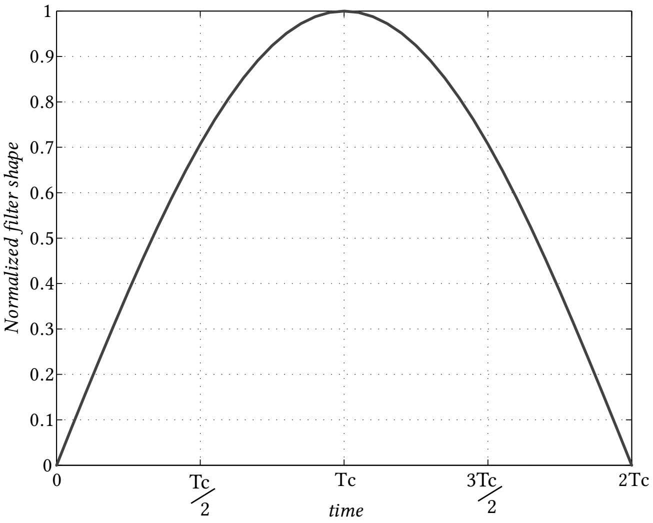


Figure 2.10 – Normalized modulation filter shape.

Fig. 2.11 the in-quadrature sequence is delayed of a chip period so that the two sequence will never change at the same time (see the modulator in Fig. 2.8 on page 26). Transition between two symbols can only occur on the sides of the rectangle, never on diagonals, so only jumps of $\pm\frac{\pi}{2}$ are possible. Avoiding phase jumps of $\pm\pi$ is a good practice since many integrated power amplifier need a constant-power signal, where a sudden change of phase may instead introduce undesired power fluctuations.

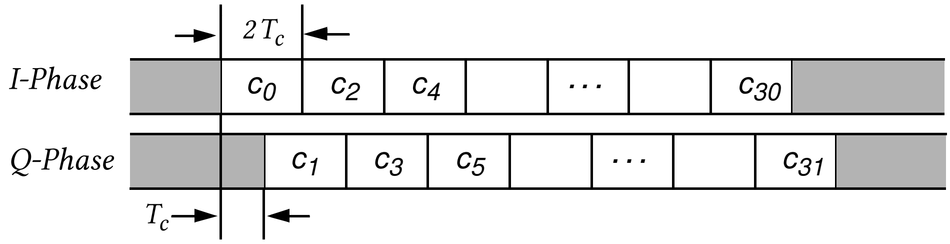


Figure 2.11 – In-phase and Quadrature components of a O–QPSK modulator

Therefore, in order to further limit the power fluctuations, the filter shape can be chosen so that symbol transitions can happen only on the circle. This can be done using a half-sine filter shape 2.10. It is described by the following closed form formula:

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

It is straightforward demonstrating that the power $P(t)$ is always constant (apart from transients). An example of the modulated signals is given in Fig. 2.12 on the facing page.

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

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

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

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

The half-sine filter is a time limited pulse, and this poses a drawback in the frequency domain. Looking at the Fourier Transform of the filter impulse response, calculated in closed form, one may easily see that the spectrum occupation is quite large. Equation

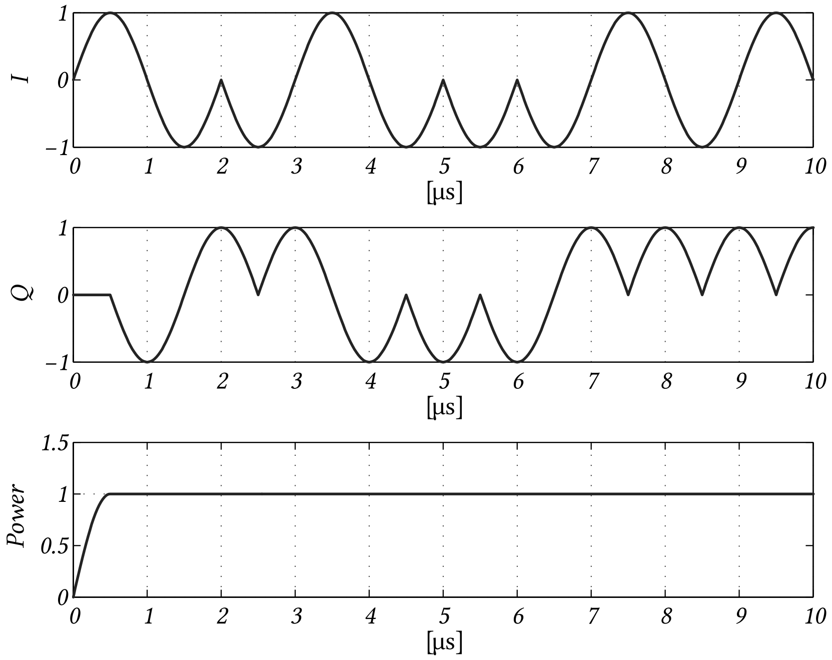


Figure 2.12 – Simulation of the modulator output signals.

Eq. (2.6) describes the spectrum[3].

$$
\begin{aligned}
S(f) \quad &= \quad \frac{2Tc}{2j}\left(sinc\left(2T_c\left(f-f_0\right)\right) e^{-j2\pi(f-f_0)T_c} - sinc\left(2T_c\left(f+f_0\right)\right) e^{-j2\pi(f+f_0)T_c}\right) \\
& \quad\; with \quad f_0 = \frac{1}{4T_c} = 0.5\ MHz
\end{aligned}
\tag{2.6}
$$

The relative baseband power spectrum mask, defined as $|S(f)|^2$, is depicted in Fig. 2.13 on the next page. The center frequency corresponds to the chosen carrier frequency. It is worth noting that the main lobe is bounded in the range $[-1.5\ MHz, 1.5\ MHz]$ and the secondary lobes are attenuated of 23 dB respect to the carrier frequency and centered nearly at $\pm 2\ MHz$ respect to the carrier frequency. The other lobes are 1 MHz apart. At the adjacent channel center frequency ($\Delta f = 5\ MHz$) 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.

[3] The *sinc* function is defined as $sinc(a) = \frac{sin(\pi a)}{\pi a}$

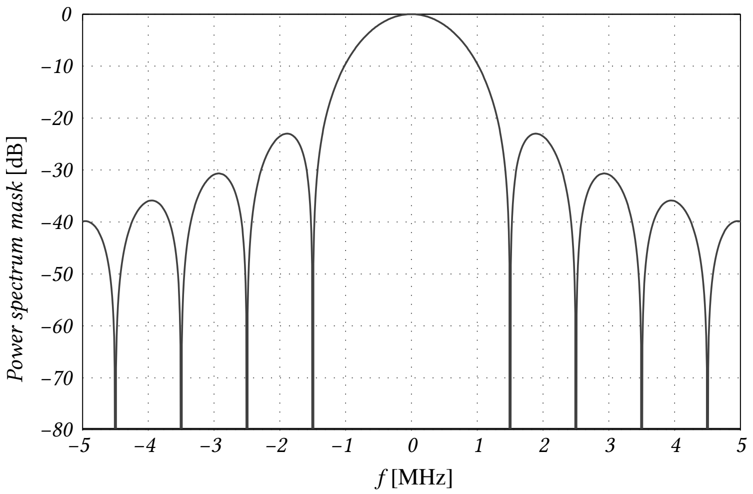


Figure 2.13 – Normalized spectrum mask of a half-sine filter.

2.2.3 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 2.14, different standards operating in the 2.4 GHz ISM band are compared. 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 [26] 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 [26].

2.2.4 Packet Error Calculation for IEEE 802.15.4

The PHY layer in the 2.4 GHz ISM band specifies a data rate of 250 kbit/s, and an O–QPSK modulation is adopted. Each set of four data bits is used to map a symbol to a predefined sequence of pseudo-random noise 32 bits long. These are then concatenated to obtain the actual data stream to be modulated on the carrier.

For the sake of conciseness, the full derivation of the bit error rate in the case of this standard is omitted here, since it is also specified with high details in the standard document [31, Annex E]. Here, it is provided only the final expression for the probability $P_b(\sigma)$

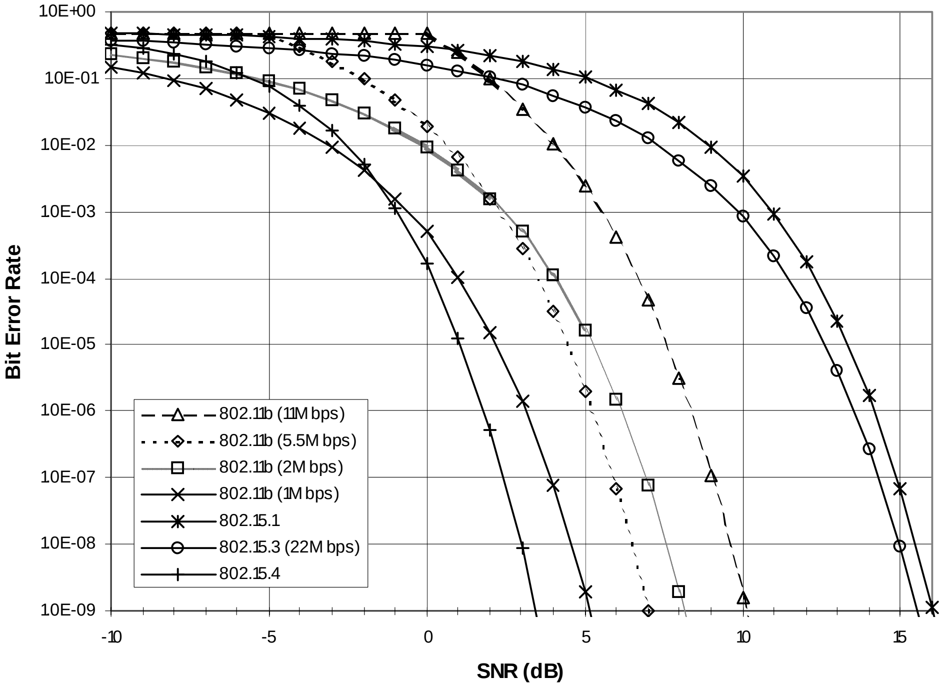


Figure 2.14 – BER performance of various standards operating in the 2.4 GHz ISM band.

modeling the Bit Error Rate (BER):

$$P_b(\sigma) = \frac{8}{15} \cdot \frac{1}{16} \cdot \sum_{n=2}^{16} (-1)^n \binom{16}{n} e^{20 \cdot \sigma \cdot \left(\frac{1}{n} - 1\right)} \tag{2.7}$$

Following the guidelines found in the IEEE 802.15.2 specifications [30], and considering a frame whose length is equal to l bytes, one easily obtain that the packet error probability $P_e(\sigma)$:

$$P_e(l, \sigma) = 1 - (1 - P_b(\sigma))^{l \cdot 8} \tag{2.8}$$

The availability of a closed form solution for the calculation of the BER, and consequently the PER, makes straightforward its implementation in a numerical simulator.

However, for reasons of computational efficiency an original approximation can be implemented for the PER estimation. It is based on the use of an approximating formula for Eq. (2.8), namely:

$$\hat{P}_e(l, \sigma) = \frac{1}{1 + e^{a(\sigma - b)}} \tag{2.9}$$

The dependence of Eq. (2.9) on the total packet length is found in the coefficients a and b, which therefore will be functions of l. They are used to shift and scale the fitting function accurately around the theoretical expression Eq. (2.8).

It is easy to numerically obtain the relationship between a, b and l. In particular they have been evaluated for different values of l between 22 and 125 bytes. To let the simulation easily adapt the model to a specific packet length, also these latter curves have been accurately fitted by using cubic functions, namely:

$$a = 9.9 \cdot 10^{-10} \cdot l^3 - 2.5 \cdot 10^{-6} \cdot l^2 + 0.0025 \cdot l + 1.7 \tag{2.10}$$

$$b = 2 \cdot 10^{-9} \cdot l^3 - 4.9 \cdot 10^{-6} \cdot l^2 + 0.0046 \cdot l - 2.4 \tag{2.11}$$

In Fig. 2.15 the behavior of a and b with respect to the length of the packet is shown.

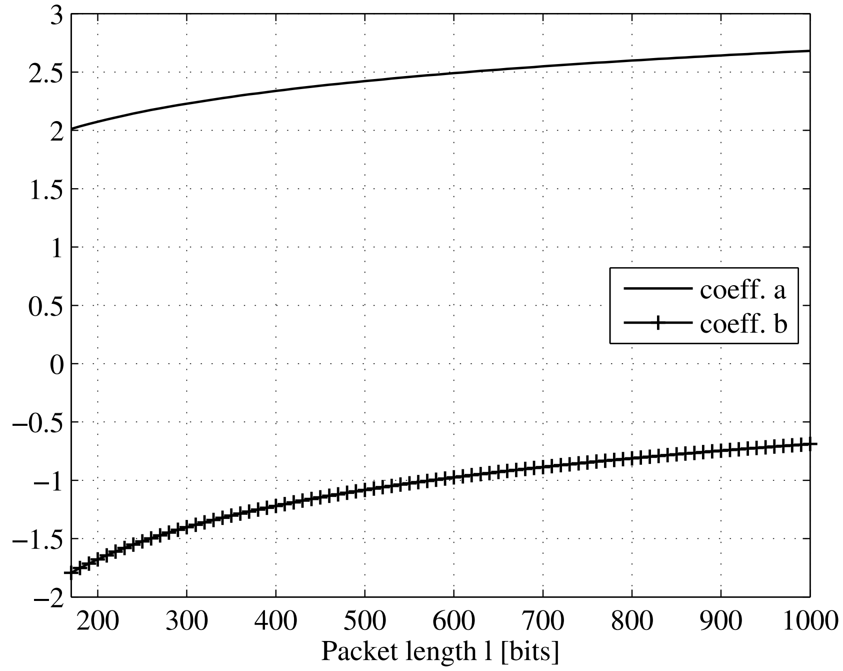


Figure 2.15 – Trend of the coefficients a and b with respect to the packet length l (in bits).

Finally, a graphical comparison of the performances of the PER estimation obtained through the theoretical expression (2.8) and the proposed model (2.9) is shown in Fig. 2.16. Indeed, approximated values provided by Eq. (2.9) differ from theoretical ones (2.8) less than 0.7% RMSE, a value that has been verified to imply negligible effects. Since during a simulation the packet error probability should be calculated each time a packet is received, and the computational efficiency of (2.9) with respect to (2.7)-(2.8) is approximately one magnitude order lower, the overall simulation efforts are considerably reduced.

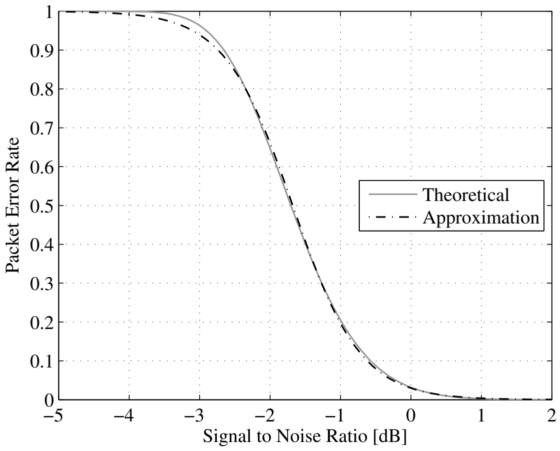


Figure 2.16 – PER estimation obtained through the theoretical model (Eq. (2.7)-(2.8)), compared with the proposed model of Eq. (2.9). A packet length of 200 bits has been considered.

Chapter 3

Introduction to the IEEE 802.11 standard

The standard IEEE 802.11 defines the MAC, and several PHYs layers, for high throughput radio communications, for wireless connectivity for fixed, portable, and moving stations (STAs) within a local area. Several amendments to this standard have been released by the IEEE 802.11 task group, each time providing new features, increasing the transmission bitrate introducing new modulation schemes, etc. A significant example is the [32, Amendment 4], which introduced new modulations and data rate up to 54 Mbit/s. This is actually know as IEEE 802.11g, and is probably the most widespread version of this standard. Other versions are presently of great interest, such as IEEE 802.11n, some of them supporting transmission speed higher than 400 Mbps.

The wireless medium, and the properties of this particular standard, enable wireless networks composed by fixed, portable and moving stations within a local area. In contrast to the IEEE 802.15.4 standard, thanks to the high transmission-rate provided, devices based on this standard could exploit the features provided by common high level protocols commonly found in wired networks, such as TCP/IP, UDP, etc.

The realization of WLANs is one of the most known employ of this standard, where a classic Local Area Network (LAN) supporting for the mobility of nodes and wireless communications exploited.

A WLAN based on the IEEE 802.11 standard has a cellular architecture, with the covered area subdivided in cells like in common mobile telephony communication. Each cell is called BSS.

3.1 WLAN components

Devices compliant to the standard within a BSS are named Stations (STAs) and are controlled by a CF. Two types of CF are defined:

✓ Distributed Coordination Function (DCF): mandatory for each BSS

✓ Point Coordination Function (PCF): optional, and often not implemented.

A CF is a set of rules to allow the access to the medium and to realize the services defined in the standard. The term "Distributed" indicates that the medium access and the data exchange is distributed among all STA, namely, there is no centralized coordinators. Conversely, the PCF function implies that a STA managing services and medium access is present, while the other STAs shall act on the basis of the instructions delivered by this central station. This topic will be more comprehensively covered in a following section describing the MAC layer.

Two or more BSS linked together with a Distribution System (DS), typically an Ethernet LAN, realize a Extended Service Set (ESS). Usually a BSS, which represents a cell in the WLAN terminology, is intended as a set of stations connected to an AP, which is connected to a DS. An AP is a STA that, in addiction to acting as a station, provides access to the DS. Furthermore, this standard also defines a topology called *ad-hoc* network, or Independent BSS (IBSS), formed by two or more stations connected together without any AP. A sketch of these network topologies is depicted here given:

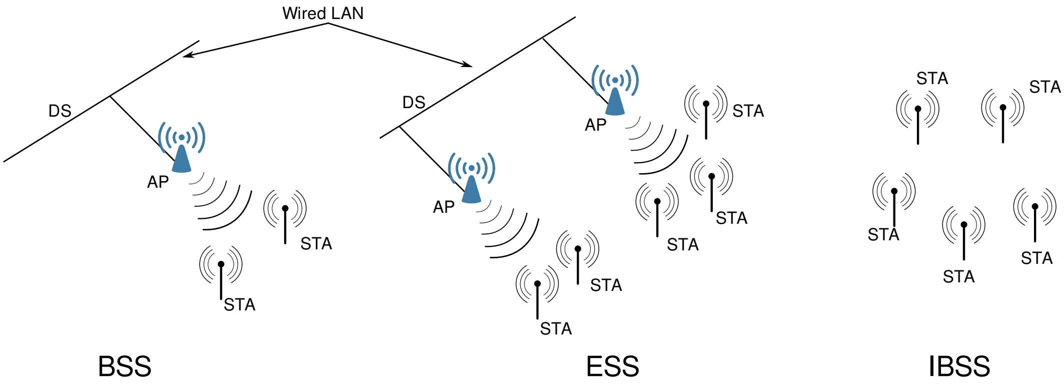


Figure 3.1 – The three possible topologies defined in th IEEE 802.11 standard.

The DS and the BSS, from a logical viewpoint, use two different access media (in the terms of the standard, the DS relies on the Distribution System Medium (DSM) while a BSS employs the Wireless Medium). The standard definitions neither preclude, nor demand, that the multiple media be either the same or different. Stations provide two types of services: the Station Services (SS) and the Distribution System Services (DSS). The latter services are provided by an AP, and they allow traffic forward at MAC level between two stations whose coverage area does not intersect.

Another device type specified by the IEEE 802.11 standard is the *portal*. This device allows the connection between a WLAN and another IEEE 802.x network.

3.2 MAC layer

The Medium Access Control is the most critical layer in the ISO/OSI stack in the description of a wireless communication standard, since it defines the provided services, the devices behavior and determines the performances of the network. The complexity of this level is exacerbated in the IEEE 802.11 standard due to the number of physical layers provided, the number of amendments delivered from the original version, the number of different scenarios which the standard covers, etc.

An example comes from one of the fundamental goal of this standard [32, Sec. 5.1.1.4]: "IEEE 802.11 is required to appear to higher layers (Link Layer Control (LLC)) as a wired IEEE 802 LAN". This requirement imposes that the mobility of nodes is reliably managed at the MAC level, and this is not common in the specification of a MAC layer. Further complications could come from the Quality of Service (QoS) requirements, which however are needed in order to fulfill the goal of providing a WLAN. This level also defines how to manage the connections among stations, the algorithms for accessing the wireless medium, the security related issues of the transmission. It provides the support for frames fragmentation, for carrier sensing, etc. Therefore the present section will not describe comprehensively the IEEE 802.11 MAC layer, focusing instead on those parts of interest for this specific work.

For the scope of the present work, the main features provided by this standard are the medium access algorithm, which is based on the CSMA/CA probabilistic algorithm, and specifically the access method called Distributed Coordination Function (DCF). Any station compliant to the standard implements DCF, and it is used both in *ad-hoc* and in *infrastructured* network, *i.e.* connected through an AP.

3.2.1 The general MAC architecture

The IEEE 802.11 MAC architecture is described in Fig. 3.2 on the following page. This layer supplies three different functions, namely the Distributed Coordination Function (DCF), the Point Coordination Function (PCF) and the Hybrid CF (HCF). The former serves as the basis for both PCF and HCF, and provides contention services. The DCF provides the common access method known as CSMA/CA, and its implementation is mandatory for any compliant device.

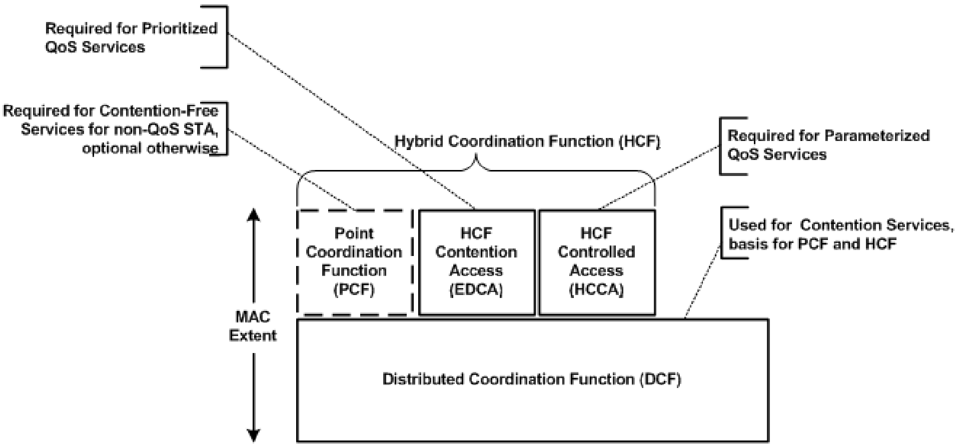

Required for Prioritized QoS Services
Required for Contention-Free Services for non-QoS STA, optional otherwise
Hybrid Coordination Function (HCF)
Required for Parameterized QoS Services
Point Coordination Function (PCF)
HCF Contention Access (EDCA)
HCF Controlled Access (HCCA)
Used for Contention Services, basis for PCF and HCF
MAC Extent
Distributed Coordination Function (DCF)

Figure 3.2 – MAC architecture

Point Coordination Function (PCF)

Instead, Point Coordination Function represents an optional medium access tachnique. It can only be used in infrastructured network, that is, only if there is an AP. In particular, it is the AP that realizes a Point Coordinator (PC), a logical unit that provides Contention Free access to the channel. Since this function is based on the DCF, all the stations in a network regulated under the PCF are inherently able to obey PCF rules, and therefore regular stations do not require particular efforts to operate in this particular type of network. If a BSS use the PCF method, then the two methods alternate realizing a CFP and a Contention Period (CP).

Synthetically, the PC continuously executes a polling among all STAs connected, to determine which is the station allowed to transmit. Anyway, it is optional for a STA to accept operating in PCF mode, *i.e.* a station can ignore poll requests from the PC. A station accepting poll request is called *CF-Pollable*.

In order to provide a contention free access to the channel, the PCF uses a virtual-carrier sense function, *i.e.* get informed of channel occupation through the *Duration* field of frames, and also enables priority rules. In the ideal case, stations are regulated in order to find the channel free any time they try to access it, since the medium is fully regulated by the AP. However, the standard does not provide explanation for the case of two BSS, in which PCF is employed, are in the same area and on the same channel. For example, networks based on the PCF could suffer for the *hidden node* problem. In this case supplementary coordination functions shall be used to avoid collisions.

Hybrid CF (HCF)

The Hybrid CF (HCF), introduced with the IEEE 802.11e version of the standard [32, Amendment 8], enhances the previous coordinator functions, introducing two new ones: the EDCA and the HCF Controlled Channel Access (HCCA). In particular, this version aims at introducing QoS enhancements, for example for the transport of video, voice and audio over IEEE 802.11-based WLANs.

In EDCA the concept of prioritization of traffic is introduced. This is accomplished mainly by implement shorter IFS (namely, Arbitration IFS (AIFS)) and shorter Contention Window (CW). In practice, a STA with high priority traffic has to wait a little less than another STA with lower priority traffic. Moreover, a contention-free access to the wireless channel is provided, defining the Transmit Opportunity (TXOP) periods, during which a STA has unlimited access to the channel and is able to send as many frames as possible, limited only by the duration of the TXOP. A TXOP is in practice a bounded periods of time which is assigned to a particular class of traffic/priority. Priority levels, in the terminology of the standard, are called Access Categories (ACs). The following table resumes the main parameters:

Table 3.1 – EDCA parameters

Access Category	**AIFS**	Maximum TXOP	CWmin	CWmax
Background (AC_BK)	7	0	31	1023
Best Effort (AC_BE)	3	0	31	1023
Video (AC_VI)	2	3.008 ms	15	31
Voice (AC_VO)	2	1.504 ms	7	15

Table 3.2 – Mapping among access categories and IEEE 802.1X priority levels

AC	**IEEE 802.1D User Priority**		Designation
AC_BK	1	Background	Background
AC_BK	2	–	Background
AC_BE	0	Best Effort	Best Effort
AC_BE	3	Excellent effort	Best Effort
AC_VI	4	Controlled Load	Video
AC_VI	5	Video <100 ms delay	Video
AC_VO	6	Video <10 ms delay	Voice
AC_VO	7	Network control	Voice

Obviously, since it is expected that frames are generated by higher level applications, it is also defined a fixed mapping among the IEEE 802.11e access categories and the priority levels defined in IEEE 802.1D. In the Ethernet terminology these are called class of service (CoS). The defined map is shown in Table 3.2 on the previous page.

In the HCCA method, the network behaves similarly to the PCF mode. There is now a Hybrid Coordinator (HC), which is again provided by an AP. A key difference is that a contention free period (CFP) can be initiated at any time between two consecutive beacons, meaning that the fixed cyclic scheme CP+CFP can be broken, and many CFP could exist in a period. During the contention period the STAs follow the EDCA rules. Similarly to EDCA, Traffic Classes (TC) are defined, as well as the Traffic Stream (TS) concept is introduced. The HC is therefore able to consider the priority for each node, but also to define sessions and manage priority for each session, giving a very precise control on the traffic flow. Moreover, each STA can communicate the lengths of its queues, and the HC is therefore enabled to use this information in order to choose the polling order for nodes.

This is the most complex coordination function defined by the IEEE 802.11. While it is common to find AP which support the EDCA features, often declared as IEEE 802.11e support, or Wireless Multimedia (WMM), very few devices implements the HCCA function, due to its high degree of complexity.

In the present work, only the basic Distributed Coordination Function (DCF) has been considered, since it is the most widespread coordination function available, and represents the basis for any other function.

3.2.2 Distributed Coordination Function

This medium access algorithm, basically, specifies that a station wishing to transmit a message has preliminarily to sense the channel for a specific amount of time, called Distributed Coordination Function DIFS (DIFS). If the medium remains free for all the duration of a DIFS, then the station is allowed to access the channel and starts its transmission. Conversely, the STA waits until the transmission stops and then calculates a random backoff time that has to be waited before the station could try to access the medium again. This prevents multiple stations from gaining access to the medium immediately after the completion of the preceding transmission.

The DCF procedure requires ACK packets to be used, *i.e.* each transmission is acknowledged by the receiver. The period between the completion of packet transmission and the start of the ACK frame is called SIFS. ACK frames have a higher priority than other traffic. Transmissions other than ACKs, indeed, must wait at least one DIFS before trying to access the channel.

The channel sense can be performed both through "virtual" mechanisms and physical ones. The latter methods exploit services provided by the PHY layer. The former are based on MAC functions, which rely on an exchange of information about the traffic, through which a prediction of the future traffic behavior is made. Specifically, the virtual carrier sense mechanism is referred to as the Network Allocation Vector (NAV), which stores the information announced through the use of RTS/CTS frames. A Request To Send (RTS) frame is sent by a potential transmitter to the receiver and a Clear To Send (CTS) is sent from the receiver in response to the received RTS frame. If the CTS frame is not received within a certain time interval the RTS frame is retransmitted by executing a backoff algorithm. After a successful exchange of the RTS and CTS frames the data frame can be sent after waiting for a SIFS. RTS and CTS include a duration field that specifies the time interval necessary to transmit the data frame and the ACK. This information is used by stations which can hear the transmitter or the receiver to update their NAV. The NAV is substantially a timer (*i.e.* counter). It decreases to zero at a uniform rate. therefore, when the timer reach zero, the medium is indicated as idle by the virtual carrier sense mechanism. If it indicates a nonzero value, then the indication is busy.

However, the RTS/CTS mechanism is optional, in the sense that a size threshold is specified and only frames larger than that threshold are sent using this technique. The threshold can be configured on a per-station basis.

Often, especially if industrial traffic is under consideration, frame sizes are very limited, typically of some tens to hundreds of Bytes. These sizes are generally significantly lower than the RTS/CTS threshold, so as the basic DCF mechanism is therefore adopted.

In this basic mode, as already said, if the transmitter senses an idle medium, combining the physical and the virtual carrier sense information, for the time of a DIFS, it is allowed to send its frame. Conversely, if it senses the medium as busy, it defers its transmission while continuing the carrier sensing. In particular it has to wait until the current transmission is concluded, and then verifying the channel and check if it remains idle for the duration of a DIFS[1]. Subsequently, it determines a *random backoff period* and set an internal timer, unless the timer already contains a nonzero values, in which case the device do not select any new random time, and start to count time from that value, as will appear clear in the following.

Upon expiration of the DIFS without any other transmission sensed, the timer begins

[1] It is worth to mention that, sometimes, the transmitter has to check if the channel is idle for an Extended IFS (EIFS), and this happens in the case the last frame detected on the medium was not received correctly.

to decrement, counting the expired slot times. During a slot the STA uses the carrier sense functionalities to verify the presence of activity during that slot, and a slot is expired and timer decremented only if no activity is sensed during the slot time. If another transmission occupies the medium, the timer is retained to its actual value, and the backoff procedure suspended. Again, the STA has to wait until the current transmission is finished, sense the channel idle for a DIFS, and can subsequently restart the backoff procedure[2]. If the timer reaches zero, the station may begin transmission.

There is other cases in which the random backoff procedure is adopted. Specifically, a STA which receives correctly a frame, thus receiving an ACK packet from the receiver, shall start a backoff timer just after the end of the ACK frame. In this case, the CW value is set to CW_{min} before choosing the random number, and the retry counters are reset. Moreover, if the STA infers an unsuccessful transmission, it starts a random backoff procedure just after the expiration of the ACKTimeout interval, without resetting the CW interval.

The basic DCF assures a fair access to the channel, trying to minimize collisions, but relies on the underlying assumption that each station can hear all other stations. This is not always the case: let's consider for instance, a station that is able to successfully receive frames from two other transmitters, which however can not receive signals from each other. This is typically known as the *hidden node* problem. In this case a transmitter may sense the medium as being idle even if the other one is transmitting. This results in a collision at the receiving station. An effective solution can be obtained from the use of RTS/CTS frames.

Backoff Time

The backoff time is, in particular, a time interval multiple of a single *slot time*.

$$\text{BackoffTime} = \text{randInt}() \times \text{slotTime} \tag{3.1}$$

A pseudo-random integer number is drawn from a uniform random distribution in the range $[0, CW]$, where CW is an IEEE 802.11 parameter called CW, which is bounded in the range $CW_{min} \leqslant CW \leqslant CW_{max}$. The slotTime is a parameter dependent on the specific physical layer adopted (for example it is equal to 9 µs in the IEEE 802.11g).

Indeed, CW is initially set to a minimum value CW_{min}. Each time a transmission failure is indicated by the PHY layer, it has to update (*i.e.* increment) its value to $2CW + 1$. Thus, a sequence of powers of 2, minus 1, is generated (see Fig. 3.3 on the facing page). The

[2] In short, the backoff procedure can be activated only if the channel is sensed idle for at least a DIFS time.

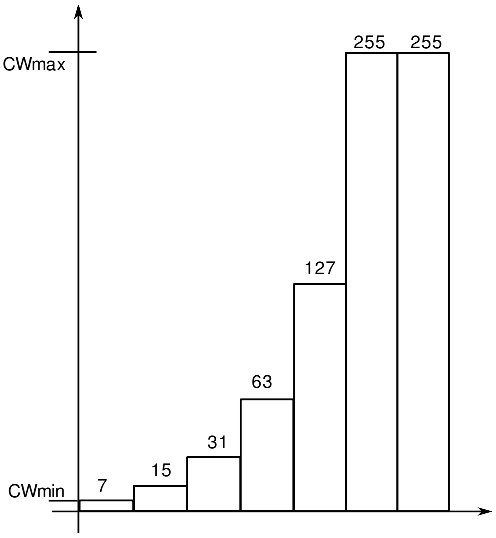


Figure 3.3 – Exponential increase of the CW counter.

value is incremented until the value CW_{max} is reached and, once reached, remains constant until the CW is reset[3]. As a matter of fact, the value of CW is restored to CW_{min} after each successful transmission. It is worth to notice that each STA maintains two counters for retransmission attempts: a STA Short Retry Counter (SSRC) and a STA Long Retry Counter (SLRC), which are updated at each new transmission attempt for a specific frame. In the case of the RTS/CTS mechanism is not used, only the SSRC is of particular interest and updated. These counters are reset to zero at every successful transmission. Finally, the standard specify the default maximum value these counters can reach, bounding the number of transmission retry to 7. Often users of IEEE 802.11 devices are in some way able to change (often only increase) these thresholds.

IFS

In the previous description about DCF, several times predefined interval of time have been defined and adopted. These time intervals between between two consecutive operations are called IFS. The standard defines four different IFS to allow different priority levels to access the medium. The smaller it is the duration of an IFS, the higher the priority of the operation relying on that time is. IFS periods are independent of the STA bit rate. The duration of a particular IFS shall be fixed for each PHY – infrared, direct sequence spread spectrum, frequency hopping spread spectrum, OFDM, etc.

Short IFS (SIFS) This space between frames has to be used for an ACK frame, a CTS

[3] A limit is given in order to protect the mechanism in the case of high-load conditions.

frame, the second or subsequent MPDU of a fragment burst, and by a STA responding to any polling by the PCF. The SIFS is the time from the end of the last symbol of the previous frame to the beginning of the first symbol of the preamble of the subsequent frame as seen at the air interface. This is the shortest IFS length, thus introducing the highest priority for frames.

Point Coordination Function IFS (PIFS) This IFS is used only by stations operating under the PCF to gain access to the medium at the start of the CFP.

Distributed Coordination Function DIFS (DIFS) Stations operating under DCF use this IFS to gain access to the medium to send data frames (MPDU) or management frames.

Extended IFS (EIFS) If a station receives a packet but it is not able to understand the content of that packet, then this IFS has to be used. This station, in fact, being not able to read the duration field of the received packet, can not foresee the occupation of the medium. With this IFS it protects itself against collisions. This is the longest IFS duration.

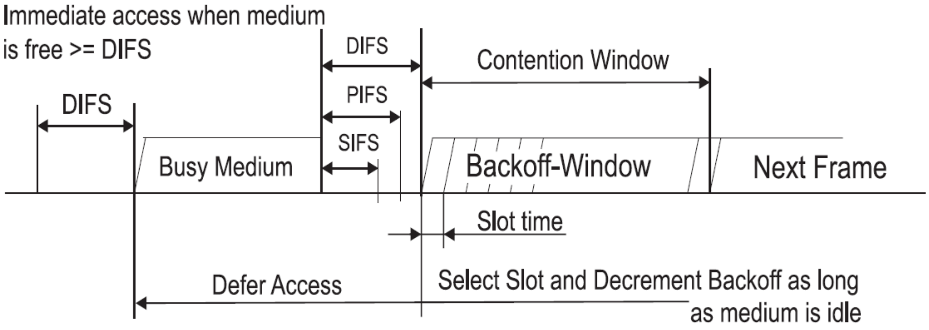

Immediate access when medium is free >= DIFS
DIFS
DIFS
PIFS
SIFS
Contention Window
Busy Medium
Backoff-Window
Next Frame
Slot time
Defer Access
Select Slot and Decrement Backoff as long as medium is idle

Figure 3.4 – The DCF access method.

3.2.3 RTS/CTS access method

In Section 3.2.2 on page 40 I briefly discussed about the *hidden node* problem affecting the DCF access method. The hidden node problem arises when a station A is able to successfully receive frames from two other transmitters (B and C) but the two transmitters can not receive signals from each other. In this case a transmitter (for instance B) may sense the medium as being idle even if the other one is transmitting. This results in a collision at the receiving station A, as depicted in Fig. 3.5 on the facing page.

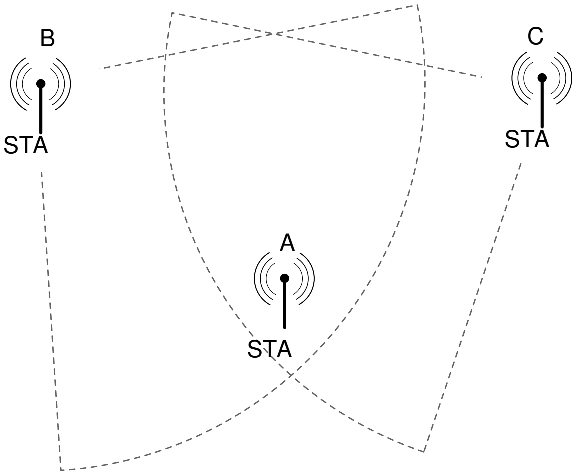


Figure 3.5 – A representation for the *hidden node* problem.

In the figure, the arcs represent the distance/area where a transmitter signal can be heard correctly. Thus, only the central node A can receive (and sense) each transmission, while the two other node are hidden one to the other.

A possible solution to the problem is defined in the the standard and foreseen the use of RTS and CTS frames. If the station B wants to transmit to the station A, the algorithm is the following:

- ✓ Station B shall sends a Request To Send (RTS) frame to A. Every STAs that receive correctly the RTS frame read its "Duration" field and subsequently update their NAV timer to defer their transmission accordingly. Please note that in this case, the station C does not receive the RTS frame.

- ✓ STA A answers with a Clear To Send (CTS) frame to allow the start of the transmission. The CTS frame is sent a SIFS time after the end of the last symbol of the RTS frame. Station C actually receives this CTS frame and reads the enclosed "Duration" field. Indeed, also STA C knows the occurrence of a transmission, even if the source node is not within its covered area.

- ✓ STA B, after the reception of the CTS frame, shall wait a SIFS time and then starts its transmission without sensing the channel.

The drawback of using RTS/CTS is an increased overhead which may reveal inadequate for the transmission short data frames: the efficiency of RTS/CTS algorithm depends upon the length of the packets. It is generally assumed, in fact, that RTS/CTS is used for large-size packets, and a *RTSthreshold* is defined on a per-STA basis so as all frames whose length is above the *RTSThreshold* are transmitted using RTS/CTS. There are two cases that represents a boundary: if the *RTSThreshold* is set to 0 then every packet shall be transmitted using RTS/CTS. Also, if *RTSThreshold* is set to a value above the maximum allowed MPDU length, then this mechanism is disabled at all.

3.2.4 Fragmentation of frames

An important feature of the MAC layer of the IEEE 802.11 standard is the possibility of fragmentation and defragmentation of information units. Therefore, it is possible to subdivide MSDU and MAC Management Protocol Data Unit (MMPDU) into smaller frames called MPDU. This process is executed to achieve better performances in the transmission process, being the transmission of smaller packets more reliable, even in poor characteristics channels. Only *unicast* frames can be fragmented while *broadcast* and *multicast* frames can't.

This procedure, following the rules of the standard, allows the transmitter to wait only a SIFS period between two successive fragment transmission (burst of fragment), as depicted in Fig. 3.6. Therefore during a burst fragment transmission the source device does not have to contend the medium with other devices. Moreover, the transmitter sends the sequence of MPDUs without interruption, until fragments are correctly delivered, preventing other devices to transmit. For the sake of completeness, enhanced versions of the IEEE 802.11 standard, for example the *e* version, provide a "'BlockACK"' feature, in order to limit the time spent in ACK transmissions[4].

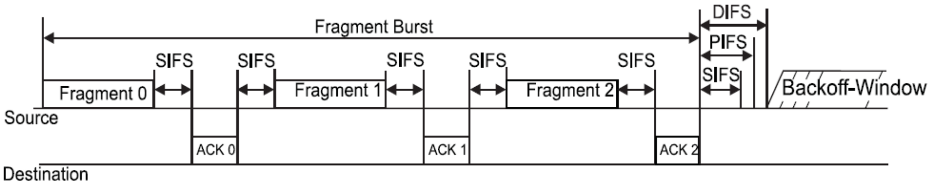


Figure 3.6 – A transmission of a series of fragment. Only a SIFS period is necessary between fragments.

3.2.5 Frame format

The MAC frame is composed by three main fields: the MAC header, the Frame Body and the FCS. The overhead introduced on a DATA frame is 36 bytes, accounting for the MAC

[4] Later on, in this chapter, we will see that, even at 54 Mbit/s transmission rate, the transmission of an ACK require 34 µs.

header and Frame Check Sequence (FCS). Instead, an ACK frame has a predefined and fixed structure 14 bytes long [32].

The meaning of this fields is the same that for the IEEE 802.15.4 standard. The Frame Body contains information as specified in the "frame type" field of the MAC header, so if the *type* is data, then the Frame Body will be the *payload*. The FCS field contains a 32 bit CRC code for error correction.

The general structure is depicted in figure.

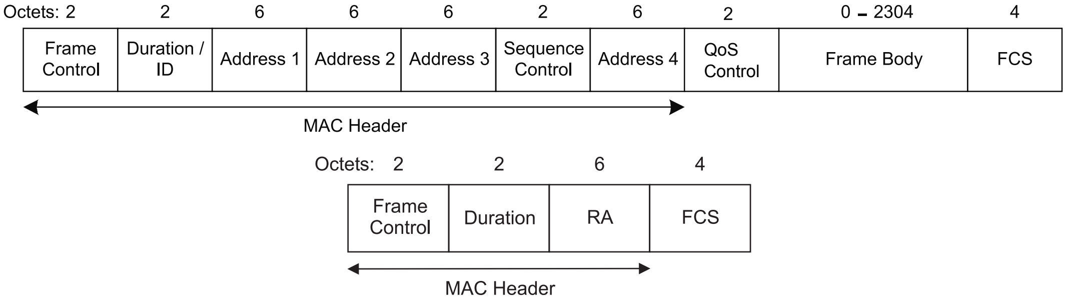


Figure 3.7 – Frame format of a IEEE 802.11 DATA and ACK frame [32]

Except for "Frame Control", "Duration" and "FCS" fields, the others are not always present, on the basis of the specific type of the frame. A brief explanation of the meaning of fields is covered here.

Frame Control

This field is a container for a long set of information. Its composition is depicted in this figure:

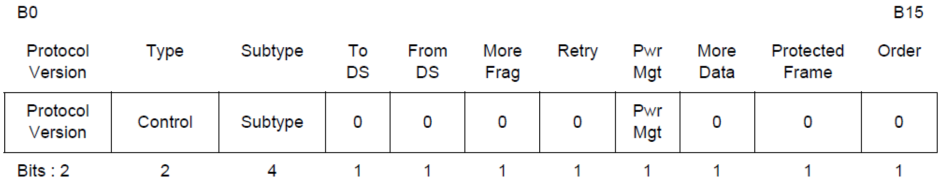


Figure 3.8 – Control field composition.

Type and Subtype The type and subtype fields identify the function of the frame. There are three possible frame type: data, control and management, and many combinations with the subtype field is possible. The following table represents only some of this combination.

Table 3.3 – Some example of combination of fields type and subtype.

Type value b3 b2	Type description	Subtype value b7 b6 b5 b4	Subtype description
00	Management	0000	Association Request
00	Management	0001	Association Response
00	Management	1010	Disassociation
		...	
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK
		...	
10	Data	0000	Data
10	Data	0001	Data+CF ACK
10	Data	0101	CF ACK
		...	

ToDS and FromDS Indicate the direction to or from the Distribution System. Combinations are listed in Table 3.9:

Figure 3.9 – Some example of combination of From/To DS fields.

To/From DS values	Meaning

ToDS=0
FromDS=0
	A frame from one STA to another STA, or management and control frame

ToDS=0
FromDS=1
	Data frame exiting the DS

ToDS=1
FromDS=0
	Data frame directed to DS

ToDS=1
FromDS=1
	Wireless Distribution System frame from one AP to another AP

More Fragment field If its value is 1 then this frame is a fragment of a MSDU. Other fragments will follow. This field is 0 otherwise.

Retry It is 1 only if this frame is a transmission retry of a precedent transmission.

Protected It is 1 if the Frame Body field has been encrypted by some sryptographic algorithm (for example the Wired Equivalent Privacy (WEP) algorithm). Only data frames and authentication frames can have this bit set to 1.

Duration

This is a 16 bit field that indicates the length of a frame. Possible combination are:

Table 3.4 – The encoding of the Duration field.

Bit 15	Bit 14	Bits 13-0	Meaning
0	0-32767		Length of the frame
1	0	0	Fixed value within frames transmitted during the CFP
1	0	1-16383	Reserved
1	1	0	Reserved
1	1	1-2007	AID in PS-Poll frames
1	1	2008-16383	Reserved

Address Fields

There are 4 address fields, each one contain a 48 bit address complying the IEEE Std802-1990. An address can be:

- ✓ *Individual*: it is the address of a STA
- ✓ *Group*: address associated with more or a group of STA:

  Multicast: an address associated by an higher-layer convention with a group of STA.

  Broadcast: an multicast address that is associated with all the STA in a network.

Each field can contain these type of indications: Basic Service Set IDentification (BSSID), Destination Address (DA), Sender Address (SA), Transmitter Address (TA), and Receiver Address (RA). Certain address field usage is specified by the relative position of the address field (1–4) within the MAC header, independent of the type of address present in that field. For example, receiver address matching is always performed on the contents of the Address 1 field in received frames, and the receiver address of CTS and ACK frames is always obtained from the Address 2 field in the corresponding RTS frame, or from the frame being acknowledged.

Sequence Control

This field is 16 bits long, and is composed by 12 bits of Sequence Number and 4 bits for the Fragment Number (FN). Each MSDU or MMPDU has an assigned SN, which is returned by a counter modulo 4096, starting at 0. The sequence number is the same for every retransmission and for the fragments of the same frame. The FN starts at 0 for the first or the only fragment of a MSDU. It is incremented by one for each successive fragment and remains constant for all retransmissions.

Frame Check Sequence

This field contains a 32 bit CRC. The *calculation fields* used to obtain the FCS are the Frame body and the MAC header. The generator polynomial is:

$$\begin{aligned} G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + +x^{12} + x^{11} + x^{10} + x^{8} + \\ +x^{7} + x^{5} + x^{4} + x^{2} + x + 1 \end{aligned} \tag{3.2}$$

3.3 PHY layer

This standard provide different possible implementations for the PHY layer, depending on the underlying medium, and significant example are:

IR Infrared

FHSS Frequency Hopping Spread Spectrum

DSSS Direct–Sequence Spread Spectrum, and its extension High Rate (HR/DSSS)

OFDM Orthogonal Frequency Division Multiplexing (@5 GHz), and the Extended-Rate PHY–Orthogonal Frequency Division Multiplexing (ERP–OFDM) version, in the 2.4 GHz ISM band.

Spread spectrum is ideal for data communications because it is less susceptible to radio noise and creates little interference, it is used to comply with the regulations for use in the ISM band.

The infrared medium is less used and will not be covered here. The same IEEE 802.11 standard, in its present form, states that this physical layer is no longer maintained, and compatibility issues with the advanced features of the standard could arisen.

Using frequency hopping, the 2.4GHz band is divided into 1-MHz-channels. Frequency Hopping Spread Spectrum (FHSS) allows for a less complex radio design than other physical layer but provides a limited transfer rate of 2 Mbit/s. Limitations come from the Federal Communication Commission (FCC) regulations that restrict subchannel bandwith to 1 MHz, causing many hops which means a high amount of hopping overhead. Also this physiscal layer is not covered in this work.

For the purposes of WLANs, the most widespread choices have been the DSSS PHY and the Orthogonal Frequency Division Multiplexing (OFDM) one. In particular, for the sake of clearness, the HR/DSSS is the commonly known IEEE 802.11b [32, Section 18], while the ERP–OFDM is the largely used IEEE 802.11g [32, Section 19].

Transmission Power

For Europe, devices that comply with IEEE 802.11 standard undergo a limit of 100mW Equivalent Isotropically Radiated Power (EIRP). In addition, all conformant Physical Media Dependent (PMD) implementations shall support at least a power level of 100mW EIRP. If a conformant PMD implementation has the ability to transmit in a manner that results in the EIRP of the transmit signal exceeding the level of 100mW, at least one level of transmit power control has to be implemented. This transmit power control shall be such that the level of the emission is reduced to a level at or below 100mW under the influence of said power control.

3.3.1 HR/DSSS PHY

The PHY layer is subdivided in two sublayers:

Physical Layer Convergence Procedure (PLCP) sublayer adapts the capabilities of the PMD system to the PHY layer services. It presents an interface for the MAC layer to write to and provides carrier sense and CCA functionalities.

PMD sublayer defines the methods for transmitting and receiving data through the wireless medium between two or more STAs each one using the same modulation system. It takes care of the wireless encoding.

The DSSS PHY is defined to use the ISM 2.4GHz band. Regulatory bodies in Europe and USA precisely defined this ISM band bounds as 2.4-2.4835 GHz. This PHY divides this band into 14 channels[5]. Channels allocation and carrier frequencies are defined in the following Table 3.10 on the next page

[5] In the US only 11 channels are available.

Figure 3.10 – Channel numbering and center frequency for DSSS.

Channel ID	Center Frequency [MHz]
1	2412
2	2417
3	2422
4	2427
5	2432
6	2437
7	2442
8	2447
9	2452
10	2457
11	2462
12	2467
13	2472
14	2484

Each channel has a bandwidth of 22 MHz (this will be clarified later), and center frequencies are 5 MHz spaced. If in the same area more WLAN will coexist then carriers employed will be spaced by 25 MHz, to limit reciprocal interference. In practice, only 3 channels can exist at the same location.

Modulation and Data rate

The IEEE 802.11 standard defines two different modulation, hence two different data rate, for the simple DSSS PHY: a basic access rate (1 Mbit/s Differential Binary Phase Shift Keying (DBPSK)) and an enhanced access rate (2 Mbit/s Differential Quadrature Phase Shift Keying (DQPSK)). With the High Rate version (IEEE 802.11b) other two data rate have been added, namely 5.5 Mbit/s and 11 Mbit/s. The HR access are based on Complementary Code Keying (CCK) modulation scheme. Optionally, a HR/DSSS/PBCC mode can be implemented, which actually uses a Packet Binary Convolutional Coding (PBCC) modulation. Since this is an optional mode, it will not covered in this bries overview.

Table 3.12 on the facing page and Table 3.13 on page 54 show the mapping between symbol and phase change, thus defining the constellation point chosen by a particular symbol, while the modulations adopted are well-known.

An overview of the most important parameters regarding this PHY layer is given through Table 3.14 on page 54.

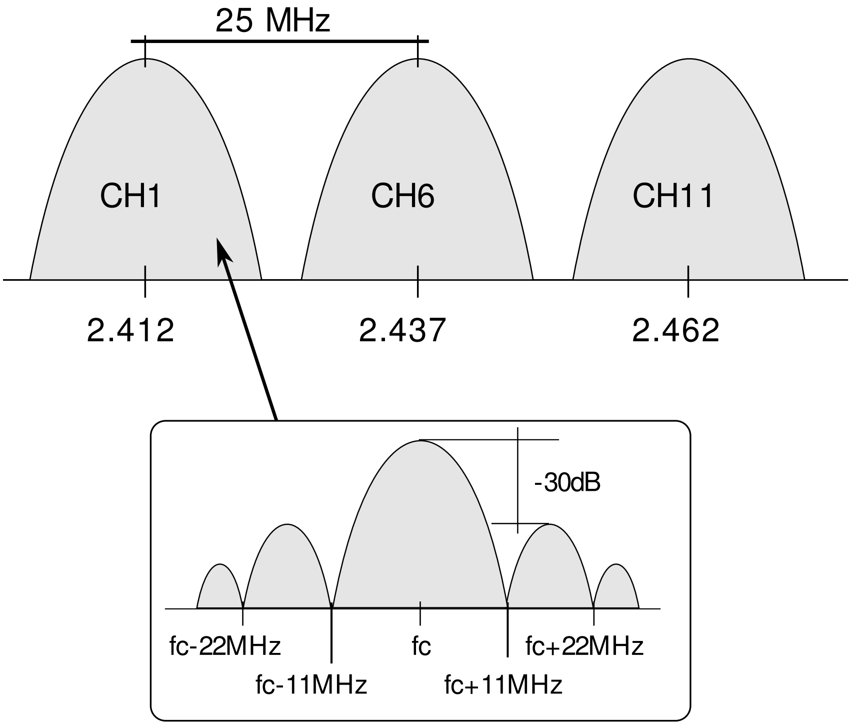


Figure 3.11 – Three DSSS non-overlapping channels.

Figure 3.12 – DBPSK Encoding table.

Bit input	Phase change ($+j\omega$)
0	0
1	π

PPDU format

The PHY layer packet, that is, the packet that will eventually be transmitted by the radio, is composed by a PLCP header, a preamble and the Packet Service Data Unit (PSDU). In practice, the MAC layer generates a MAC packet, called MPDU, starting from the data arrived from the higher levels (the MSDU). The MPDU is passed down to the PHY through the specific PHY service, and its name becomes PSDU, and represents the payload of the PHY packet. Please note that in all the physical layer of interest for this work, only the PSDU is effectively modulated at the higher possible rates, while the preamble and the header are often transmitted at lower rates, in order to exploit the higher robustness of those modulation schemes.

The standard provides a device with the choice between a long and a short preamble and header. The ability to use a short preamble is optional, and its use allows to reach

Figure 3.13 – DQPSK Encoding table.

Bit input	Phase change ($+j\omega$)
00	0
01	$\pi/2$
11	π
10	$3\pi/2$

Figure 3.14 – Some parameters ragerding the HR/DSSS PHY.

Parameter	Value
Slot Time	20 µs
SIFS	10 µs
CW_{min}	10
CW_{max}	1023
Preamble length	{144, 72} µs
PLCPHeader length	{48, 24} µs

a significant increase in the throughput. The use of such a preamble and header breaks the interoperability between the two types of devices. It is however common to use short preambles since nowadays devices commonly support this feature.

In the case of a short preamble and header, the PPDU format is given in Fig. 3.15.

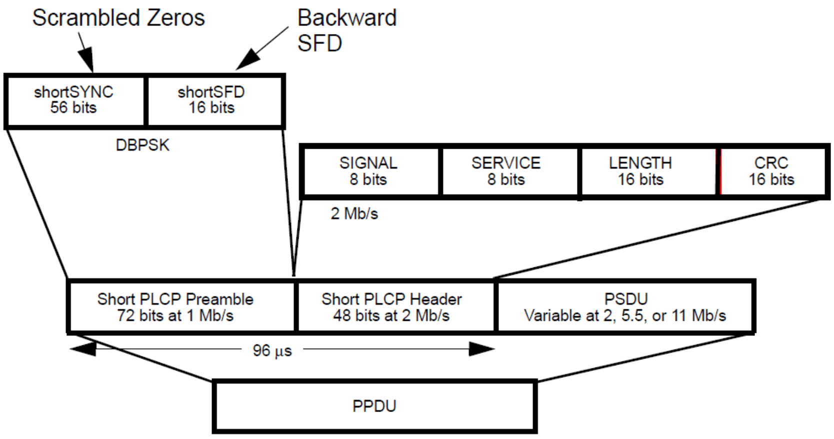


Figure 3.15 – Short preamble PPDU format.

The short PLCP preamble is sent using the DBPSK modulation at a rate of 1 Mbit/s. The Header is sent with a data rate of 2 Mbit/s (DQPSK modulation), while the rest of the packet is sent using one of the possible rates provided by this PHY.

Spreading Sequence

This standard specifies, for 1 and 2 Mbit/s operations, the following 11-chip *Barker* sequence to be used as a Pseudo–Noise (PN) sequence:

$$+1, -1, +1, +1, -1, +1, +1, +1, -1, -1, -1$$

where the first bit is the leftmost one.

For the high data rates, with the CCK modulation, the spreading code length is 8 and is based on complementary codes. The following formula is used to derive the CCK code words that shall be used for spreading both 5.5 Mbit/s and 11 Mbit/s:

$$\begin{aligned} C = \{&e^{j(\phi_1+\phi_2+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_3+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_4)}, \\ &-e^{j(\phi_1+\phi_4)}, e^{j(\phi_1+\phi_2+\phi_3)}, e^{j(\phi_1+\phi_3)}, -e^{j(\phi_1+\phi_2)}, e^{j\phi_1}\} \end{aligned} \tag{3.3}$$

where C is the code word $C = [c_0, \dots, c_7]$ and the ϕ_i terms are defined by the standard document [32, Clause 18] and are here omitted for brevity.

Transmission time

The transmission time can be calculated through the following equation:

$$T_{TX} = T_{preamble} + T_{PLCPHeader} + \left\lceil \frac{(l + PBCC) \times 8}{DataRate} \right\rceil \tag{3.4}$$

where l is the length, in Bytes, of the PSDU (*i.e.* the packet coming from the MAC layer, that represents the payload). *PBCC* is either 1 or 0, respectively in the case of the PBCC modulation is used or not. The other parameters are given in Table 3.14 on the facing page.

Transmit spectrum

The transmitted spectrum is defined with respect to the $\sin(x)/x$ function peak. Since this standard also serves the scope of defining the measurement procedures for verifying the compliance of a device with the regulations themselves, parameters are given that allow designers to prove the effectiveness of their devices with respect to the definitions. In particular, emissions of the radio have to comply with the mask defined in Fig. 3.16 on the next page. In particular, the spectrum should be bounded in:

$$-30dBr \quad \text{for} \quad f_c - 22\text{MHz} \leqslant f \leqslant f_c - 11\text{MHz}, f_c + 11\text{MHz} \leqslant f \leqslant f_c + 22\text{MHz}$$

$$-50dBr \quad \text{for} \quad f < f_c - 22\text{MHz}, f > f_c + 22\text{MHz}$$

where dBr is a measure in dB relative to the peak of the $\sin(x)/x$ function. The measurement has to be done through a spectrum analyzer with a resolution bandwidth of 100 kHz and a video bandwidth of 100 kHz, too.

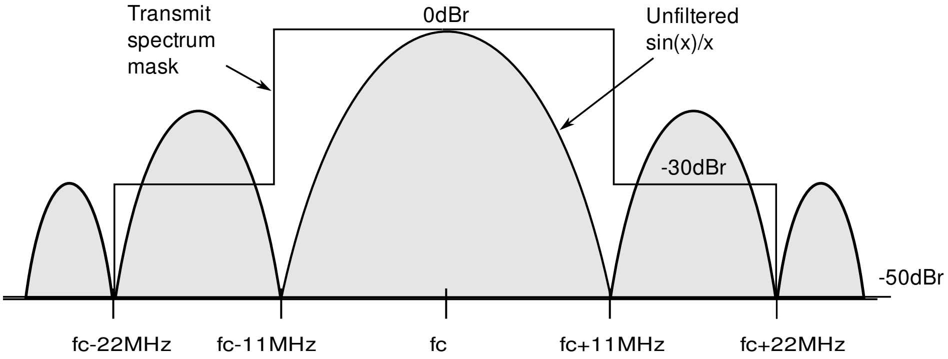


Figure 3.16 – Transmit spectrum mask for DSSS physical layer.

The document also gives very precise methods for the compliance of a device with respect to the modulator performance. In that case the Error Vector Magnitude (EVM) index is used. Many other compliance tests are defined (*e.g.* center frequency tolerance, chip clock frequency tolerance, adjacent channel rejection, etc) but their description wold fall out of the scope of the present work.

Clear Channel Assessment

For HR/DSSS operations the IEEE 802.11 standard specify that a device must be able to perform CCA according to at least one of the following mode:

CCA Mode 1: Energy above threshold It is sufficient that the signal's energy on the interested band is above the ED threshold to sense the channel busy. The threshold is typically -76 dBm, but it depends also on the transmission power.

CCA Mode 4: Carrier Sense with Timer the STA initialize a timer (duration 3.65 ms) and starts decreasing it. If the timer expires without sensing a signal with HR/DSSS characteristics then CCA reports an idle medium. Otherwise it reports a busy medium.

CCA Mode 5 CCA reports a busy medium at least while both the signal is of type HR/DSSS and the energy is above threshold.

It is worth observing that these modes are very similar to the definition of the CCA modes in the IEEE 802.15.4 standard. Indeed, if a simple DSSS compliant device is used, it implements the previous CCa mode 1, and the following

CCA Mode 2: Carrier Sense If a DSSS signal, above or below the ED threshold, is received then the channel is sensed busy, without care on 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.

which are exactly the same definitions given by the IEEE 802.15.4 standard.

3.3.2 ERP–OFDM PHY

Many of the technical aspects regarding this PHY layer present several similarities with those detailed in the previous Section 3.3.1 on page 51. This physical layer is described in [32, Clause 19] of the IEEE 802.11 standard.

This enhanced version of PHY layer, commonly known as IEEE 802.11g, exploits the potential of the OFDM modulation in order to provide high transmission rates, from 6 to 54 Mbit/s. This has broadened the use of IEEE 802.11 devices, which nowadays are implemented in almost any portable device, from computers to smartphones and televisions.

Table 3.5 on the next page lists the modulations and parameters adopted by this PHY. A compliant device must implement at least the transmission rates 6, 12, and 24 Mbit/s. This ensures that devices compliant to this PHY are in any case able to talk to each other.

PPDU format

The Extended-Rate PHY (ERP) of the IEEE 802.11 standard provides several options for the physical layer implementation, often given to assure interoperability between this version and the IEEE 802.11b version (HR/DSSS described in Section 3.3.1 on page 51). The PPDU format divided in three types: one implementing a long preamble, to be compatible with the HR/DSSS PHY using long preamble; a short preamble based PHY, again to give backward compatibility with the HR/DSSS PHY using short preambles; and the "new" ERP–OFDM preamble.

The first two preambles are introduced for backward compatibility for devices based on the IEEE 802.11b, to provide a WLAN communicate properly also in the case of presence of older devices. Choosing this two preamble types, and consequently the PPDU format, induces the use of the rates {1, 2, 5.5, 11} Mbit/s.

Table 3.5 – IEEE 802.11g Physical Layer Parameters

Mode	Data Rate [Mbit/s]	Modulation	Code rate	N_{BPSd}	Ack Rate [Mbit/s]	N_{BPSa}
1*	6	BPSK	1/2	3	6	3
2	9	BPSK	3/4	4.5	6	3
3*	12	QPSK	1/2	6	12	6
4	18	QPSK	3/4	9	12	6
5*	24	16-QAM	1/2	12	24	12
6	36	16-QAM	3/4	18	24	12
7	48	64-QAM	2/3	24	24	12
8	54	64-QAM	3/4	27	24	12

Rate marked with * are mandatory

In the case of the ERP–OFDM format is chosen, then starting from the incoming MPDU several parts are added to form the final PPDU. First of all, a PLCP fixed preamble is inserted, which is composed by repetitions of short and a long training sequences used mainly for timing and frequency acquisitions, diversity selection, etc. Then the PLCP header is added. The content is clearly stated by the figure, and is derived from the information passed through PHY service calls. The `RATE` and `LENGTH` fields are subsequently encoded (convolutional coded, rate 1/2), mapped on a Binary Phase Shift-Keying (BPSK) modulation and one OFDM symbol, as described in the next paragraph. The SERVICE field, together with the PSDU, 6 tail bits (all zeros) and eventually some padding bits forms the part of the final PPDU which will be transmitted at the chosen (high) rate. IT will be encoded, modulated and mapped on OFDM symbols and finally on the 48 OFDM subcarriers. A sketch of the PPDU frame format is given in Fig. 3.17.

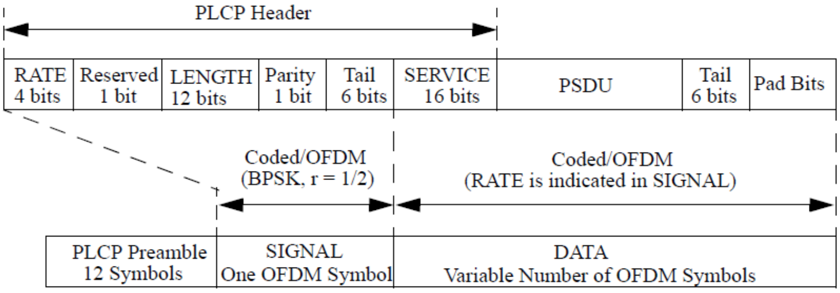


Figure 3.17 – General PPDU format for the ERP–OFDM PHY.

Modulation, generation of a PPDU

The PPDU general format is given in Fig. 3.17 on the facing page.

A comprehensive discussion about the modulation techniques implemented by this standard, while being very interesting, is out of the scope of the present work. In the following text, only the principal arguments of interest for the studies presented through this work, and for the implementation of a network simulator compliant to this standard will be presented.

Through this PHY layer, a system able to transmit payloads at {6, 9, 12, 18, 24, 36, 48, 54} Mbit/s. The OFDM system adopts 52 sub-carriers[6] These are modulated using PSK modulations for lower rates, or Quadrature Amplitude Modulation (QAM) modulations for the higher ones. Table 3.5 on the preceding page gives the correct parameters for the modulations. Moreover, a FEC technique is adopted, exploiting a convolutional coding with rates 1/2, 2/3 or 3/4. The support for 6, 12 and 24 Mbit/s is mandatory for devices compliant to this version of the standard.

Actually, the modulation and rates provided are the same of the IEEE 802.11a specifications, which introduced the operations of WLANs in the 5 GHz band. The present PHY borrows many of those specifications, and often in the standard document references to those paragraphs are found.

The process of creation of a new PPDU is composed of several steps. Considering the structure of Fig. 3.17 on the facing page has already been reached, the following steps are, summarizing:

- ✓ scramble the PPDU. The six zero bits at the end of the packet, once scrambled, must be replaced with six zero bits (unscrambled)
- ✓ encode the frame through the convolutional encoder, rate 1/2.
- ✓ puncture (omit) some bits of the encoder output to reach the correct coding rate, *i.e.* the specified data rate.
- ✓ group the bit stream into N bit-long groups and reorder bits following the algorithm defined in the standard. This is called *interleaving*.
- ✓ re-group bits in N bit-long groups and convert these into complex number according to the modulation scheme adopted.

[6] In the actual implementation, only 48 out of 52 subcarriers are used for data, while 4 subcarriers are "guard" subcarriers.

- ✓ group these complex number stream into groups of 48 numbers. These will be mapped on 48 OFDM subcarriers.
- ✓ four subcarriers (-21, -7, 7 and 21) are inserted as pilot subcarriers.
- ✓ calculate the Inverse Fast-Fourier Transform (IFFT) of the 52 subcarriers obtaining the time domain representation of the signal.
- ✓ append each symbol one after another starting from the leftmost of Fig. 3.17 on page 58 and transmit on the channel (upconversion to the specific RF frequency).

Various passages are therefore required in order to obtain the final signal to be transmitted on the wireless channel. The process requires also a good computational effort. For this reason, the original IEEE 802.11a PHY specifications (which are older than these ones) uses a SIFS of 16 µs, to the scope of accommodating more time for the convolutional code to finish decoding the frame. In this clause, however, the SIFS time is 10 µs. Therefore, at the end of each transmitted PPDU a period of no transmission called *signal extension* is added. This is actually a 6 µstime interval which ensures a proper interoperability between standards and correct decoding of frames.

The convolutional encoder uses the generator polynomials $g_0 = 133_8$ and $g_1 = 171_8$. Its constraint length is $K = 7$.

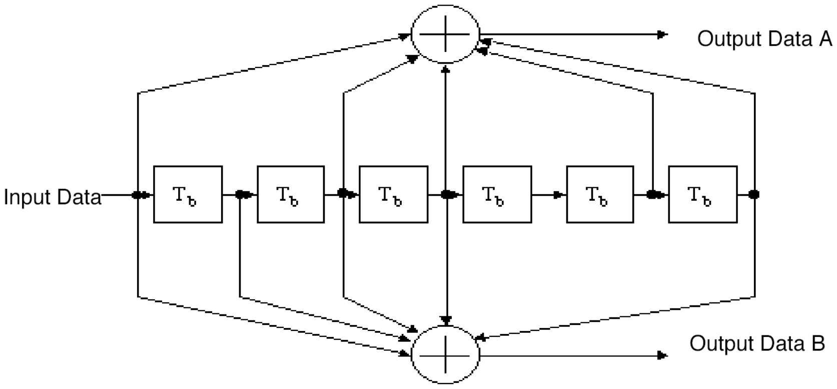


Figure 3.18 – The convolutional encoder used in this PHY

To obtain the higher rates, puncturing is adopted, following the puncturing pattern detailed below:

$$\text{Encoded data rate } 1/2 \longrightarrow \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \longrightarrow \text{Encoded data rate } 2/3$$

$$\text{Encoded data rate } 1/2 \longrightarrow \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \longrightarrow \text{Encoded data rate } 3/4$$

where a 0 in the matrices indicates a bit of the bit stream that is omitted. At the receiver side, in that position a 0 “dummy” bit will be inserted. Therefore, starting from the basic FEC which for each bit generates a couple of bits, to reach higher data rates the puncturer omits transmitting some bits, therefore using a redundancy of 3 bits for 2 original bits, or 4 bits for 3 original bits.

An overview of the most important parameters regarding this PHY layer is given through Table 3.19.

Figure 3.19 – Some parameters ragerding the HR/DSSS PHY.

Parameter	Value
Slot Time	9 µs
SIFS	10 µs
CW_{min}	{15,31}
CW_{max}	1023
Preamble length	16 µs
PLCPHeader length	4 µs

Transmission Time

According to the standard, the time to transmit a DATA (T_{data}) and an ACK (T_{ack}) frame are given by, respectively:

$$T_{data}(l, m) = 20 + 4 \cdot \left\lceil \frac{36 + l + 22/8}{N_{BPSd}(m)} \right\rceil + 6 \quad \mu\text{s} \tag{3.5}$$

$$T_{ack}(m) = 20 + 4 \cdot \left\lceil \frac{14 + 22/8}{N_{BPSa}(m)} \right\rceil + 6 \quad \mu\text{s} \tag{3.6}$$

where l is the payload size in bytes, m is the PHY mode, and N_{BPSd}, N_{BPSa} indicates the number of bytes per symbol, as specified in Table 3.5 on page 58. In Eq. (3.5) and Eq. (3.6), 20 µs accounts for the time necessary for the transmission of both the preamble and the physical header, whereas 6 µs represents the signal extension added by the standard for compatibility purposes. The symbol transmission time is 4 µs. Only an integer number of symbol is allowed. The ceiling function $\lceil\cdot\rceil$ hence models the modulator behavior, which insert padding bits at the end of the PSDU part of the frame. In Eq. (3.5) and (3.6) the numerator represents the amount of data to be transmitted, accounting for the actual data length l, the 36 bytes of MAC overhead and 22 bits of PHY overhead.

In Table 3.5 on page 58, the two rightmost columns are relevant to the transmission of acknowledgment frames. In particular, the standard [32] indicates that each *control* frame has to be transmitted at one of the rates in the basic rate set, which is the set of rates that each host in the network has to support. In our simulations and experiments, we assumed that this corresponds to the set of mandatory rates (*i.e.* 6, 12 and 24 Mbit/s). Moreover, it is foreseen that the acknowledgment frame has to be transmitted at the highest rate in the basic rate set that is less than or equal to the rate of the corresponding data frame (*e.g.* for a data frame sent at the maximum rate, the ACK packet is transmitted at 24 Mbit/s).

In order to experimentally verify the aforementioned equations, a simple self-developed traffic generator that periodically exchanges packets between two STAs has been used, and with a real-time spectrum analyzer the wireless channel has been observed. For instance, sending 46 Bytes-long packets (PSDU length, l in Eq. (3.5)), at the higher rate (*i.e.* $m = 8$), Eq. (3.5) and Eq. (3.6) return a transmission time equal to 42 µsand 34 µsfor data and acknowledgement packet, respectively. A portion of the channel power, captured by using the spectrum analyzer in *zero-span* mode, is reported in Fig. 3.20.

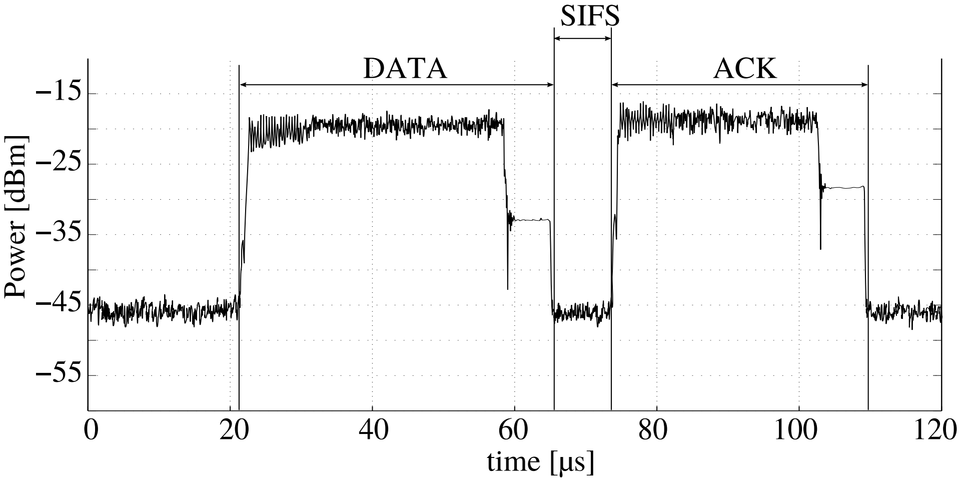


Figure 3.20 – A zero-span analysis of a IEEE 802.11g frame transmission, performed at 54 Mbit/s.

Transmit spectrum

As specified for the HR/DSSS PHY, the transmitted spectrum must conform to a specific mask. Fig. 3.21 on the next page shows the mask as specified by the standard.

Considering the most widespread use of this PHY, in which channels are spaced by 20 MHz, the transmitted spectrum must respect a 0 dBr bandwidth not larger than 18 MHz. Considering a 11 MHz offset with respect to the carrier, the spectrum must be below -20 dBr. -28 dBr at 20 MHz offset and -40 dBr after 30 MHz offset.

The standard also specifies that compliance measurment through a spectrum analyzer are to be performed using a 100 kHz resolution bandwidth and a 30 kHz video bandwidth.

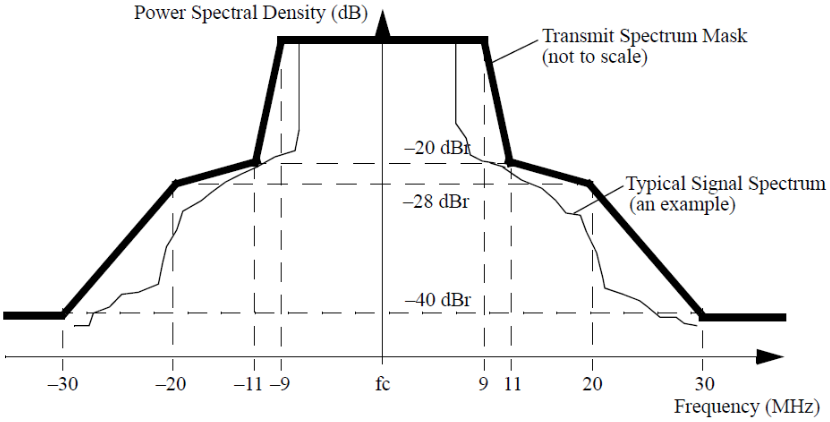


Figure 3.21 – Transmitted spectral mask for the ERP–OFDM PHY.

CCA

The ERP–OFDM PHY specifies that compliant devices have to implement only one CCA function, equivalent to the mode 3 of the IEEE 802.15.4 standard. In practice, the carrier sens mechanism shall be able to identify symbols from the modulations provided through this physical layer specifications. If an ERP–OFDM modulation is detected, then the medium is declared busy if the energy received at the antenna is above the specified threshold. The threshold is equal to -76 dBm.

3.4 Packet Error Calculation for the ERP–OFDM PHY

In order to discuss of the PER model for the IEEE 802.11g standard, a brief overview of the transmitter architecture is given. Actually, the robustness of the transmission is obtained through the use of a convolutional encoder, followed by a bit interleaver. Moreover, to obtain the higher transmission rates starting from the basic encoder output, a puncturing operation is performed. At the receiver side a hard-decision Viterbi decoder is typically used to obtain back the original bit stream. The modulation parameters foreseen by the standard have already been presented in Table 3.5 on page 58.

The correct reception of a frame carrying a payload whose length is l Bytes, adopting the modulation m among those listed in Table 3.5 on page 58, is achieved if and only if both the DATA packet and the subsequent ACK are correctly received and decoded. Therefore, it is possible to write:

$$P_{s,TX}(l, m, \sigma) = (1 - P_{e,DATA}(l, m, \sigma))(1 - P_{e,ACK}(m_A, \sigma)) \tag{3.7}$$

where $P_{e,DATA}(l, m, \sigma)$ and $P_{e,ACK}(m_A, \sigma)$ are the error probability on the DATA packet and the ACK one, respectively, being σ the SNR at the receiver antenna.

It is worth observing that each packet is composed by a first set of octets that are transmitted with a rate−1/2 convolutional encoding and a BPSK modulation, that is, the PLCP Signal field, whose length is 3 octets. The remaining part of the packet, containing also the l Bytes long payload, is effectively transmitted with the chosen modulation. In this light, using the same notation as above, a function $P_e(n, m, \sigma)$ can be introduced, representing the error probability for a frame (not only the payload) of n octets. Therefore, $P_{e,DATA}$ and $P_{e,ACK}$ can be expanded as:

$$\begin{aligned} P_{e,DATA}(l, m, \sigma) &= 1 - (1 - P_e(3, 1, \sigma)) \cdot \left(1 - P_e\left(\tfrac{36+l+22/8}{N_{BPSd}(m)}, m, \sigma\right)\right) \\ &= 1 - (1 - P_e(3, 1, \sigma)) \cdot \left(1 - P_e\left(\tfrac{38.75+l}{N_{BPSd}(m)}, m, \sigma\right)\right) \qquad (3.8) \\ P_{e,ACK}(m, \sigma) &= 1 - (1 - P_e(3, 1, \sigma)) \cdot \left(1 - P_e\left(\tfrac{14+22/8}{N_{BPSa}(m)}, m, \sigma\right)\right) \\ &= 1 - (1 - P_e(3, 1, \sigma)) \cdot \left(1 - P_e\left(\tfrac{16.75}{N_{BPSa}(m)}, m, \sigma\right)\right) \qquad (3.9) \end{aligned}$$

Hence, the goal is finding an appropriate formulation for $P_e(n, m, \sigma)$. An upper bound to the packet error probability P_e is found in [53], where a binary convolutional code and the Viterbi decoder with hard-decision are adopted, as defined by this standard [32]. For the case of a packet having length l octets and transmitted with the modulation m, it is:

$$P_e(l, m, \sigma) \leqslant 1 - [1 - P_u^m(\sigma)]^{8l} \tag{3.10}$$

where P_u^m is the union bound [70] on the first-event error probability. This is expressed by:

$$P_u^m(\sigma) = \sum_{d=d_{free}}^{\infty} a_d P_d(\sigma) \tag{3.11}$$

where d_{free} is the free distance of the convolutional code, a_d represents the number of paths of distance d from the correct path, and $P_d(\sigma)$ is the probability of choosing an incorrect path of distance d from the correct one.

Coefficients a_d are code dependent, and can be found through the transfer function $T(D)$ of the convolutional encoder [52, 21], or may be obtained via numerical searches. In particular, for the case of the standard IEEE 802.11g the encoder is specified in Section 3.3.2 on page 59. In order to help a practitioner interested in implementation of Eq. (3.11), the required parameters have been calculated in Table 3.6. Considering an hard–decision Viterbi decoder, the probability $P_d(\sigma)$ can be calculated as in the following equation:

Table 3.6 – The coefficients a_d to be used in Eq. (3.11). The first 10 values are given, providing them a very accurate approximation of the sum in (3.11).

Code rate	Free distance	Puncturing pattern	$\{a_d\}$
1/2	10	*none*	11, 0, 38, 0, 193, 0, 1331, 0, 7275, 0, 40406, 0
2/3	6	1 1 1 0	1, 16, 48, 158, 642, 2453, 9174, 34705, 131585, 499608
3/4	5	1 1 0 1 0 1	8, 31, 160, 892, 4512, 23307, 121007, 625059, 3234886, 16753077

$$P_d(\sigma) = \begin{cases} \sum_{k=(d+1)/2}^{d} \binom{d}{k} p^k (1 - p^{d-k}) & d \text{ odd} \\ \\ \frac{1}{2}\binom{d}{d/2} p^{d/2} (1-p)^{d/2} + \\ \sum_{k=d/2+1}^{d} \binom{d}{k} p^k (1 - p^{d-k}) & d \text{ even} \end{cases} \tag{3.12}$$

where p represents the bit error probability, due to the specific modulation scheme and channel model. In the final model, the packet error probability has been slightly overestimated by replacing in Eq. (3.10) the inequality symbol with its upper bound.

Expressions for the bit error probability for each of the modulation adopted by this standard are typically found in many communication books [2, 52, 26]. For example, the bit error probability for a BPSK modulation, which is in this case equal to the symbol error probability, is:

$$p_{BPSK}(\sigma) = Q(\sqrt{2\sigma}) \tag{3.13}$$

where the well-known Q–function is defined as:

$$Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^{\infty} e^{-\frac{w^2}{2}} dw \tag{3.14}$$

Such calculation of the packet error probability has been implemented both for the case of a Additive White Gaussian Noise (AWGN) channel and for a Rayleigh channel, which in turns entails different expressions for the bit error probability.

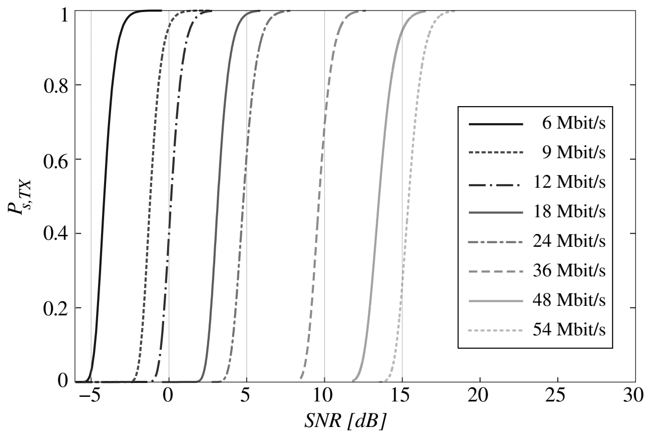


Figure 3.22 – Probability of a successful transmission at the first attempt, for all data rates *vs* SNR. Payload size of 46 bytes.

In order to give a clear picture about the model just described, it is possible to calculate the probability of a successful transmission in an interesting case. Let us consider the transmission of a packet with payload of $l = 46$ bytes (*i.e.* an amount of data typical of the lower level of factory automation systems). The probability that the packet is correctly received by a node, *i.e.* the DATA and ACK frames are both received correctly without the need of any retransmission at the *first* attempt can be obtained using Eq. (3.7) and following the procedure outlined in Eq. (3.9) through (3.12). The Fig. 3.22 plots the resulting probability with respect to SNR for each data rate described by the standard (Table 3.5 on page 58).

Chapter 4 The simulator

Several network and protocol simulators are nowadays freely available to researchers and designers. For example, it is possible to find many specific projects dealing with some core issues or enabling a practitioner to work with one well defined protocol. Beside these, more important and interesting examples come from *Opnet, Ns2* and OMNeT++. The first is a commercial package, everywhere appreciated for protocol tuning or for the design and optimization of networks; anyway, it is not conceived to implement and assess new or non-standard protocols. More general well-known environments are *Ns2* and OMNeT++, which are open source projects providing a rich set of tools for the design and analysis of networks. The core functions made available by such simulators are rather limited; anyway, extensions can be developed in a simple manner. The scientific community has developed in the time a number of simulation frameworks based on such cores, which represent a good basis for the analysis of network protocols.

On this basis, the present chapter describes the simulator realized through this work, purposely designed for the evaluation of the coexistence of several interfering wireless networks in the same environment, which is typically represented by the factory automation scenario. It allows a cross–layer approach. The work is based mainly on the results presented in [5, 13] and [7].

4.1 The development environment

The design and development of the simulator has been realized in the OMNeT++ discrete event simulator[50]. There are several reasons behind this choice. When this work started there was a very high activity in the network simulation community, as it is the case at the present moment. Several simulation environment are available, and the main projects in this field are OMNeT++ *Network Simulator* (NS, version 2 or 3) in the case of open source freely available simulators, or solutions like those provided by *Opnet* [51], *Matlab Simulink* [45], etc in the case of commercial packages. The latter softwares were initially excluded for several reasons: *Opnet* was not licensed to my institution and its characteristics and capabilities were not comprehensively known. General purpose simulators, or numerical solvers, like *Matlab* represented a viable solution, but since they are not specifically conceived for network simulations they have been excluded. Among the open-source projects, it is worth mentioning that there are many simulator specifically designed to solve specific problems, and an example for all is the *Pythagor simulator* [16], that is designed for IEEE 802.11 networks only. However, the need for a development platform specific for network simulations enabling an effective design of new network protocols, scenarios and models led to the restrict the choice between the well-known NS2 (or the forthcoming NS3) and OMNeT++. NS2, while being more widespread, seemed more confusing in its code organization, and the availability of user contributed code and discussion groups more fragmented. The final choice of OMNeT++ has been performed on the basis of its high modularity in the representation of each network component, the rich set of provided Application Program interfaces (APIs), the good source code organization and the ease in the development of extensions. It is also supported by a very large user community, and has been enriched of several new features over time.

OMNeT++ is an object-oriented modular simulation environment, that models discrete event system, and whose primary application area is the simulation of communication networks. However the generic and flexible architecture permits its use for modeling multiprocessors and other distributed hardware systems, evaluating performance aspects of complex software systems and also performances of business processes. In the present version it comes as a complete Integrated Development Environment (IDE), that helps in writing and keeping organized the code of a project, in editing and configuring simulations and also represents the same environment where outcomes from simulations can be analyzed, and eventually exported for further post-processing. OMNeT++ provides also a graphical interface, called TkEnv, which is enabled when a simulation is started. This enables the analysis of the simulation behavior, thus realizing some basic debugging functions, and show what links have been created and what does not work, how packets move on links, if packets flow works as expected, the temporal scheduling of packet that will be transmitted, etc. It can optionally be suppressed to avoid memory overheads in the case debugging functions are not needed, making execution speed grow.

With OMNeT++, each basic block is a “module” which is composed by

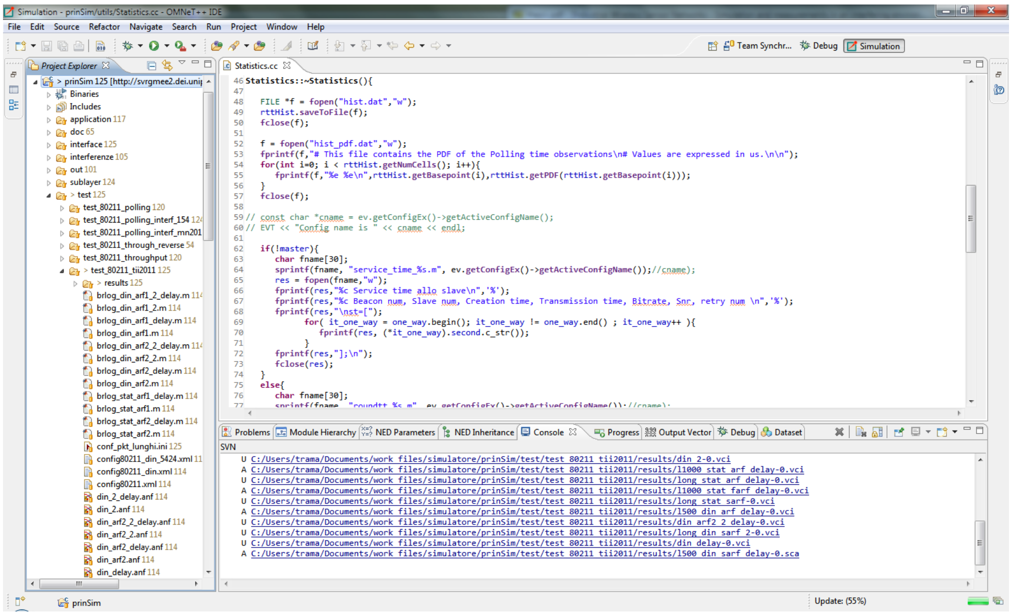
✓ logical description, specified through an internal programming language, called *NED language*. Here a user defines the logical characteristics of the module: the name, the configuration parameters to expose to the simulation, the connection points for other modules (input and output gates), etc.

- ✓ functional description, written in C++ programming language. Here the behavior of the module is specified, how it reacts in the case of a specific event is triggered, which data are collected, etc. Protocols and applications are therefore written in C++, exploiting the APIs provided with the development environment.

- ✓ message description. In the case the module defines some specific message type, a message declaration language is used. This is a handy language, through which one specifies the name and composing fields of the message. It is an OMNeT++ task to automatically convert this in a C++ class.

Starting from simple modules, which realizes the atomic functions of each component of a simulation project, "compound" modules can be created, suitably connecting simple modules together through the defined connection points.

A screenshot of the OMNeT++ development environment is given in Fig. 4.1.


Simulation - prinSim/utils/Statistics.cc - OMNeT++ IDE
File Edit Source Refactor Navigate Search Run Project Window Help
Team Synchr... Debug Simulation
Project Explorer
> prinSim 125 [http://svrgmee2.dei.unip
Binaries
Includes
application 117
doc 65
interface 125
interferenze 105
out 101
sublayer 124
> test 125
test_80211_polling 120
test_80211_polling_interf_154 124
test_80211_polling_interf_mn201
test_80211_through_reverse 54
test_80211_throughput 120
> test_80211_tii2011 125
> results 125
brlog_din_arf1_2_delay.m 114
brlog_din_arf1_2.m 114
brlog_din_arf1_delay.m 114
brlog_din_arf1.m 114
brlog_din_arf2_2_delay.m 114
brlog_din_arf2_2.m 114
brlog_din_arf2_delay.m 114
brlog_din_arf2.m 114
brlog_stat_arf1_delay.m 114
brlog_stat_arf1.m 114
brlog_stat_arf2_delay.m 114
brlog_stat_arf2.m 114
conf_pkt_lunghi.ini 125
config80211_din_5424.xml 11
config80211_din.xml 114
config80211.xml 114
din_2_delay.anf 114
din_2.anf 114
din_arf2_2_delay.anf 114
din_arf2_2.anf 114
din_arf2_delay.anf 114
din_arf2.anf 114
din_delay.anf 114
Statistics.cc
Problems Module Hierarchy NED Parameters NED Inheritance Console Progress Output Vector Debug Dataset
SVN
prinSim
Update: (55%)

```
Statistics::~Statistics(){

   FILE *f = fopen("hist.dat","w");
   rttHist.saveToFile(f);
   fclose(f);

   f = fopen("hist_pdf.dat","w");
   fprintf(f,"# This file contains the PDF of the Polling time observations\n# Values are expressed in us.\n\n");
   for(int i=0; i < rttHist.getNumCells(); i++){
      fprintf(f,"%e %e\n",rttHist.getBasepoint(i),rttHist.getPDF(rttHist.getBasepoint(i)));
   }
   fclose(f);

// const char *cname = ev.getConfigEx()->getActiveConfigName();
// EVT << "Config name is " << cname << endl;

   if(!master){
      char fname[30];
      sprintf(fname, "service_time_%s.m", ev.getConfigEx()->getActiveConfigName());//cname);
      res = fopen(fname,"w");
      fprintf(res,"%c Service time allo slave\n",'%');
      fprintf(res,"%c Beacon num, Slave num, Creation time, Transmission time, Bitrate, Snr, retry num \n",'%');
      fprintf(res,"\nst=[");
            for( it_one_way = one_way.begin(); it_one_way != one_way.end() ; it_one_way++ ){
               fprintf(res, (*it_one_way).second.c_str());
            }
      fprintf(res,"];\n");
      fclose(res);
   }
   else{
      char fname[30];
      sprintf(fname, "roundtt_%s.m", ev.getConfigEx()->getActiveConfigName());//cname);
```

```
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/din_2-0.vci
   A C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/l1000_stat_arf_delay-0.vci
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/long_stat_arf_delay-0.vci
   A C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/l1000_stat_farf_delay-0.vci
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/long_stat_sarf-0.vci
   A C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/l500_din_arf_delay-0.vci
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/din_arf2_2_delay-0.vci
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/long_din_sarf_2-0.vci
   U C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/din_delay-0.vci
   A C:/Users/trama/Documents/work_files/simulatore/prinSim/test/test_80211_tii2011/results/l500_din_sarf_delay-0.sca
```

Figure 4.1 – A screenshot of the OMNeT++ IDE.

4.2 A preliminary experience

Exploiting the features provided by this tool, a network simulator for the coexistence analysis of interfering WSNs had been realized [5, 13, 6].

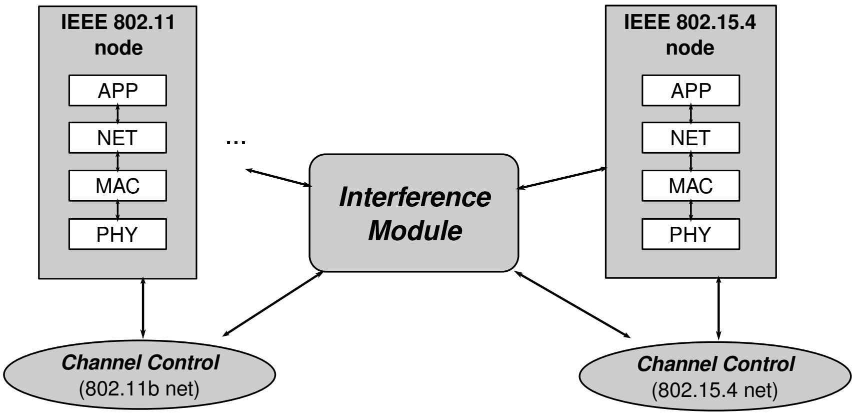


Figure 4.2 – A schematic overview of the simulator proposed in [5].

In the quoted works, each network was isolated from the others. At each transmission of a frame on the wireless medium, a central *Interference module* was informed, transmitting to the Interference module a frame containing all the information to identify the real transmitted packet (*i.e.* the sending time, the length, the sender and receiver module, etc). This was the core module of the simulator. It represented the only common point for the two different network, typically based on IEEE 802.15.4 and IEEE 802.11 standards, respectively. A schematic diagram of the architecture is represented in Fig. 4.2. When the real packet had to be received by the destination node, the *Interference module* was queried. This entity, from the knowledge of all the frames on the medium at a given time determines all messages that could have a time overlap with the packet being delivered. If any packet with temporal coincidence with this last was found, the *Interference module* calculated if a collision was realized. Interference was meant as a comparison between the actual frame reception power and the power that the interfering frame would have at the antenna of this receiver. The Signal To Interference Noise Ratio (SINR) was then calculated and the value used to obtain the actual PER for the frame. A Bernoulli trial with a probability equal to PER was used to decide if the message had to be discarded.

The simulator supported IEEE 802.11b– and IEEE 802.15.4–based nodes, the mobility of nodes, a cross–layer approach, and allowed the analysis of different interesting case studies. However, one serious limitation of this software was that decisions was taken in a central entity, which managed the entire network. If the model for the physical process would be updated, or enriched with new features, these changes should be also reflected coherently in the Interference module. Moreover, the implementation of new channel models, or new transmission schemes were to be also implemented in the Interference module, highly impacting both on the simulator performance and on the code readability.

4.3 The new simulator architecture

Starting form those experience, and that coming from other interesting frameworks based on OMNeT++ that in the meantime appeared publicly on the OMNeT++ community, a thorough revision process of the simulator has been performed.

As a matter of fact, other research groups tackled the problem of network simulators able to simulate multiple networks coexistent on the same area. Among all, the most complete and accurate of the realized projects are MiXiM [40] and INET [49][1]. The former, in particular, presents a base physical layer that handles collisions among data packets in the air in a more effective way with respect to the previous version of the simulator.

Namely, a virtual channel shared among all nodes is realized, where each packet is represented by a "virtual" *signal*. This is a three–dimensional vector carrying frequency, time and space information with it. Each time a new frame is sent on the air the state of the virtual channel is updated. This happens also when a frame is no longer able to interact with any other, and is deleted from the channel. Frames intersecting both in time and frequency are considered as possible interferers. In this light, the channel model can be represented as a digital filter for signals, allowing one to easily separate the logical management of frames from its physical modeling [74]. The physical layer modeling presented in this framework has therefore been chosen as the underlying process model in the new simulator. This however, represents a general purpose implementation, and do not fulfill all the needs of the research activities presented in this thesis, especially in terms of channel models and PER models.

Therefore, the "past" simulator has experienced an in–depth revision process, where the physical process interactions are modeled using the same schemes provided by MiXiM. Some models have been ported and extended from the past experience [6], in particular the MAC and PHY protocol layers described by the standards IEEE 802.15.4 and IEEE 802.11 [31, 32]. Regarding the IEEE 802.15.4 standard the packet error calculation algorithm has been improved, and the description given in Section 2.2.4 on page 30 represents the actual implementation of PER estimation in the simulator. Moreover, the support for IEEE 802.11g networks has been added, which required a complete reimplementation of the physical layer for IEEE 802.11 since the standard defines several new modulation schemes, and consequently the estimation of the PER should be updated accordingly, as specified in Section 3.4 on page 63.

[1] It is worth noting that for the case of wireless networks the correct framework should be INETMANET, that is, a simulator specifically designed for Mobile Ad–hoc Networks (MANET).

4.4 A typical simulation setup

In the following Fig. 4.3 a common scenario for a simulation is represented, showing the principal entities composing a simulated network.

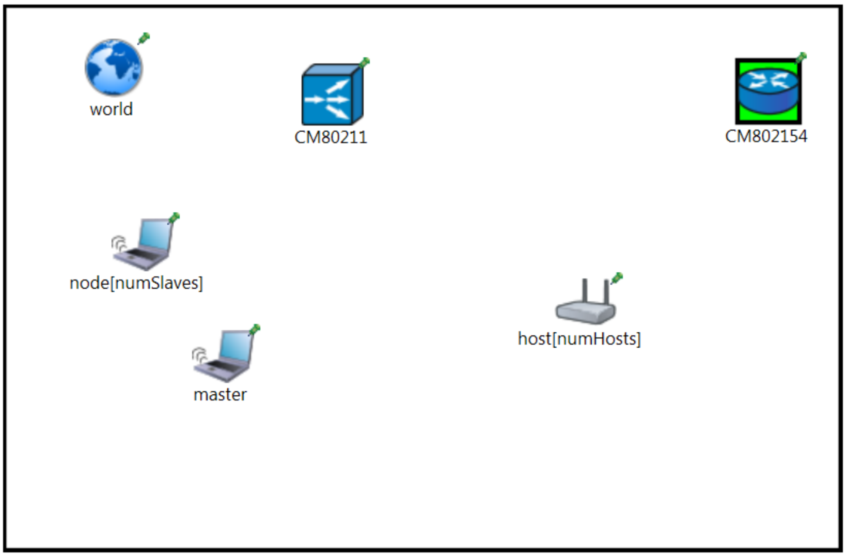


Figure 4.3 – A screenshot of a typical simulation setup.

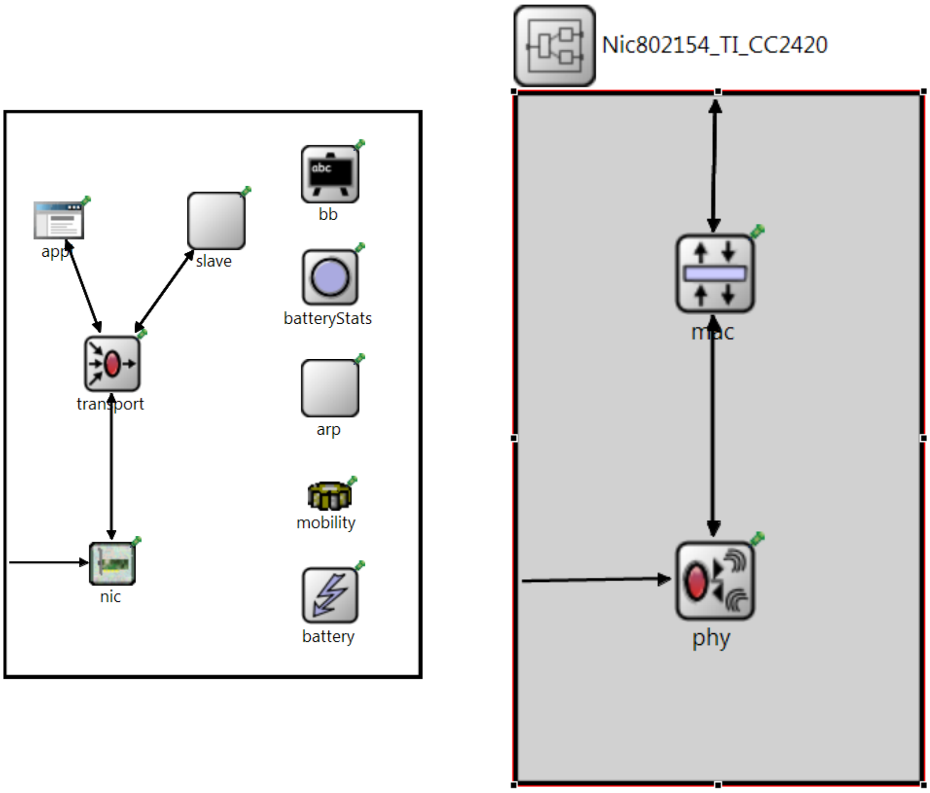
In the figure, the main modules are those representing the "Channel Manager" nodes and those implementing the communication devices.

The former are indicated in Fig. 4.3 with the names `CM80211` and `CM802154`. These modules implements all the mechanisms needed for the management of connections among nodes. If two nodes are in the transmission range of one of the other, *i.e.* a message sent by a node results in a received power level above the sensitivity threshold of the other node, they will be connected. Conversely they are leaved unconnected. Two nodes will be also disconnected if their reciprocal distance increases in such a way that they are no longer able to hear one to the other. Clearly, each connection manager manages only the nodes pertaining to its own network.

Communication devices, in Fig. 4.3, are nodes compliant to the standard IEEE 802.15.4 and IEEE 802.11. In the following a brief overview of these nodes will be given.

4.4.1 IEEE 802.15.4 nodes

The implementation of the IEEE 802.15.4 standard follows the specifications given in Chapter 2.



(a) *(b)*

Figure 4.6 – The architecture of a IEEE 802.15.4 node in the simulator.

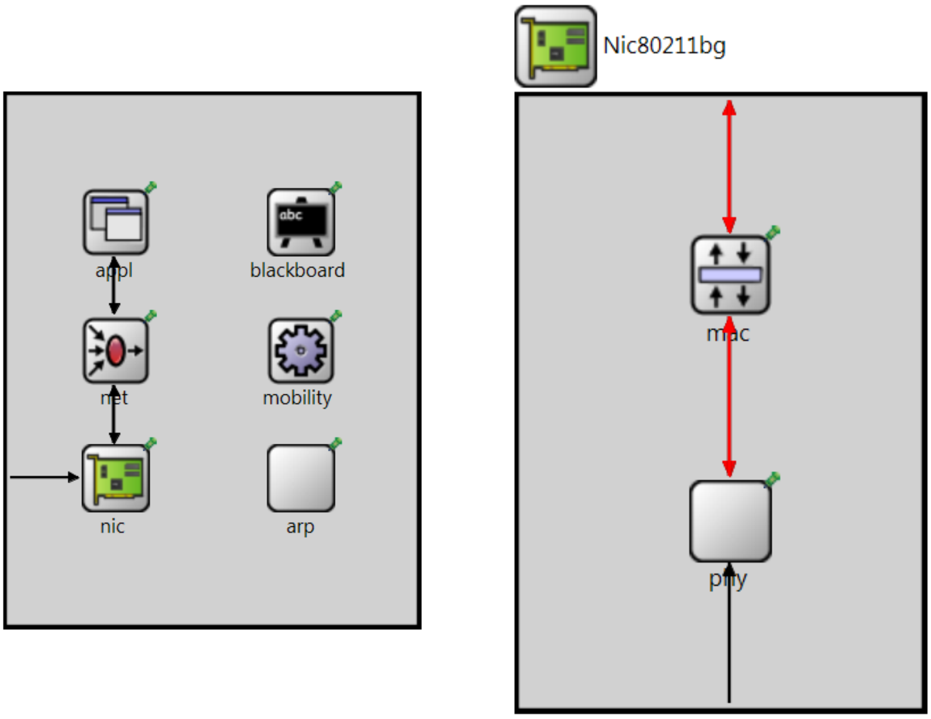
As can be observed in Fig. 4.6, the architecture of a node realizing sensor based on the IEEE 802.15.4 standard is rather simple. It is composed by a Network Interface Card (Nic), implementing the lower levels of the protocol stack, a "Transport" layer which receives data from one or more application layer. In practice, these application layer represent packet generators, and also implements the methods for the correct reception and management of packets.

The Nic implements the MAC and PHY layers, as depicted in Fig. 4.6b. These realizes a subset of the functions provided by the standard. For example, both FFD and RFD nodes have been implemented, the MAC allows for the operations both in *beacon* and in *beaconless* mode. In this implementation only the CCA mode 1 is defined, that is, when the CSMA-CA algorithm is performed, to assess free the channel we evaluate only if the power at the receiver is below a predefined threshold. The physical module implements an abstraction of the physical process required to form a PPDU frame from a given MPDU. It also realizes the packet error calculation described in Section 2.2.4 on page 30.

Others entities present in the architecture shown in Fig. 4.6a on the preceding page are the “Battery”, the “Arp” and the “mobility” modules. The former is introduced for the purpose of simulating the power consumption of a node, associating a discharge profile for each operation performed by the node. The “Arp” module represents the abstraction of the classic ARP service, which traduces the physical address of a machine with its logic address. It is not strictly needed for the architecture of the simulator, since nodes could have been addressed directly by their physical address, but in this way a clear separation between the two has been introduced, and also helped in the light of the realization of a “Transport” module. Finally, the “mobility” module provides functions for a dynamic change of the position of the node during a simulation. This could happen exploiting various paths or models. The simpler one is a straight line movements, with a constant velocity. This one is also used for setting up a fixed node (*i.e.* velocity equal to zero).

4.4.2 IEEE 802.11 stations

Since the architecture of these devices is very similar to that of IEEE 802.15.4 nodes, only the principal differences are here highlighted.



(a) *(b)*

Figure 4.9 – The architecture of a IEEE 802.11 node in the simulator.

An interesting difference is the absence of the “Battery” module, because this type of device are not conceived for battery efficiency, and the simulation of battery consumption is not necessary. Even if only one application layer is here shown, the simulator support the presence of more application layers, through the “Transport” module.

About the Nic, it implements the basic DCF procedure for channel access (*i.e.* the classic CSMA/CA), while PCF, EDCA and HCCA are not supported yet. The RTS/CTS method is also supported, as well as its threshold is a simple configurable parameter. Fragmentation is not supported, instead.

4.4.3 The “Transport” module

In Fig. 4.6a on page 73 and Fig. 4.9a on the facing page the module directly communicating with the Nic is represented by the abstraction of a Network layer of the ISO/Open System Interchange (ISO/OSI) stack.

Often in the simulation studies performed through this simulator there is no interest in the analysis of particular network protocols, since the simulated scenarios are typically simple. However, there are situations in which more traffic sources/flows have to be considered at the same node. It is the case, for example, of the implementation of the Precision Time Protocol (PTP) synchronization protocol [33]. In this case, the protocols acts as an application running at the same level of the other traffic sources.

The “Transport” layer therefore, provides a mechanism very similar to *sockets*. Each application layer must associate itself with the Transport module, choosing one (or more) communication port, during the network initialization phase. The Transport module stores the association between port and module, in order to correctly deliver the messages arriving from the plower levels. It also implements a queue, in order to simulate the finite and not zero elaboration process to perform these operations, and also to avoid multiple packet to be transferred at the same time.

This module has been further extended in order to effectively account for other critical real-life effects. In fact, at this level some models of the delays introduced by real network components, *e.g.* APs or PC communication boards, have been implemented, stemming them from the practical experience coming from measurements, such as those presented in [61, 60], and those presented in Chapter 6.

4.4.4 Rate adaptation for IEEE 802.11

A relevant feature implemented in the simulator is the support for rate adaptation algorithms for IEEE 802.11 networks, enabling nodes to modify their transmission technique to the actual channel conditions.

In the scientific literature, a lot of algorithms have been proposed to this aim. Nonetheless, the widespread solution is typically the use of Automatic Rate Fallback (ARF) algorithm [39, 46].

As a brief overview, ARF specifies that, given the set of supported transmission rates $R = \{r_1, r_2, \dots, r_Q\}$, a station decreases its transmission rate from r_i to rate r_{i-1} after K consecutive failed (unacknowledged) transmissions, while increasing it from rate r_i to rate r_{i+1} after N consecutive successful transmissions. However, if the first transmission attempt (*probing* transmission) fails after the transmission rate has just been increased, then the station will immediately switch back to the lower rate without performing the remaining $K - 1$ attempts at the higher rate. Moreover, in [39] it was proposed to included a timer, started by a station when it decreases the transmission rate and whose expiration allows the station to switch up to an higher rate even if it has not performed the requested N consecutive successful transmissions.

An adaptive version of this model, Adaptive ARF (AARF), have been also implemented.

Since the nature of the simulator, whose aim is to represent a design platform for new or modified protocols for real-time networks, these algorithms have been chosen as representative of the literature rate adaptation models. New rate adaptation algorithms are instead been proposed, designed and tested exploiting this simulator, as for example the Static retransmission rate ARF (SARF) and Fast rate reduction ARF (FARF) models. They will be detailed in Chapter 7.

In the following Section 4.6 on page 83 the rate adaptation capabilities implemented in the simulator are exploited in an environment with a low level of SNR.

4.4.5 Channel models

In the actual version of the simulator there are different implemented channel models. These are typical literature models well-suited for indoor wireless communications. This to the aim of further improving the accuracy of the physical layer implementation. The architecture of the physical layer, however, allows a very easy implementation of new channel modes.

In particular, the classic free-space propagation model is implemented. In a default configuration, the signal power decreases with the square of the distance. A configuration parameter allows an easy setting of this exponent, where for example a value between 3 and 4 results often more realistic. Moreover, a two–slope path loss model has been implemented, following the guidelines of the IEEE 802.15.2 standard [30]. It follows the classic free space propagation model till 8 meters, that is, the received power decrease with the square of the distance, and after that threshold assumes that power decreases with the 3.3 power of distance.

$$P_{r|dBm}(d) = P'_{r|dBm}(d_0) - 33\log_{10}\left(\frac{d}{d_0}\right)$$

where $d_0 = 8$ m, and $P'_{r|dBm}(d_0)$ represent the received power at the distance of 8 meters calculated through the Friis formula.

More complex models implemented in the simulator are the Rayleigh fading, the Rician fading and the Nakagami fading models [26, 52], that are often adopted in the modeling of an industrial (or indoor) environment. These are often considered small–scale effects, in that they represent a statistical description of the channel behavior superimposed to the effect of path-loss, shadowing, etc. In general, while the Rayleigh fading better describes situations with a high number of scatterers and a non-Line of Sight (LOS) transmission, the Rician fading is better suited for the LOS case. Often the latter is considered a generalized version of the former. The Nakagami model, is a "refined" version of the others, and describes well situations in which, for example, multiple Rayleigh–fading signals independent each other sums at the receiver. It is widely adopted to models indoor channels with an high number of scatterers.

4.5 A first case–study: coexistence between IEEE 802.11 and IEEE 802.15.4 networks.

Throughout the work that lead to this thesis, the simulation outcomes have been always validated through real-life measurements, which have been often used also to finely tune and characterize the behavior of the simulator itself. In this preliminary case–study, the analysis of coexistence between IEEE 802.11 and IEEE 802.15.4 networks have been considered. The simulations have been carried out adopting the testbed depicted in Fig. 4.10, induced by the availability of results from [1], and [28].

Three different scenarios are adopted: the first one, used to analyze the Packet Loss Rate (PLR) statistic, involves two IEEE 802.11b nodes exchanging data each other under the influence of the traffic produced by an IEEE 802.15.4 compliant WSN.

The latter is composed by a FFD acting as a PAN coordinator, three other FFDs acting as traffic generators, and a RFD representing the receiver host. The traffic generated by transmitters is represented by broadcast frames. The parameter here was the interarrival

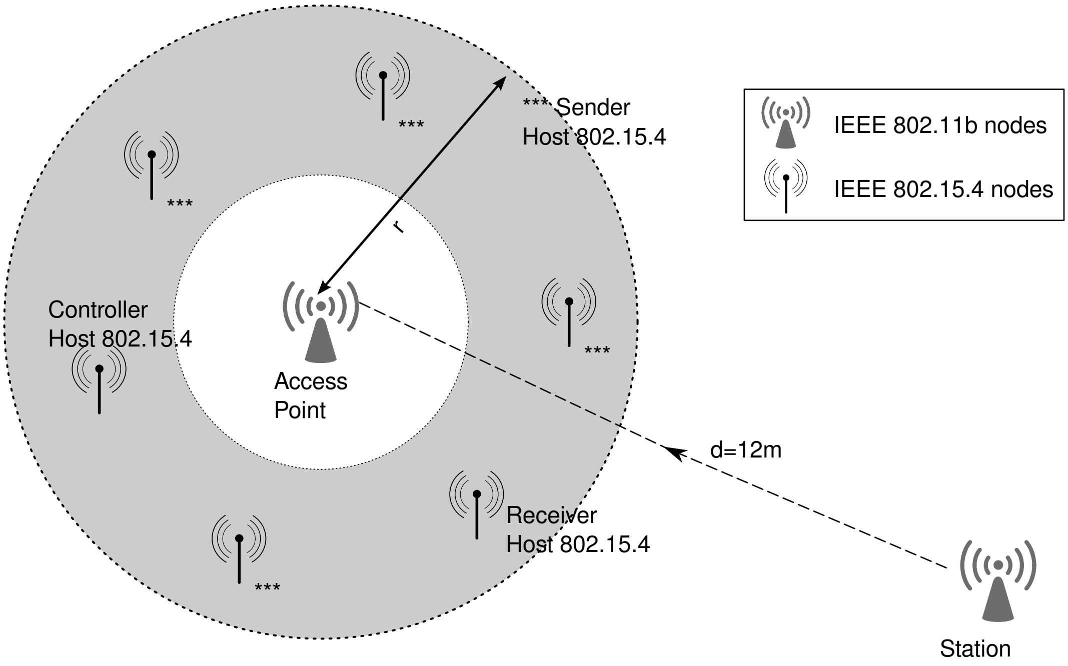


Figure 4.10 – The set-up of the environment in the simulation.

time between two packets sent by the same host.

The simple WLAN adopted in this scenario, is composed by a station and an AP, at a distance of 12m one from the other, performing an up-link transmission toward the station. Nodes of the WSN are organized in a circular region, with a radius of 2 meters, around the STA.

The second scenario has been deployed to investigate the effect on the PLR of the variation of some parameters. From the geometrical perspective, the testbed is the same of Fig. 4.10, whereas now the PLR of the WSN is assessed, under the impairments due to WLAN traffic.

The latter scenario investigates the effect of the variation of the radius *r*, that corresponds to an increase in the distance of sensors from the WLAN STA.

Parameters considered in these simulations are the polling time of the IEEE 802.15.4 nodes, the duty–cycle and the packet size of the IEEE 802.11b network. *duty-cycle*, in this context, is regarded as the fraction between the time the wireless channel is occupied by an IEEE 802.11b packet, and the time between the starting bit of two consecutive packets. Considering also the IFSs foreseen by the IEEE 802.11b standard, and the random backoffs due to the contention of the channel by the nodes (CSMA-CA algorithm), the definition given above is accounted for in this scenario by changing the rate of generation of new packets by the AP, in the time unit.

On the other hand, general parameters, valid among all simulations, are the path loss model, the bit-rate of IEEE 802.11b (11 Mbit/s) and of IEEE 802.15.4 (250 Kbit/s) and the CCA mode 2. Geometrical position of nodes remain fixed in the first and the second scenario, where coordinates of WSN hosts have been generated randomly. In the last scenario, WSN' nodes have been moved proportionally with the distance (*i.e.* angles with respect to the WLAN's STA have been kept fixed). The path loss model chosen is the simple two-slope model.

4.5.1 First scenario: PLR statistic

Simulation runs have been carried out by varying the duty cycle of the WLAN network from 35% to 100%. Results obtained allow the assessment of the PLR either of the WLAN itself under the interference of the WSN, and vice-versa.

In Fig. 4.11 on the following page, the PLR of the WLAN obtained from simulations is shown and compared to experimental results taken from [1]. The experimental results have been taken either with the WSN switched off, and with the WSN active, with a polling time of 30 and 100 ms.

The graph of Fig. 4.11 on the next page shows that the adopted WLAN, even in a clean environment and without impairments due to WSN's traffic, experiences an abrupt increase in the PLR after the threshold of 75%. In real devices software and hardware related issues, as buffers and queue management, have to be taken into account, that degrade the performances of the WLAN. Moreover, it must be noticed that the data rate of IEEE 802.11b, here fixed at 11 Mbit/s, is a theoretical data rate. The effective throughput achievable by this kind of network is lower, typically below 8 Mbit/s, and this is due to the mandatory interframe spaces between packets foreseen by the standard, and to the medium contention algorithm employed (CSMA-CA). This implies that under stressful conditions (high duty cycles) real devices have to drop out packets from their queues because the channel is already "saturated" by other transmissions.

In the simulation environment, the models of MAC and PHY layer of the IEEE 802.11b standard do not account correctly those effects, as it is explained in [5]. This triggered a revision of the models of the MAC protocols in order to account for the aforementioned effects.

Simulations returned data with a very high accordance with the experimental ones, as seen in Fig. 4.11 on the following page. It is worth noticing that, with a packet interdepar-

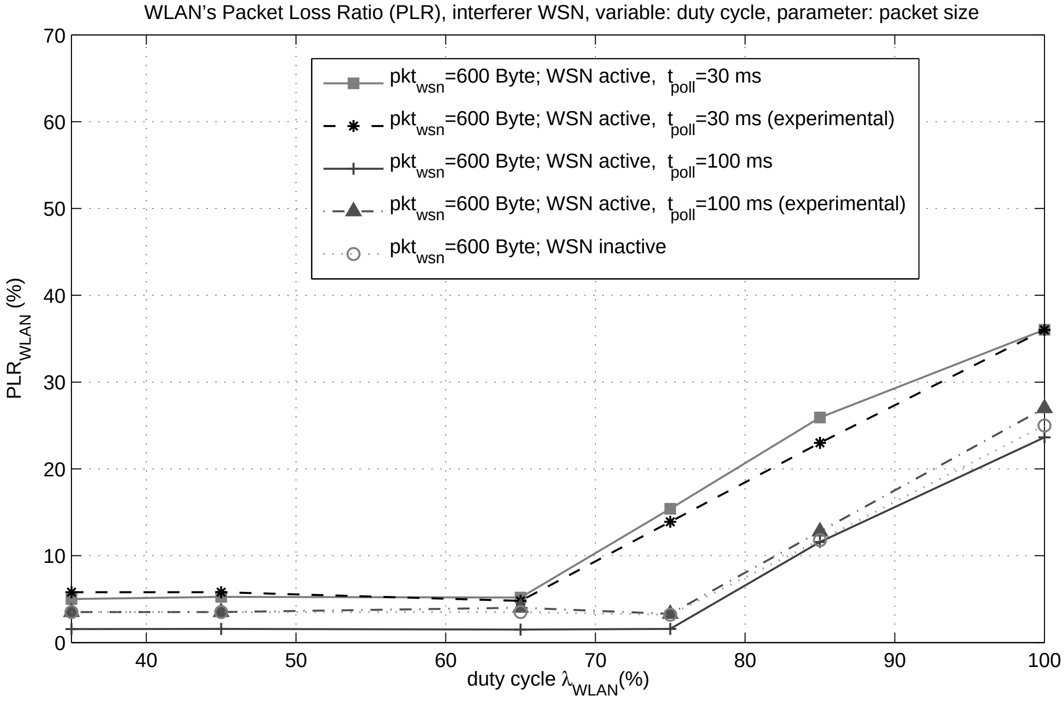


Figure 4.11 – Packet Loss Ratio (PLR) of the WLAN, under the interference produced by the IEEE 802.15.4 WSN.

ture time for the WSN of 100 ms, the simulations returned a PLR that underestimates the measured ones. However, one should note that the experimental PLR experienced with the WSN turned off has a curve very close to the experimental PLR with a WSN polling time of 100 ms.

4.5.2 Second scenario: variation of some parameters on the PLR of the WSN

The same simulation performed for the analysis given in the previous section, returned interesting data for the assessment of the PLR of the IEEE 802.11 WSN in different conditions. As in the previous figure, the variable taken into account is the duty-cycle of the WLAN, that now represents the interferer. The results obtained are shown in Fig. 4.12 on the next page.

Here, the parameters chosen are two values for the length of an IEEE 802.11b packet, and two values for the interdeparture time of packets for the slaves of the WSN, considering all the four possible combinations. Fig. 4.12 on the facing page does not highlight

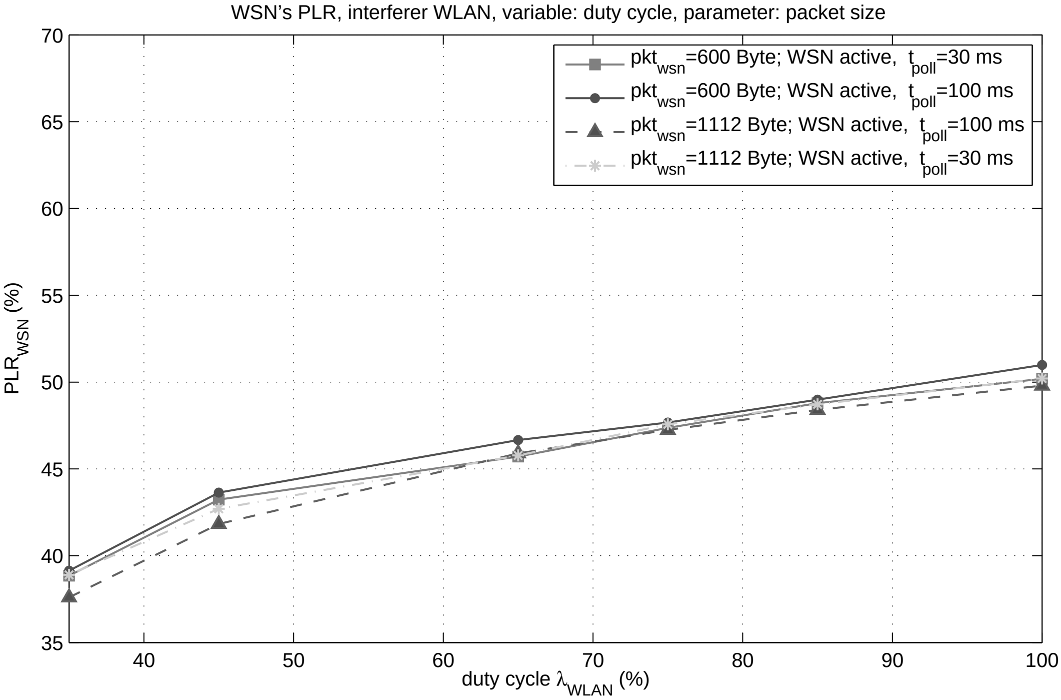


Figure 4.12 – Packet Loss Ratio of the WSN, under the interference produced by the IEEE 802.11b WLAN.

meaningful differences between the different PLR curves obtained for the four cases considered. However, the same simulations, see Fig. 4.11 on the preceding page, showed that changing the polling time of the WSN from 30 ms to 100 ms leads to a decrease of the PLR of the WLAN, to which respect the WSN was the interferer.

Therefore, one should note that, in an experimental perspective, if the focus is on the measurement of the PLR for the WSN, the interdeparture time can be kept fixed as well as the length of the packet: the attention can be dedicated to other variables, as the number of nodes, the channel employed, etc. Conversely, if the focus is the PLR of the WLAN, it can be inferred that the duty-cycle can be set below a conservative threshold of 60%, as in that range this parameter has a very small influence on the PLR curve.

4.5.3 Third scenario: variation of the radius of the WSN

On the basis of the presented experiments, as already discussed, the models implemented in the simulator have been adapted to real-life measurement results obtained with commercial devices. Data produced by simulations well agree with the experimental ones.

This justifies the use of this simulator in order to assess the effect of the variation of a particular parameter on the PLR statistic of the WLAN and the WSN.

An interesting case–study could be monitoring the effect of an increase in the distance of WSN nodes with respect to the WLAN's STA. Fig. 4.13 shows the results obtained through simulations.

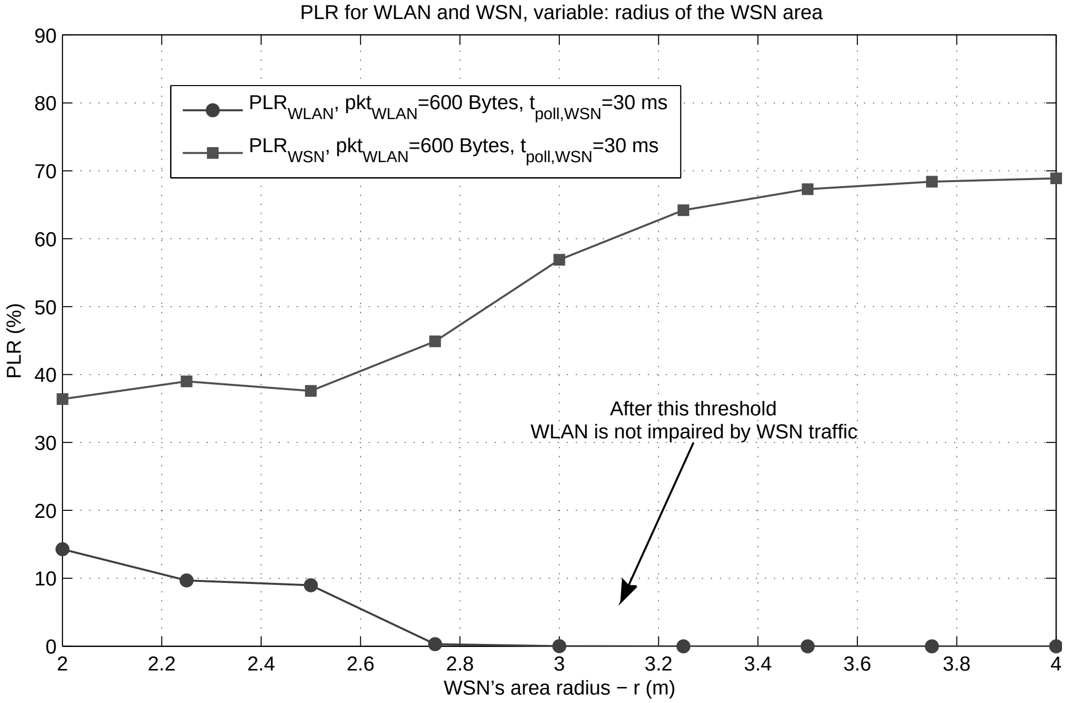


Figure 4.13 – Packet Loss Ratio of the WLAN and the WSN, for multiple values of the radius *r* of the area containing WSN nodes.

The positioning of the nodes in the simulated environment has been adapted accordingly to the new values of the radius considered in the experiments. In particular, the position of the WLAN nodes has been kept fixed, being the same used in the previous scenarios. The distribution of WSN nodes has been kept as in the previous experiments for the distance of 2 meters. From this base distribution, each node moves along an ideal line from the STA to the node itself, scaling the distance proportionally to the ratio between the new radius and 2 meters. This to avoid strange situations in which, choosing a new random distribution in a circular area centered at the STA, with an increasing radius each time, particular configurations could appear, such that the positioning of nodes for a particular distance will be equal to the one for a different area.

From Fig. 4.13 on the preceding page one can immediately observe that a threshold exists, after which the WLAN is no more influenced by the WSN transmissions. In fact, for a radius of 3 meters, and above, the WLAN experiences no errors. Conversely, the WSN performances, in term of PLR, worses with the increase of the distance. More in depth, the PLR level remains rather stable from 2 to 2.5 m, showing after that distance, an abrupt increase, reaching a level of 70% when the radius has been fixed at 4 m. One can also note that the WSN experiences the greatest variation in the PLR curve going from 2.5 up to 3.25 m, while the curve becomes rather flat around the 4 m threshold. This difference in the slope of the curve, experienced while the nodes come closer to the WLAN's AP, can be justified by the fact that the SINR at a WSN's receiver does not change linearly with the distance, as the received power varies with the square (or the power of 3.3) of the distance.

4.6 Results in the field of real–time wireless networks.

In order to understand the capabilities of the designed simulator a simple example is here described for the sake of clarity. More complicated experiments may be performed anyway. The sample scenario is depicted in Fig. 4.14. This simulations reflect similar experiments presented in [61, 60]. The main network is composed of a traffic source, connected through an Ethernet segment to the AP. The traffic is directed to the node named *Slave 1*. The traffic source generates a periodic traffic, with a period of 10 ms, exchanging packets with a limited payload of 46 bytes, since this traffic represents a typical situation in a industrial control network. The slave reply to this packet with an analogous one, which represents the answer containing the measurement collected at the sensor node queried.

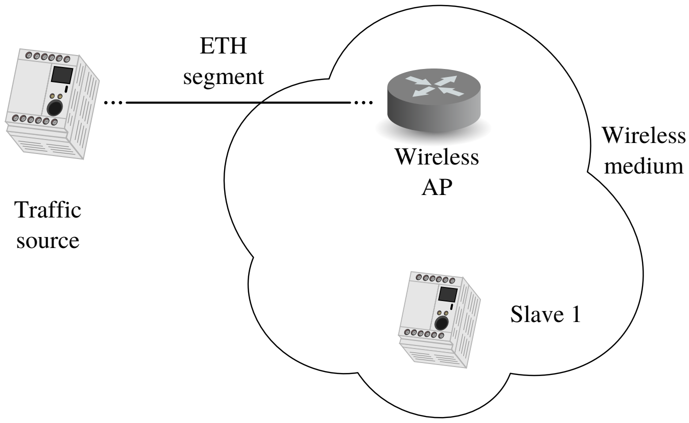


Figure 4.14 – The scenario considered in the examples.

This experiment, even if comprising a limited number of nodes, highlights some interesting improvements one can obtain through the use of the presented framework with respect to other existing simulators. In particular, the experimental results presented below prove that outcomes obtained through inaccurate models often found in simulation frameworks could bring to misleading conclusions.

4.6.1 Effects of some simulation inaccuracies

In the first experiment the impact of a correct calculation of the duration of frames on the round-trip time (RTT) at the traffic generator T_{RT} is analyzed. This is defined as the interval occurred between the instant in which the application level packet left the traffic source, and the instant in which the first bit of the response packet reaches the source. The hypothesis under which the simulation has been conducted was that the power received by a node is high enough to allow a correct communication at the higher data rate. This corresponds, comparing to Fig. 3.22 on page 66, to a SNR greater than 20 dB (in the simulation it was about 34 dB). These conditions bring the network to a *best-case* situation.

Under these assumption the RTT can be calculated as:

$$T_{RT} = 2T_{DIFS} + 2T_{w,DATA} + T_{SIFS} + T_{w,ACK}$$

where $T_{w,DATA}$ and $T_{w,ACK}$ represents the time to transmit a DATA and an ACK packet, respectively, and are calculated through (3.5) and (3.6). T_{DIFS} and T_{SIFS} are two inter-frame spaces defined by the communication standard, and their duration is 28 µs and 10 µs, respectively. In fact, a station wishing to communicate a packet to another has to sense the channel. If it is sensed free for a T_{DIFS} it has grant to transmit. The other station, upon the correct reception of the frame, waits for a T_{SIFS} and then sends back its acknowledgment packet. Finally, it waits for another T_{DIFS} and sends its response to the previous station. Considering the results presented in Chapter 3 on page 35, and Fig. 3.20, the theoretical RTT is 184 µs.

For example, in some literature frameworks an erroneous calculation of the control frames transmission time has been found. In particular two typical situation has been encountered:

i) the ACK packets are sent at the same rate of the data message, or

ii) the control frames are sent at fixed and wrong low rate (*e.g.* 1 or 2 Mbit/s).

Applying Eq. 3.6, in case *i*) the transmission time of the ACK packet results 30 µs, while in case *ii*) it results 94 µs. The final effect is resumed in Table 4.1.

Table 4.1 – Round-Trip Time values in different cases.

Case	Ack Rate [Mbit/s]	T_{RT} [µs]
Standard	24	184
i)	54	180
ii)	2	244

The above two simulation inaccuracies lead to a wrong calculation of the time required to complete a correct transmission, and this affects also other related metrics, such as the minimum cycle time (MCT) indicator. Moreover, the transmission rate implies the actual modulation/demodulation scheme, that in turn determines the PER, and the effects on the overall communication can be detrimental.

Such effect becomes more evident if the SNR level decreases, so as the rate adaptation technique adopted lowers the transmission rate to the best suitable one. In this case, the foreseen backoff procedure is triggered, and a randomness in the RTT value is added. An experiment in this direction has been realized lowering the SNR level at around 14 dB. This, in the considered scenario, brings the network to work correctly in between data rate 36 and 48 Mbit/s, continuously switching between them. The results obtained through the simulation campaign have been summarized in Table 4.2, which shows the mean, the standard deviation and the maximum value of T_{RT} taken on a set of ten thousand samples.

Table 4.2 – Round-Trip Time values in different cases, with a SNR of 14 dB.

Case	$\mu_{T_{RT}}$ [µs]	$\sigma_{T_{RT}}$ [µs]	max(T_{RT}) [ms]
Standard	284	211	1.9
i)	277.3	197.8	4.4
ii)	360.2	236.2	5

One should note that both cases show an inaccurate modeling of the expected behavior, with case *i*) being the closer to the correct one. In this case one finds an optimistic result. However, it is worth noting that in this situation a control frame has a weaker robustness because of the wrong choice of the modulation scheme. Therefore, control frames could be lost, and the controller would interpret this as a transmission error retransmitting the packet. This effect leads to a maximum round-trip time of 4.4 ms. Conversely, in case *ii*) the probability of loosing an ACK frame is close to zero, but the longer ACK transmission time leads to a relative error of about 27 % on the mean round-trip time, and the maximum value is about 2.5 times.

4.6.2 Simulation of the behavior of real network components

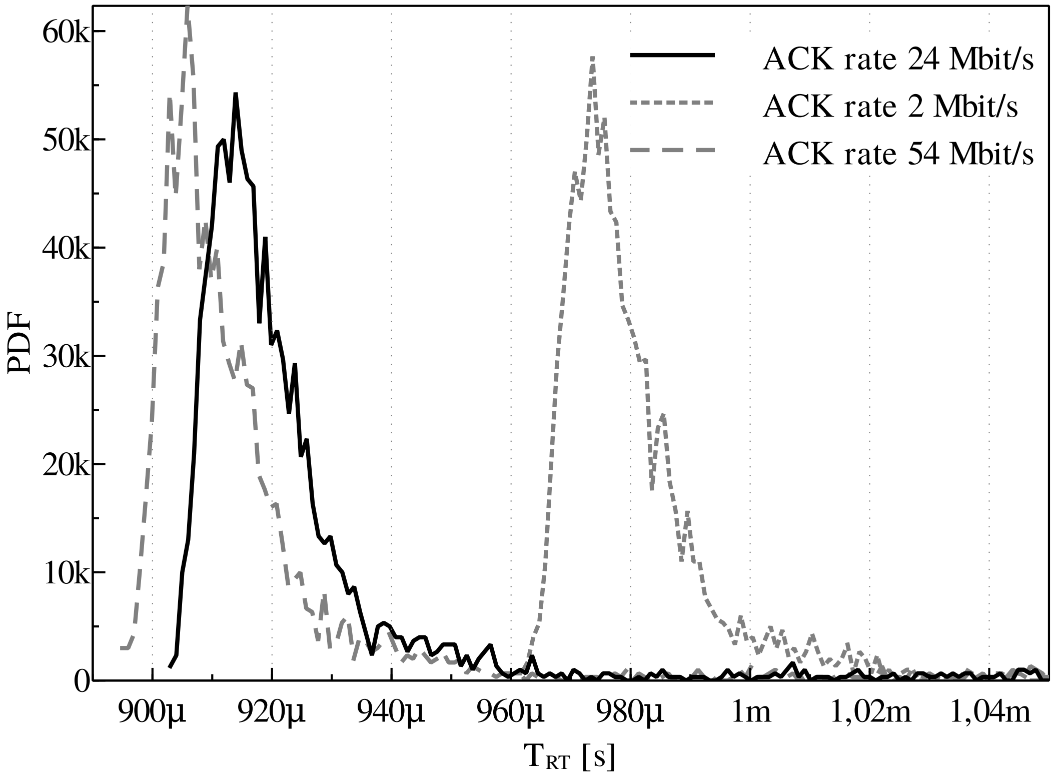


Figure 4.15 – Simulation for the round-trip time when component delays are considered.

Another experiment would further highlight the importance of a correct modeling of network components in order to obtain accurate simulation results. The simulations of the first experiment have been repeated considering know the processing delay models described in this work[2] Moreover, reproducing the cases *i*) and *ii*) in the calculation of the frame transmission time in our model, a direct comparison could be preformed, and is provided in Fig. 4.15. The solid line represents the correct behavior, while the dashed ones represent the two aforementioned cases. A direct comparison between Fig. 4.15 and Table 4.1 on the previous page makes evident the importance of a correct model for the components employed in industrial/control communication networks. The time T_{RT} needed

[2] Chapter 6 will describe in detail the experimental measurements performed on real components, and the models obtained. They have been directly used here, anticipating some results.

to complete a transaction lies between 0.9 and 1 ms, about 5 times the theoretically expected one, and shows a predominant random behavior. From the figure it is noticeable the spread in delays CDF in the order of 60/80 μs, as already pointed out. This difference in the prediction of the time spent to complete the polling of a node is much more evident if a more complex network is considered, where the number of nodes easily reach the order of tens, leading to errors in the setting of network cycle time.

4.7 Conclusions

The simulator presented in this chapter has been designed with the specific aim of studying interference effects among wireless networks, coexisting in the same area. Almost all the simulations presented in this work of thesis has been realized exploiting this simulation software. Actually, the scientific community in the time has increased its interest in this type of simulators, hence other frameworks presently allow interference analysis on wireless networks.

This tool, developed in OMNeT++, supports the IEEE 802.15.4 and IEEE 802.11(b/g) standards, and is easily extensible to support other communication protocols too. As an example, in its preliminary version it had been extended to simulate WirelessHART–based networks, comparing their performance with those of a "plain" IEEE 802.15.4 WSN (see [13] for further references).

The behavior of the simulator, and the outcomes obtained have been often compared with experimental results, as a way of performance and accuracy characterization of the tool. Some examples are given in Section 4.5 on page 77 and 4.6. Moreover, an analysis of some of the most widespread simulation tool freely available coming from the research community has been performed, highlighting in some cases some inaccuracies in the simulation of the behavior of the IEEE 802.11 standard, which result more apparent in the context of real–time networks, where time–related indicators are often adopted.

A revision of the implementation of the IEEE 802.11 MAC protocol, along with an experimental measure of the transmission times through a real–time spectrum analyzer allowed to increase the accuracy of the presented tool, as shown in Section 4.6 on page 83. Finally, with particular reference to the results that will be presented in the following Chapter 6, models of the delay introduced by some network components have been implemented in the simulator. Once again, the influence of these models in a simple yet relevant case of application is analyzed in Section 4.6, highlighting the considerable difference between the round–trip time calculated through a common simulator and that obtained through the presented tool, which can be in the order of 750 µs.

Chapter 5

Retransmission strategies

The overall traffic in many wired industrial networks comes in a very large part from periodic exchanges of data between a central controller and its attached sensor/ actuator stations, as for example in WorldFIP or PROFIBUS-DP [68]. A typical setting is to have two time windows, alternating in time. A first one, the *periodic window*, is reserved exclusively for the treatment of periodic traffic, and is cyclically repeated. A second window is instead reserved for sporadic traffic.

As already pointed out, there is an increasing interest in wireless extension of wired segment, where similar characteristics will be required [80, 81]. However, the wireless medium is much more subject to external interference, which severely impacts the reliability of the communication and causing the controller operating on an inconsistent view of the physical process. Nonetheless, the most viable solution is to consider as given the underlying wireless physical layer. Assuming it is not perfectly reliable one of the major control knob to improve the delivery rate, for wireless sensor networks based on the IEEE 802.15.4 standard, are link-layer retransmissions.

The present chapter, mainly based on [23], will deal with the topic of retransmission strategies for centralized wireless industrial systems based on the IEEE 802.15.4 standard subjected to external interference. The goal is to propose alternative strategies, with respect to a classic ARQ scheme, which can reduce the fraction of failed nodes during the periodic window. To this aim, a comprehensive set of experiments, also supported by simulations, is presented, showing the effectiveness of the proposed methods.

5.1 The need of retransmissions

In a realistic industrial scenario it is assumed that the presence of machinery, fixed and moving metallic object, moving persons, and also an harsh electromagnetic environment, with a radiated disturbances and different coexistent radio communication systems, will likely introduce an increased probability of error in wireless factory automation control systems. The problem of a limited reliability with respect to the one provided by wired communication systems is therefore a limiting factor for the adoption of a wireless network for distributed data sensing through small, low-power wireless sensors.

A IEEE 802.15.4 based network is here considered. To adapt its behavior to an actual implementation of a well-known industrial communication system, the PAN coordinator is implemented in order to cyclically serve each connected slave, in a typical "Round Robin" fashion. The wireless standard considered [31] uses link-layer retransmission as an error correction technique, and since the communication rate (*i.e.* the modulation performances) is fixed, the most effective control knob for an increased reliability of the communication, in the sense of an increased number of successfully delivered packets, is a clever (incisive) management of these retransmissions.

In the implementation of a periodic communication cycle, subdivided in two time windows allocated one for periodic traffic and one for aperiodic one, a notable difference between wired and wireless systems is that in the latter one extra time has to be allocated in the periodic window in order to accommodate retransmissions for cyclic traffic. In the most typical yet straightforward approach ARQ-similar schemes are adopted. In practice, for each node a limited number of transmission trials are available, which are handled successively, *i.e.* without serving other nodes in between two trials. If all the available trials have been exhausted and the controller failed to receive the requested data, the node is declared *failed* in this cycle. In any next cycle, the same node will be queried in the same fashion, without taking into account its past behavior. Nonetheless, it might well happen that, due to the coherence time of the wireless channel, that is, the time-varying nature of errors and interference, that failed node can be successfully served in the upcoming cycle. This method will be henceforth called the BIR algorithm, in uniformity with the names chosen for the strategies that will be introduced later.

However, the extra time provided to accommodate retransmission trials can be used in a more incisive way. In particular, exploiting the notion of coherence time of the wireless channel, a very effective approach is to not insist in making all trials for a node subsequently, but to defer them to a later moment, in the chance to find the channel in a "good" state.

The proposed techniques, therefore, will exploit this argument in order to increase the probability of success. Besides, another valuable strategy is to reuse the trials that "good" nodes have not used in order to serve "bad" nodes with an increased number of trials, thus avoiding to assign a fixed number of trials per node. A scenario with an unbalanced interference is also considered in the following, *i.e.* a scenario in which a node experiences systematically an higher impact of interference than others, for example considering that the interferer is closer to that node. In this interesting case, it will be demonstrated that reordering the polling sequence in a simple adaptive way increases significantly the overall performances of all the proposed techniques, and even for the basic BIR algorithm.

5.2 Considered retransmission strategies

The different retransmission strategies proposed through this work are here described.

5.2.1 Bounded Immediate Retransmission (BIR)

This is a baseline scheme mimicking the atomic behavior of packet transactions in PROFIBUS. It works as follows: for each node at most K trials can be made. If the first trial fails, the controller immediately starts the next trial, until either all K trials have been exhausted, the controller successfully receives a response, or the periodic window ends. When processing for node i has ended and there is still time available, processing for the next node j starts. Node j is not served in between two trials for node i.

Technically, this scheme is a simple ARQ scheme with a bounded number of immediate retransmissions. The efficiency of this scheme (and all other schemes as well) over fading channels (or other types of non-stationary channels like the interference channels considered here) will depend on the channel coherence time, i.e. the time for which the channel does not change its characteristics [55]. When this time is large enough to cover several packets in succession, then a packet transmitted to a node with a currently bad channel will likely fail. The BIR scheme furthermore has the disadvantage that the trials *not* used by one node cannot be used by another node.

5.2.2 Unbounded IR (UIR)

This scheme is similar to the BIR scheme in that all trials for node i are carried out in succession without serving other nodes in between, but in the UIR scheme there is no limit on the allowed number of trials – the controller can perform as many retransmissions for a node as would fit into the periodic window. This scheme, which is of little practical value, has been included as a baseline scheme to stress the effect of bounding the number of retransmissions. One obvious problem with this scheme is that when one of the stations polled early in the cycle suffers from a bad channel, the controller will spend a lot of time with this station at the cost of not treating the remaining stations.

This scheme is not expected to be of any help to overcome the chosen interference pattern, nor are better performances than BIR expected. In particular, this scheme is very unfair, wasting possibly all transmission attempts to fulfill the polling of the first node.

Nevertheless, averaging over all the nodes, it may show better performances in terms of cycle loss.

This strategy has been included to stress the effect of bounding the number of retransmissions.

5.2.3 Queued Retransmission (QR)

The controller maintains a FIFO queue of node addresses. When this queue is empty or the periodic window is exhausted, the controller stops working on periodic data exchanges. Otherwise, the controller removes the first entry, say address i, from the front of the queue and performs one single trial towards node i. If this trial fails, a new entry for i is appended to the tail of the queue and the next head-of-line entry is served. At the beginning of the periodic window the queue is initialized with all N addresses in the sequence from 1 to N.

In this approach first all N nodes are tried once. The successful nodes are not considered any further during the current cycle, and for the failing nodes a new trial is appended to the queue. This has two effects. First, the spacing between the first trial and the first retransmission is larger on average than with immediate retransmissions. This allows to deal with larger channel coherence times. The spacing between the second and third trial is in general smaller and random. Secondly, the number of retransmissions that can be performed for one node might be larger than K, provided other nodes have required fewer than K trials. A further benefit of this approach is its simple implementation.

5.2.4 Adaptive versions (AQR, AUIR, ABIR)

In the previous three strategies (QR, BIR, UIR) the sequence in which nodes are initially polled is fixed from $1, 2 \dots N$.

In these adaptive schemes the controller maintains for each node long-term statistics about the relative frequency with which one trial is successful. At the start of a cycle, instead of initializing the polling sequence with $1, 2, \dots, N$, it is initialized according to the average success probability for a trial: nodes having the largest success probability (and hence requiring the fewest average number of trials until success) are tried first. By picking the nodes with the smallest average numbers of trials first, the average number of nodes that can be successfully served within the periodic window is maximized.

The adaptive version defined here are primarily intended to improve the fairness of each scheme. However, better performances in term of unserved nodes are in general also expected. Nevertheless, it is worth noting that the adaptive version of BIR does not produce an increase in the number of served nodes, as the number of trials reserved to a node is bounded and unused node trials are definitively lost after its poll round is completed.

The adaptive schemes are intended to be implemented in devices with reduced computational and memory capabilities, so the estimation complexity as well as memory usage should be limited. Therefore a simple exponentially weighted moving average estimator has been adopted, that is, in other words, a first order infinite impulse response (IIR) filter. The estimator of the trial success probability after trial $n + 1$ is:

$$\bar{s}_{n+1} = \alpha\bar{s}_n + (1 - \alpha)s_{n+1} \tag{5.1}$$

where $\bar{s}_n$ is the estimated probability after n trials, s_{n+1} is the outcome (success or failure, represented as 1 and 0, respectively) of the $n+1$-th trial and α is a parameter. The estimator update for each node is performed with a product and two sums, and requires one memory position to store $\bar{s}_n$, the previous step estimation. In the practical implementation a fixed point arithmetic is used, so a normalization and truncation of all the entries in Eq. (5.1) are performed. The motivation for choosing this type of estimator is that it "forgets" the past after some time, which is appropriate for time-varying wireless channels. The factor α controls how quick the memory vanishes. Throughout the rest of the work a factor $\alpha = 0.9$ is chosen, *i.e.* most weight is put on the history and new observations influence the estimation only with a weight of 10%. Other values of α have been experienced, but $\alpha = 0.9$ gave a good compromise between stability and agility of the estimator. With the truncation introduced, the memory length results in about 45 step (about 17 seconds), preventing sudden variation in the polling sequence but allowing a proper steady state behavior.

The frequency by which new observations are taken for one channel does in general not allow to track quick channel variations, like e.g. generated by fast fading. However, the presented results show that already this simple estimator can yield very good gains for static scenarios in which the nodes suffer from interference by different degrees.

It is worth noting that wireless fair scheduling schemes discussed in [48, 73] exploit the fading nature of the wireless channel to serve different nodes when the channel towards the current node becomes bad (see also [8] for an early facet of this idea). These schemes have some similarities to the queueing-based schemes proposed in this work, but most of the publications regarding wireless fair queueing concentrate on aggregated throughput (possibly subject to fairness constraints) and do not consider packet deadlines. Furthermore, they often assume independent channels among the involved stations. To the best of the author knowledge, the queueing-based retransmission schemes proposed here have not been described so far in the context of industrial communications, nor have they been investigated under external interference, where several channels suffer simultaneously (i.e. are correlated).

5.3 System model and assumptions

5.3.1 Network model

A system of one central station (the *controller*) and a number N of sensor / actuator stations (the *nodes*). It is simply said *station* if a distinction amongst node and controller is not made.

All stations are stationary. The set of nodes does not change over time and all node addresses are known to the controller. It is assumed that the network has already been set up, i.e. all nodes have successfully associated to the controller.

All stations share a common wireless medium (i.e. work in the same frequency band) such that each node can communicate with the controller. On the physical layer that all stations are assumed to be compliant to the IEEE 802.15.4 standard. They operate in the 2.4 GHz ISM band and use all the same modulation scheme and transmit power. The controller is considered as a master and the nodes are considered as slaves. A polling-based scheme is adopted, i.e. a scheme in which the controller sends a *request-frame* to an individual node (possibly carrying some output data for this node) and the node immediately answers with a *response-frame* (again, possibly carrying some data). The controller is able to determine whether it successfully received a response-frame or not, i.e. it can obtain binary feedback. It is assumed that a client-server interaction pattern is adopted, i.e. the data generated by one node is not immediately relevant to any other node, but only to the controller.

An important assumption is that the controller does not perform any carrier-sensing operation before transmitting a request-frame, and vice versa a node does not perform carrier-sensing before transmitting a response packet. This is commonly satisfied in industrial master-slave systems, but has the disadvantage that the system has no chance to respond directly to external interference.

5.3.2 Organization of the cycle

The cycle organization is fairly straightforward and shown in Fig. 5.1 on the facing page. A cycle has a fixed duration (*cycle period*) of Δ_c seconds. A cycle starts with a synchronization or *beacon packet* broadcast by the controller to all nodes. The main purpose of this packet is for the nodes to maintain time synchronization with the controller. It is not strictly

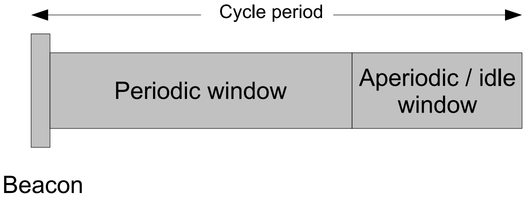


Figure 5.1 – Organization of transmission cycle.

necessary for a node to receive each beacon and in the following no cares about the beacons are put anymore.

The remaining cycle is sub-divided into two windows. In the first window, the *periodic window*, the controller handles all periodic traffic. The size of the periodic window, Δ_p, is chosen so that each node can be polled at least once (involving transmission of a request packet and a response packet) and furthermore some additional time budget is available to perform retransmissions (which again consist of request and response). All these retransmissions, however, must take place within the periodic window. It may happen that for one or more nodes the controller has not obtained any response at all at the end of the periodic window. In this case, the nodes are said to have *failed* during the cycle. In practical implementations, nodes failing successively for a number of periodic windows would be removed from the polling sequence. However, it has been chosen to avoid this policy, *i.e.* nodes are always polled, no matter how often they have failed in the past.

The following aperiodic/idle window is of no further concern to us – it can (like e.g. in WorldFIP) be used to carry out aperiodic message exchanges, or it can be a pre-defined idle phase which the controller can use to process the responses. The only relevant assumption about this window is that it is in no way available for handling periodic traffic anymore.

It is further assumed that all stations are active throughout the beacon transmission and the following periodic window, i.e. they do not perform any sleeping activities during this window.

5.3.3 Traffic model

In order to verify the performance indexes of interest for the present work, it suffices to assume a very simple traffic model: in each cycle the controller produces for node k output data of length $l_{o,k}$ and vice versa node k produces input data for the controller of length $l_{i,k}$. In addition, a fixed-length packet header and trailer consisting of l_h bytes is added to the user data. A simplifying assumption has been made, that is, all output data have the same length, i.e. $l_{o,k} = l_o$ for some constant $l_o > 0$ and all k. Similarly, for the input data it

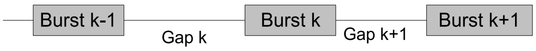


Figure 5.2 – Interference pattern

is assumed $l_{i,k} = l_i > 0$.

Finally, there is a total number of $N \times K$ trials available ($K > 1$ and K being an integer) during the periodic window.

5.3.4 Interference model

In both experiments and simulations the interference is generated artificially from a relatively simple stochastic process. It is furthermore assumed that external interference is the dominating source of channel errors, other sources like fading have not been considered. The interferer does not perform any carrier-sensing on the common channel, so that the interferers behavior is not influenced by the controller and the nodes.

The interference process is depicted in Fig. 5.2. It alternates between *bursts*, during which the external interferer transmits, and *gaps*, where the external interferer is silent. The interferer is static and uses the same transmit power in all the bursts. It is assumed that the interferer uses a completely different rate and modulation scheme than our IEEE 802.15.4 stations, so that the interferers signals can be regarded as white noise for the controller and nodes. Therefore, the interferers activities correspond to a time-varying noise level. This is common assumption for modeling external interference.

To keep the generation of the interference simple, the gap lengths are modeled as an independent and identical distributed (IID) sequence of exponential random variables of a given average gap length. The burst lengths are iid and have a uniform distribution drawn from a given interval $[b_l, b_u]$. This choice reflects maximum uncertainty about the length of the interference bursts.

There is, however, a crucial difference between the proposed experiments and the simulations regarding the directivity of the interferer. In the experiments a directional antenna with a relatively narrow beam has been used. This means that the reception of the node onto this antenna points is distorted, but the transmissions of this node can still be heard by the other nodes when they are outside the antenna beam. In contrast, in the simulations the interferer is assumed to have a perfectly omni-directional antenna. For this reason the experimental and simulation results are not directly comparable.

5.3.5 Major performance measures

The main performance criterion is the *delivery rate.* More specifically, the *downlink delivery rate* gives the average fraction of nodes which successfully receive their output data within the periodic window. Vice versa, the *uplink delivery rate* gives the average fraction of nodes which successfully deliver their data to the controller within the periodic window. Since controller and nodes use a request-response communication pattern, the downlink delivery rate is always at least as large as the uplink delivery rate. Therefore, it has been put major attention on the uplink delivery rate, or equivalently, the (average) number of nodes for which no uplink packet is received (denoted as *average number of unserved nodes*). A related measure is the *cycle loss*, which for an individual node denotes the percentage of cycles where the controller does not successfully receive an uplink packet.

A second important criterion is the *fairness index* of each scheme. For a specific setup with N nodes, at the controller for each node the average time between arrivals of responses is computed, and the fairness index is defined as the difference between the maximum and minimum of these averages (taken over all nodes). Please note that the fairness index is expressed in seconds, smaller values indicate better fairness.

5.4 Measurement results

5.4.1 Measurement setup

Table 5.1 – Relevant network parameters.

MAC-Parameters	**Value**
Fixed overhead size	152 bits
Beacon payload size	48 bits
Downlink payload size	48 bits
Uplink payload size	168 bits
Max. trials for BIR scheme	2
Cycle time	400 ms
Periodic window size	330 ms
IF parameter	**Value**
Avg. gap length	$\{10, 30, 50\}$ ms
Burst length distr.	$\{U[1, 10], U[2, 20]\}$ ms

Table 5.1 resumes the most relevant parameters for all the experiments presented hereafter in this chapter. The cycle time has been set to 400 ms. For one single trial (consist-

ing of request and response) the parameters for the packet lengths and the transceiver turnaround times have been chosen so that one trial takes 20 ms. Fig. 5.3 depicts a typical poll cycle: the poll request has a length of 200 bits, whereas the response from the node is 320 bits long. Summing up these quantities and considering two transceiver turnover times we obtain a total of 4984 bits, that correspond to about 20 ms at the maximum rate for the IEEE 802.15.4 standard.

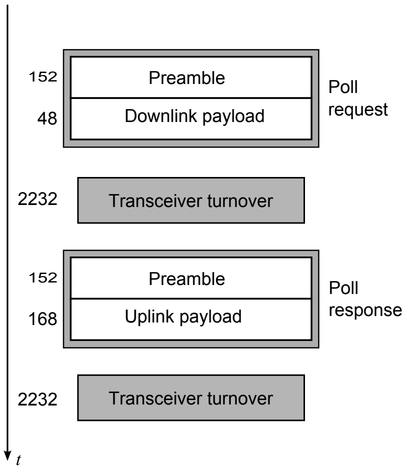


Figure 5.3 – Data exchange in a typical poll cycle. Numbers represent bits. Data rate is 250 Kb/s. The total number of bits in a poll cycle is 5032, resulting in about 20 ms for a cycle.

The periodic window has a length of 330 ms, which is sufficient to accommodate 16 trials. The overall setup consists of the controller, $N = 8$ nodes and an interfering node. With eight nodes to serve and 16 trials available we have $K = 2$. The controller has been placed in the center (position $(0, 0)$), the nodes have been placed equidistantly on a circle of radius r around the center. The rightmost node is placed at position $(r, 0)$, the leftmost node at position $(-r, 0)$.

Fig. 5.4 on page 100 shows the measurement setups that have been used in the experiments. All the experimental parts of this work have been performed in a non-anechoic room, so non-ideal effects of a real-life environment could not be excluded. However a preliminary scan of the chosen band showed no spurious emissions. The experiments are carried out with TelosB compliant motes platforms carrying ChipCon CC2420 transceivers that are IEEE 802.15.4-compliant and which operate in the 2.4 GHz ISM band.

Considering the position of the interferer two settings have been considered:

✓ In the first setting the interferer is placed very close to a particular node. This is called, somewhat imprecisely, the *one-disturbed-node* scenario. It represents a scenario in which the nodes of a network suffer from interference differently, i.e. some nodes are more disturbed than others.

✓ In the second setting the interferer is placed very close to the controller, so that all nodes are disturbed in the same way by the interferer and the controller is disturbed most. This is called the *all-disturbed-nodes* scenario.

In the one-disturbed-node scenario the chosen node positions causes the rightmost nodes to be most affected by interference, while the leftmost node is the least affected. The interferer follows the pattern described in Section 5.3.4 on page 96.

In the all-disturbed-nodes setup the nodes have been evenly spaced on a semi-circle of 60 cm radius, with the controller in the center. A directional antenna has been placed far away from nodes in order to satisfy the far-field hypothesis. It has been verified that the received signal strength at each node is approximately the same. This allowed to assume that, in the area of sensors, the electromagnetic field is isotropic. In our one-disturbed-node setup the interference has been directed only toward one node, namely node 3. Both setups are jointly show in Fig. 5.4 on the following page. Two antennas have been sketched, a switch plugs the RF interference signal into the left antenna in the one-disturbed-node setup, with the beam directed only toward the node close to the antenna. The interference power has been regulated to disturb only communications regarding this node.

The TinyOS 2.1 operating system has been used to develop the application of each node. This application implements the overall polling scheme (beacon transmission, poll-request and poll-response generation on controller and nodes, respectively) and four of the different retransmission strategies used in this work, namely UIR, BIR, QR and AQR. The underlying protocol stack is basically the default protocol stack delivered with TinyOS 2.1, but the CSMA/CA and CCA mechanisms have been modified to effectively get rid of the carrier sense functionalities.

The interference signal behavior has been described in Section 5.3.4 on page 96, and its parameters have been varied as specified in Table 5.1 on page 97. The real signal used in these experiments was an AWGN signal, with a bandwidth of 5 MHz, centered over the same IEEE 802.15.4 channel used for the WSN (in this case the 26-th, i.e. 2.48 GHz), produced by a RF Agilent E4433B signal generator. To generate the described pattern a pulsed mode of the RF generator has been used, exploiting a baseband signal generator as trigger. The baseband signal generator is able to reproduce an arbitrary waveform from a succession of values from a file. Thus firstly this succession of points has been generated, sampling the stochastic process described in Section 5.3.4 on page 96, then this has been loaded into the baseband signal generator, to reproduce a signal according to the succes-

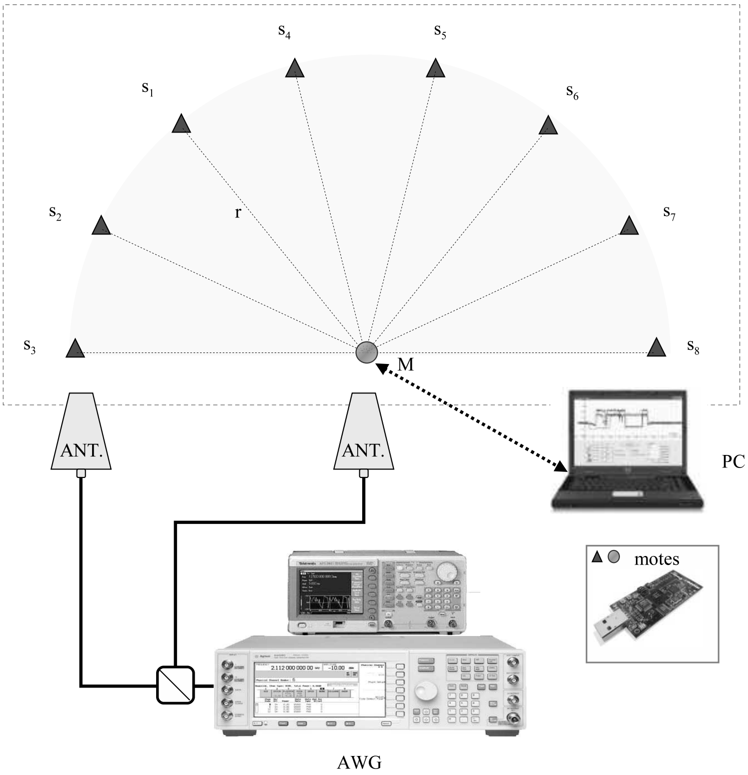


Figure 5.4 – Measurement setups used in experiments.

sion. Finally, this signal has been connected to th RF generator and used as a trigger to switch on and off the radio.

Note that in this approach the interferer does not react (by not using any carrier sense mechanism) to the traffic generated by the sensor network.

The RF signal produced has been irradiated with a directional antenna placed behind the controller, with the main lobe covering the WSN area. With this setting, in the all-disturbed-nodes scenario an interferer can prevent both uplink and downlink transmission.

The software produced and installed on sensors provides also a "sense" functionality, allowing each mote to sense the channel for 30 seconds, collect RSSI measurements and return a packet with the computed statistics. This mechanism estimates the noise/interference floor of each node, and is used to verify the IF level sensed at each node before experiments. This allows to set up the correct power of the IF signal, in order to get an SNR value sufficient to block any exchange of packets if the IF is active.

5.4.2 Results

In the presented experiments we have fixed some parameters with respect to the simulation, in order to shorten the time requirements for the tests. First, we have fixed the IF power, which was set to 8 dBm at the instrument side (RF signal generator). With this setting we achieve a $SNR < 0$ dB both at the controller and at the nodes. The channel has been chosen as channel 26 of the standard, centered at 2.48 GHz. This is the highest available IEEE 802.15.4 channel, which does not overlap with the commonly chosen Wifi channels at our department. The MAC parameters have been chosen according to Table 5.1 on page 97. The IF signal has been reproduced with the same characteristics as presented in Section 5.3.4 on page 96. We have considered four different mean values for the exponential random variable (representing gap spaces between bursts), namely 10, 20, 30 and 50 ms. For the sake of conciseness, the results shown here uses only the short interference burst length, *i.e.* the burst lengths are randomly chosen from $U[1, 10]$ ms (see Table 5.1 on page 97).

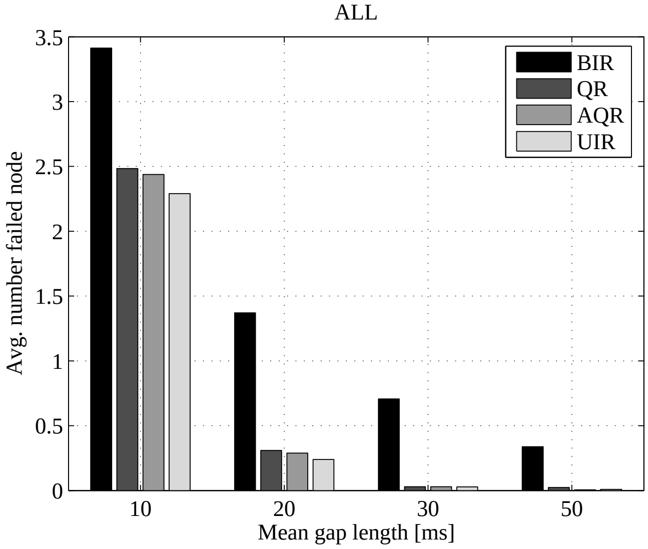


Figure 5.5 – Average number of unserved nodes in the all-disturbed-node scenario, $U(1, 10)$ ms interference bursts, average IF gap time on the x-axis

In order to obtain comprehensive statistics from the data collected during the measurements, we have chosen to run a single experiment for approximately 15 minutes, allowing the transmission of M=2300 cycles. Each experiment has been carried out with three repetitions, spaced in time, to avoid correlations with environmental parameter variations. In

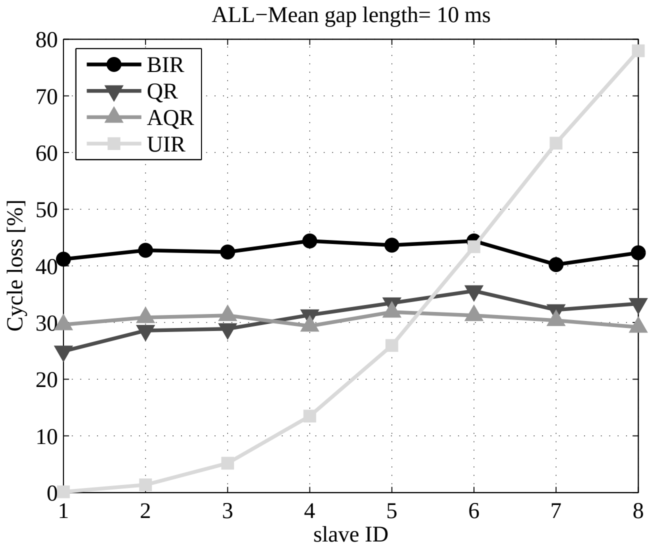


Figure 5.6 – Cycle loss for all-disturbed-node scenario, $U(1, 10)$ ms interference bursts, 10 ms average IF gap time *vs* node ID.

all cases the results of the repetitions were very close to each other, so the first repetition has been used for statistics and plots.

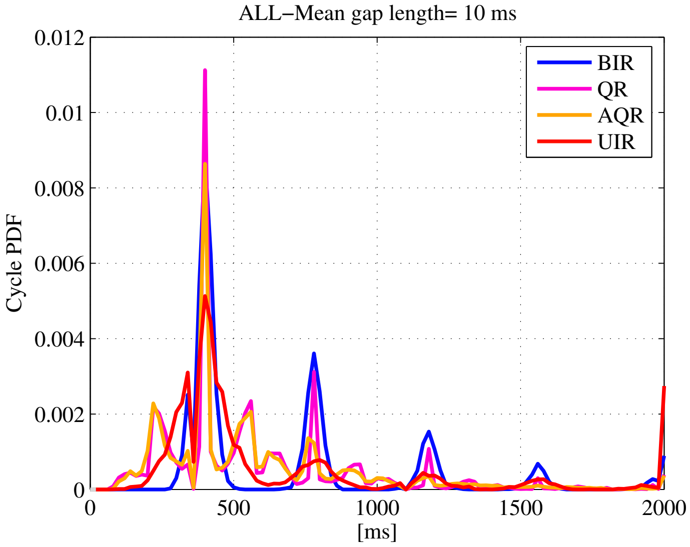


Figure 5.7 – Measured uplink cycle iat probability density function: average over the nodes.

The results regarding the one-disturbed-node scenario are very close to the ones obtained in [25] and hence are not reported here. In the following only the all-disturbed-nodes scenario results are shown.

Fig. 5.5 on page 101 shows the average number of unserved nodes for the different retransmission schemes. Please note that this is a mean value taken over all the nodes and is not useful to stress inequalities and unfairness in the performance of different nodes. Fig. 5.6 on the facing page specifically underlines such a perspective instead. It depicts different cycle loss experienced by each node using different retransmission schemes.

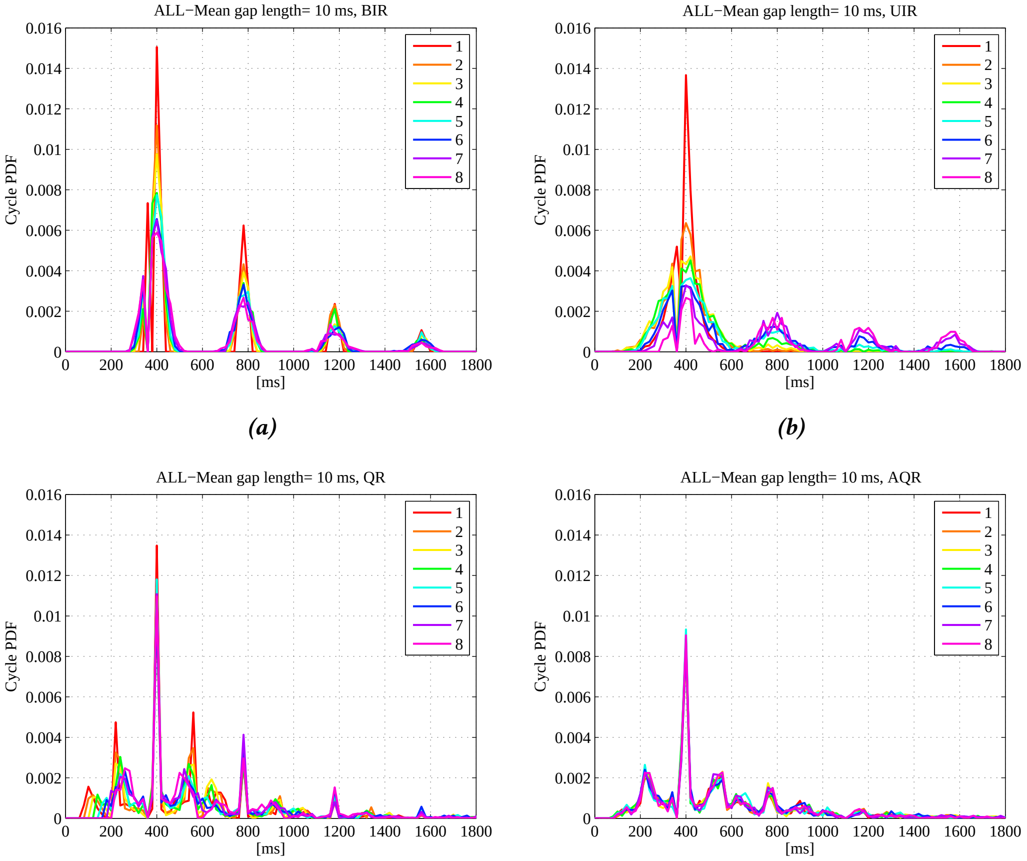
The following points are noteworthy:

✓ In terms of the average number of unserved nodes (Fig. 5.5 on page 101) the BIR scheme is the worst approach, since it allows only K transmission attempts to each node without a dynamic allocation of trial budget. The AQR scheme is slightly better than the QR scheme and UIR performs as the best one. As expected, longer interference gap space allows better performance.

✓ A more in-depth analysis (Fig. 5.6 on the facing page) shows that UIR is not a fair approach because the nodes experience very different cycle loss: node number 8 reaches 80% of cycle loss, and in general the cycle loss increases with the node index in our setup. This behavior can be explained by the fact that this strategy spends an unbounded number of trials for each node to get a successful transmission, therefore last nodes will have less trials to carry out their data delivery. In other words, this means that devices with high node ID would seldom, if ever, be correctly polled.

✓ The results reported are obtained with the IF average gap value of 10 ms and hence in a highly interfered scenario. Less aggressive (i.e. with IF average gap value of 20, 30 and 50 ms, as can be seen in Fig. 5.5 on page 101) interference patterns show similar results even if difference among polling policies are less sharp.

In a polling system it is often important to guarantee that inter-arrival times between uplink (and downlink) packets are fixed or at least have low variance. To this aim in the following for each node the uplink inter-arrival time (iat) is analyzed through a normalized histogram, also referred to as EPDF. Our histograms reflect the raw difference between timestamps of uplink packets, taken immediately after their arrival. The main lobe of the pdfs is around 400 ms (the cycle time), but if some cycle is lost, the pdfs show spikes around multiples of 400 ms. Fig. 5.7 on the facing page shows the EPDFs for the different retransmission schemes. In these EPDFs the samples for all nodes have been combined. As for Fig. 5.5 on page 101 this averaged graph can not appreciate the fairness of the different



(a)

(b)

(c)

(d)

Figure 5.8 – Measured uplink cycle iat probability density function *vs* node ID (mean gap length 10 ms) and for: (a) BIR policy; (b) UIR policy; (c) QR policy; (d) AQR policy.

polling policies. Fig. 5.8 on the facing page compares the same pdfs, showing the behavior of each node. It is worth noting that:

- ✓ Fig. 5.7 on page 102 shows that BIR scheme is the most "deterministic" one in the sense that the spikes around the multiples of 400 ms are very sharp. The high cycle loss of BIR is reflected in the number and amplitude of the side lobes. QR and Adaptive QR (AQR) are very similar to each other. They follow the spikes of BIR, but have more probability mass in between these spikes. AQR's lower cycle loss causes its secondary spikes to be smaller than those of QR. UIR has a very wide main lobe, but seems not to have much probability mass in secondary lobes .

- ✓ Fig. 5.8 on the facing page underlines differences in a per-node fashion. Fig. 5.8a on the preceding page shows for the BIR scheme that for increasing node IDs the EPDFs become less tight around the multiples of 400 ms: peaks are getting lower and wider. Fig. 5.8b on the facing page shows a huge unfairness for the UIR scheme: node 1 is quite deterministic and loses no packets, while node 8 has many secondary wide lobes. Fig. 5.8c on the preceding page shows for the QR scheme that most variation occurs around the main lobe. The symmetric peaks around the main lobe shift and get lower increasing node ID. Fig. 5.8d on the facing page shows that AQR policy is the most fair: the pdfs of the different nodes are almost identical.

In a polling system it is also of interest to describe the maximum delay experienced using a particular polling scheme. To this extent the maximum delay for each baseline scheme is analyzed through a bar plot. This interesting quantity has been calculated resuming data on inter-arrival times, and considering only those polling cycles leading to a successful transmission of data to the master node. The maximum delay has been defined as the maximum inter-polling time between those polling cycles for each node. Fig. 5.9 on the next page shows this plot for each average gap length. The bar plot highlight again that the BIR scheme has the most deterministic behavior, in the sense that it defines an upper bound for the delivering delay of a packet unrelated from the maximum polling window time. Conversely, QR and AQR schemes, while do not change their performances with the gap length increase, saturate the polling window time with their transmissions. This expense, however, leads to an higher delivery rate for these two polling scheme. Furthermore, UIR scheme seems to increase its performances with the increase of the gap length. Observing Fig. 5.5 on page 101 we note that for gap length higher than 10 ms, this scheme approaches the delivery rate of AQR and QR, while the upper delay bound decreases. This however must be weighted with the unfairness of UIR scheme.

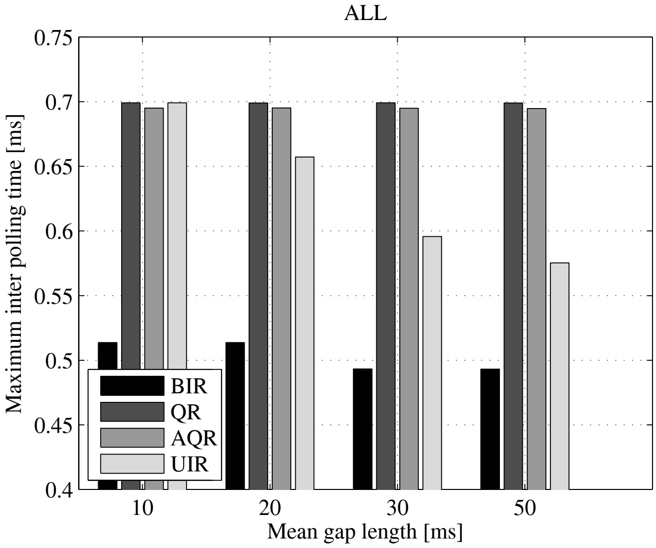


Figure 5.9 – Maximum inter polling time, taken among all the correct polling cycles.

Summarizing, it can be said that the worst approach is BIR if we are interested in the packet delivery measure. Conversely, it shows the best behavior in term of "determinism" of the delivery, in the sense that if a transmission is successfully carried out we can also state what is the maximum delay for that transmission. This is a very interesting behavior in an industrial network context, especially if we are addressing real–time wireless networks.

The performances of QR and AQR are very similar, but AQR pays when analyzing fairness (as will be confirmed by simulation results of the following Section 5.6 on page 109). UIR is a good solution in mean, but is the most unfair approach and probably the only one that could not be used in a real industrial system.

5.5 Experimental characterization of the simulator

The experimental campaign just described provided a number of results, whose interest could be beyond the scope of performance analysis for the proposed retransmission strategies. Indeed, the behavior of the simulator presented in Chapter 4 on page 67, while have been already verified being in adherence with some preliminary experimental results, could be furthermore characterized by means of the experimental outcomes presented in this chapter.

Considering the simulator ability to investigate cross-layer interference, a comprehensive comparison could lead to an accurate tool, effective for the tuning of parameters and for the interference effects evaluation. It could be exploited in order to assess some performance figures of wireless sensor nodes networks, when deployed in an harsh environment.

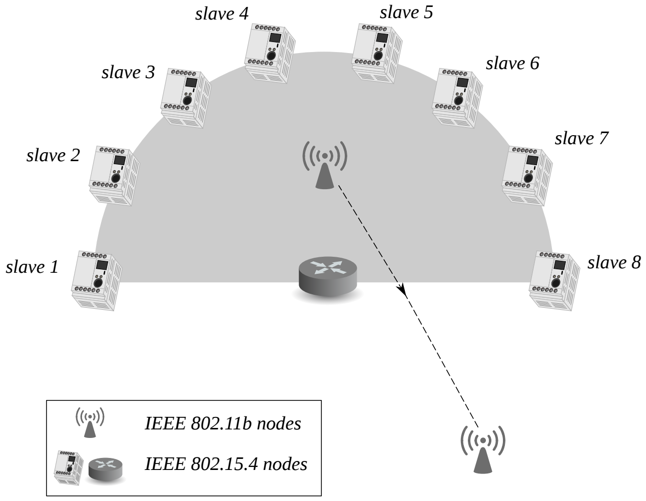
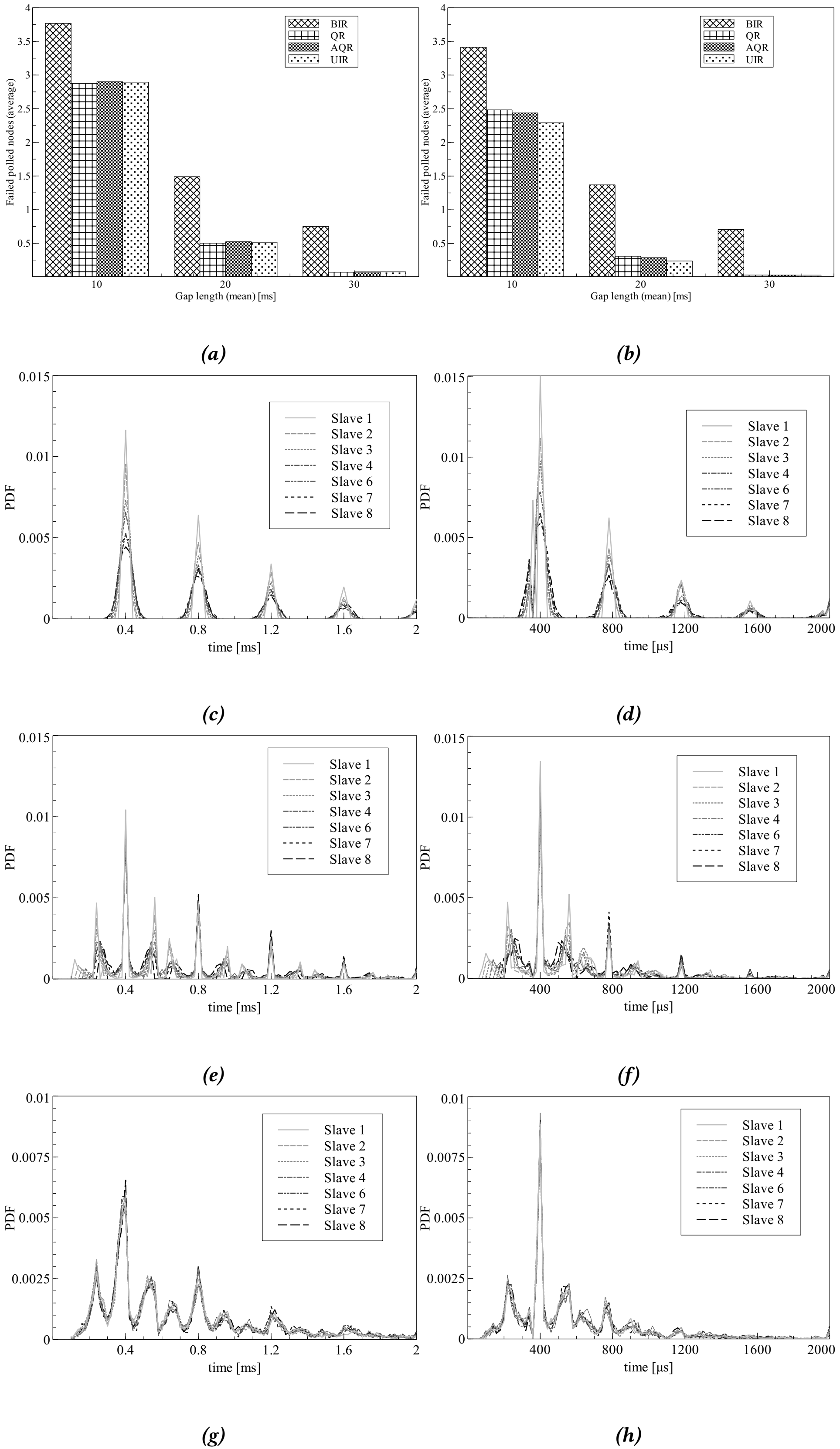


Figure 5.10 – The simulated testbed for retransmission strategies analysis. It reflects the same experimental setup of Fig. 5.4 on page 100, in the *all-disturbed-nodes* scenario.

Therefore the same communication system described in Section 5.3 on page 94 is here investigated via the proposed simulator. The same environmental conditions and parameter values have been set in the simulator configuration, thus trying to emulate the same network analyzed through measurements. Fig. 5.10 shows the simulated setup, here the interferer is a IEEE 802.11 network, where the CSMA/CA mechanism has been disabled, *i.e.* it is not sensible to transmissions by IEEE 802.15.4 nodes, to emulate the same conditions of measurements. The interferer is realized by an up–link transmission from a node placed in the middle of the semi–circle. This has been verified to reach the most similar conditions of the experimental results.

In the following a side-by-side comparison of the outcomes from simulator and measurements is presented. In Fig. 5.11 on the next page, therefore, the left-side graphs refer to data obtained through simulations, while the right-side refers to measurements outcomes and in particular reflects the same data set which generates the graphs in Section 5.4 on page 97.

A first set of results obtained through the described simulation environment shows the average number of nodes whose data have not correctly been received by the master. The direct comparison between Fig. 5.11a on the next page, relevant to the simulation outcomes, and Fig. 5.11b on the following page highlights the very good agreement between



(a) *(b)*

(c) *(d)*

(e) *(f)*

(g) *(h)*

Figure 5.11 – Simulation and experimental results for the IEEE 802.15.4 described network. On the left side the outcomes from the simulations, on the right side the same indicator obtained through the experimental measurements.

simulation outcomes and measurement data. Moreover, it is worth noticing that the simulator mildly overestimates the number of unserved slaves, that anyway is a conservative approach from a designer point of view.

Another meaningful set of data comes from the EPDF of the polling time of the network, which enables the characterization of the interarrival time between two packets originated by the same node, and its associated variability. As already seen in the previous Section, if packets are correctly received by the master, the interarrival time is expected to be "deterministic" and equal to a cycle duration, *i.e.* 400 µs. Instead, in the case of one or more cycle loss, the interarrival time becomes a multiple of the cycle time. Moreover, in case retransmissions occur, the EPDF should highlight a variability around a specific peak. In the light of this considerations, the simulation results for the presented retransmission strategies are presented in Fig. 5.11c on the preceding page, 5.11e and 5.11g. They can be directly compared to, respectively, Fig. 5.11d on the facing page, 5.11f and 5.11h. The comparison between the various curves shows that a practitioner may benefit from simulations in order to speed up a design that will hence require less in-the-field experiments in order to fine tune the communication network to the actual needs. It is anyway once again pointed out that measurements are still required in order to advantageously characterize a number of hidden effects that a simulation should include, as quoted in Section 4.6.1 on page 84.

5.6 Fairness index

In this section the results obtained about the fairness index are shown, which are obtained through a simulation-based performance study.

The fairness indices of the different schemes for long and short interference bursts, and an interference gap time of 10 ms, are shown in Fig. 5.12 on the following page and Fig. 5.13 on the next page, respectively. Taking aside the UIR scheme (which for interferer transmit powers between -60 and -50 dBm assumes extremely high values, not displayed properly), it can be seen that the AQR scheme has the best fairness, followed by the ABIR and, for all but the highest interferer transmit powers, the QR scheme. The AUIR scheme, which showed the best performance in terms of the number of unserved nodes, now shows the worst performance in the interesting range between -60 and -50 dBm, and never approaches the fairness performance of AQR and ABIR. The transitional behaviour (the "bump") of all adaptive schemes between -60 dB and -50 dBm can be explained from fluctuations in the adaptive packet loss rate estimator (compare Eq. (5.1)): for large enough interferer transmit powers the interferers transmissions “reliably” destroy transmitted packets, therefore the estimator directly observes the interferer statistics. In the range with low interferer transmit powers (between -50 and -60 dBm) not all packets are destroyed by the interferer, and therefore the observations of the interferers behaviour become themselves noisy.

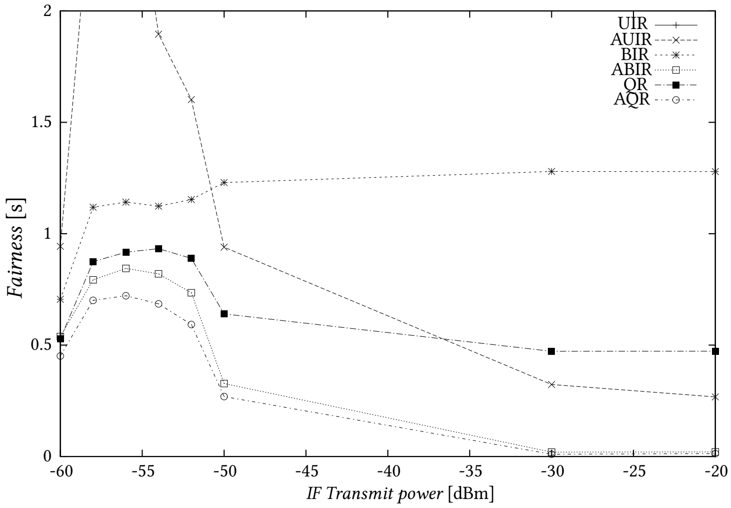


Figure 5.12 – Fairness index in the one-disturbed-node scenario, long interference bursts, average IF gap time = 10 ms

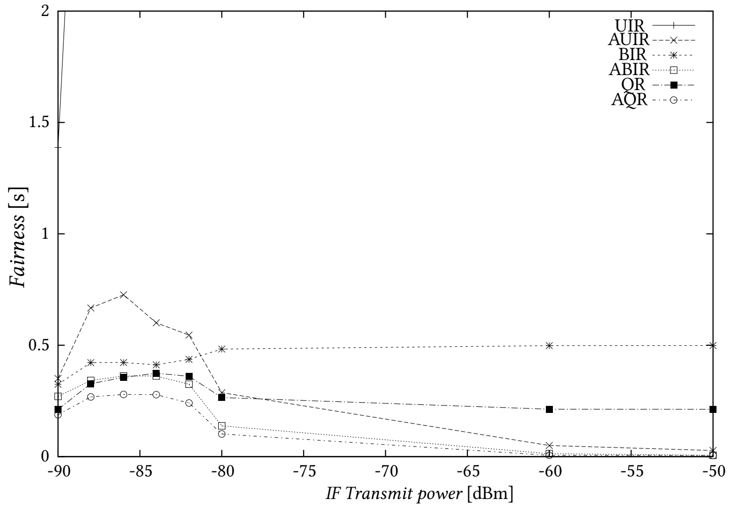


Figure 5.13 – Fairness index in the one-disturbed-node scenario, short interference bursts, average IF gap time = 10 ms

Some results can be described also for the case of the one-disturbed-node scenario. Fig. 5.14 on the following page and 5.15 show the average number of unserved nodes for the short and long average interference burst lengths respectively and for an average interference gap length of 10 ms. In Fig. 5.16 on page 113 we show similar results for the case of 30 ms average gap lengths and short bursts. The results for the 50 ms case look very similar to the curves for the 30 ms case and are not shown here.

In all these figures, a “transient” zone is noticeable in the leftmost part of the graphs. For all schemes there is significant performance variation for interferer transmit powers between -60 and -50 dBm, while for interferer transmit powers larger than -50 dBm the performance of each scheme does not change much anymore. This is due to the increasing number of nodes impaired by the interferer. For values of interference greater than -50 dBm all the nodes are involved and hence the number of unserved nodes saturates.

For the 10 ms average gap duration case and short interference bursts, it can be seen in Fig. 5.14 on the following page that for the shorter burst lengths the BIR and ABIR schemes show consistently the worst performance due to the lack of trial re-use. The UIR scheme performs better since it allows as many attempts as necessary to leap over the (short) interference burst. The performance of UIR degrades as the interference bursts become longer, as is exemplified in Fig. 5.15 on page 113 for the case of long interference bursts and 10 ms average gap duration. Here, UIR has the worst performance (followed by the BIR and ABIR schemes) since for long interference bursts UIRs persistence on serving a bad node has consequences for all following nodes, the last ones possibly not being polled at all.

The AQR and QR schemes show very similar behaviors. AQR does not significantly improve the basic QR scheme since the queuing mechanism is already a “self adapting” policy that postpones the impaired nodes to be served after the good ones and the only difference between the two schemes is the starting order, that is scrambled just after few steps. The AUIR scheme shows the best performance especially in the “transient” region. Apart from BIR and ABIR, all the other schemes perform similarly; this proves further the effects of an unbounded number of retransmission trials.

It is interesting to note that the relative performance of the UIR and (A)BIR schemes depends on the length of the interference bursts. The difference between the UIR and the QR scheme points to the value of increasing the spacing between the first trial and the first retransmission. It is also very interesting to note the significant improvement in performance of the AUIR scheme as compared to the UIR scheme. It should also be mentioned that the performance of all schemes depends, for given interference gap length, of course on the average interference burst lengths, so that longer bursts lead to overall reduced performance. This can be seen from comparing the ranges on the y-axis of Figs. 5.14 and 5.15.

For 30 ms and 50 ms average gap times (see Fig. 5.16 on the facing page) and short interference burst lengths the BIR and ABIR schemes show the worst performance, whereas the other schemes do not differ very much. This figure highlights the effect, already stressed in Fig. 5.14, of bounding the number of retransmissions for each node. In fact, two distinct set of curves are noticeable. The same is true for the longer bursts at 30 ms average gap duration and also for long and short burst lengths at 50 ms average gap duration.

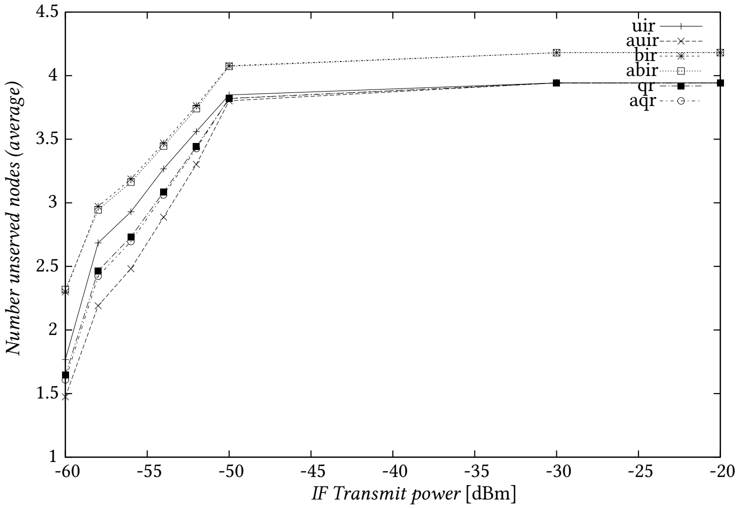


Figure 5.14 – Average number of unserved nodes in the one-disturbed-node scenario, short interference bursts, average IF gap time = 10 ms

5.7 Summary

In this chapter, some new and original retransmission schemes for polling–based networks are presented, for the specific case of IEEE 802.15.4 WSNs. The aim is improving the network reliability and fairness among nodes.

The presented techniques have been firstly detailed and the system model described. Subsequently the setup for measurements have been presented. A comprehensive experimental session has been conducted, showing the performances of the proposed techniques. The results served also as a way of characterization of the simulator presented in Chapter 4. The latter one showed very good adherence to the experimental results. Finally, some

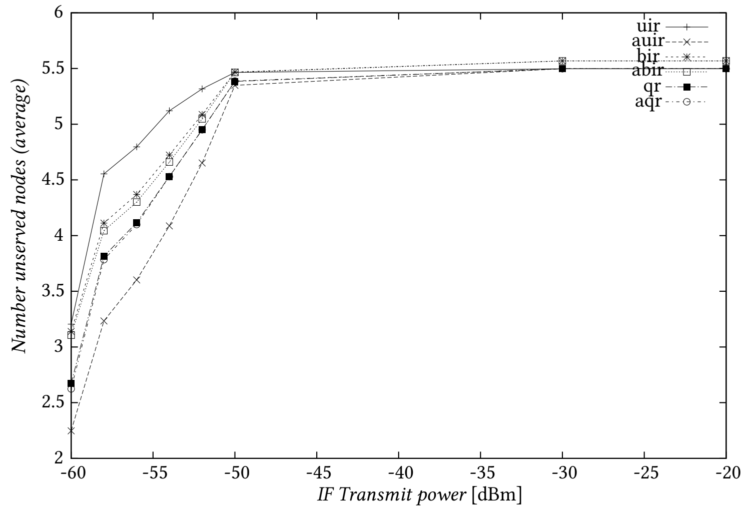


Figure 5.15 – Average number of unserved nodes in the one-disturbed-node scenario, long interference bursts, average IF gap time = 10 ms

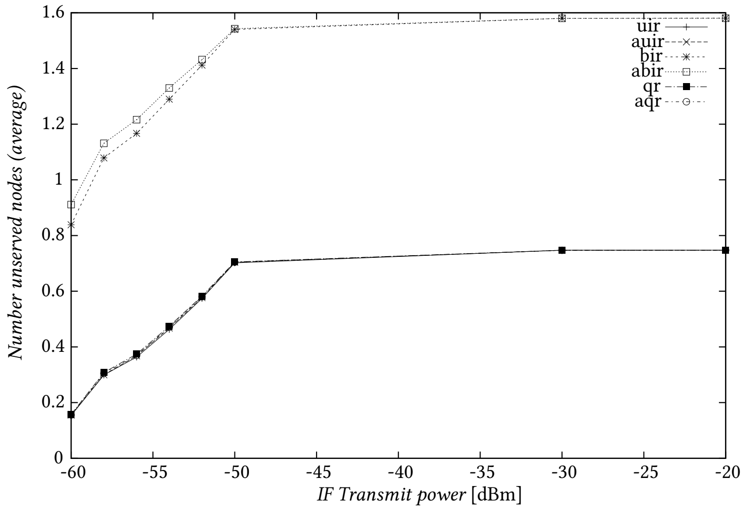


Figure 5.16 – Average number of unserved nodes in the one-disturbed-node scenario, short interference bursts, average IF gap time = 30 ms

outcomes from simulations have been presented, where in particular the fairness of each scheme has been discussed.

As a conclusion, when all nodes are distorted in the same way (as in the all-nodes-disturbed scenario), there are practically no differences between UIR, AUIR, QR and AQR, but all these schemes are significantly better than BIR and ABIR. When the interference situation among the nodes becomes heterogeneous, the performance of the schemes starts to differentiate as well and the AUIR scheme shows the best performance, followed by AQR and QR. On the other hand, the performance of the BIR and ABIR schemes is either the worst one or the second-worst one, so they should be avoided in practice. The likely explanation for this is (A)BIRs inability to use unused trials from good nodes to increase the number of trials for nodes with worse interference conditions.

Chapter 6

Characterization of Network components

One of the core issues concerning network analysis through simulations is the accuracy of the simulator itself with respect to the specific behavior of the physical models. Indeed, the latencies introduced by the radio chips switching from transmitter to receiver mode, the delay experienced in forwarding a packet from a network segment to another, the effect of in-band interference on the quality of received signal are some significant examples.

In this perspective the behavior of network components, which is typically neglected in theoretical studies and related simulations, is of particular interest in the field of real-time wireless communications. Indeed, some studies [37] have already dealt with the characterization of some network apparatus, such as *switch*, for wired real–time communication, *i.e.* latencies, jitter, etc.

This chapter, based on the experience of [61, 60], presents a comprehensive experimental analysis of the internal behavior of network components, in the case of a wireless communication, for example to realize an extension of a wired RTE pre-existent segment.

The deployment of a wireless extension of a wired link requires some efforts in converting the information being transmitted from the frame format relevant to the wired segment to a suitable frame structure for the wireless one. The conversion, moreover, could take place at various level of the ISO/OSI stack. An *Intermediate System* is the specific device that takes care of the operation needed to realize the interconnection. In the deployment of a wireless extension of a wired RTE link, the interconnection takes place at the Data Link-Layer (DLL), that is the lowest possible layer, just above the physical one, namely by using purposely designed devices, called *bridges.* For the specific case of an Ethernet (IEEE 802.3) network linked to a wireless IEEE 802.11 network, the bridge is an AP.

The schematic picture represented in Fig. 6.1 highlights that a protocol conversion is

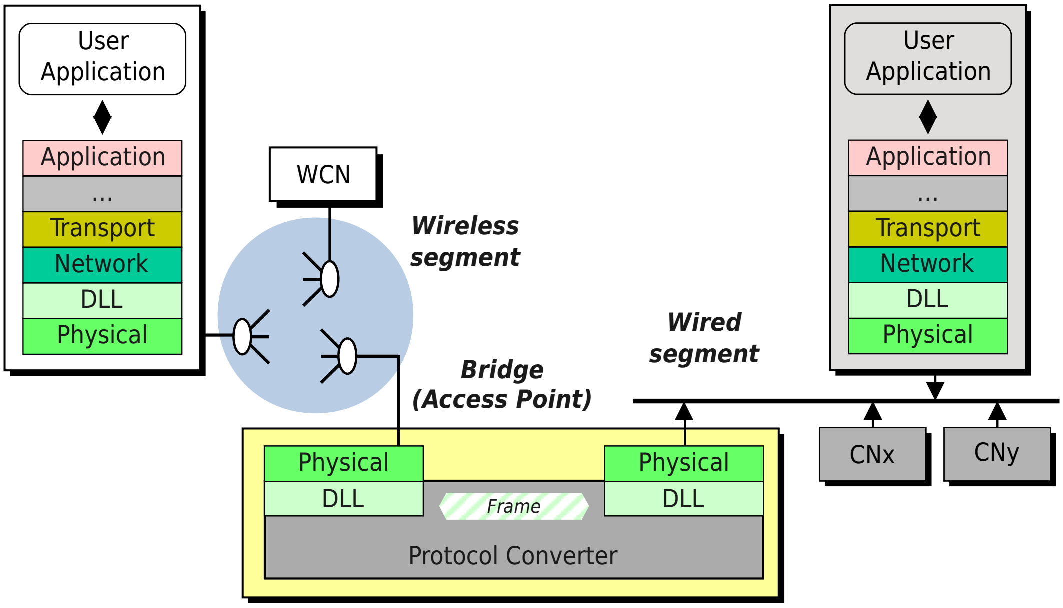


Figure 6.1 – An AP serving as a "bridge" between a wireless and a wired network.

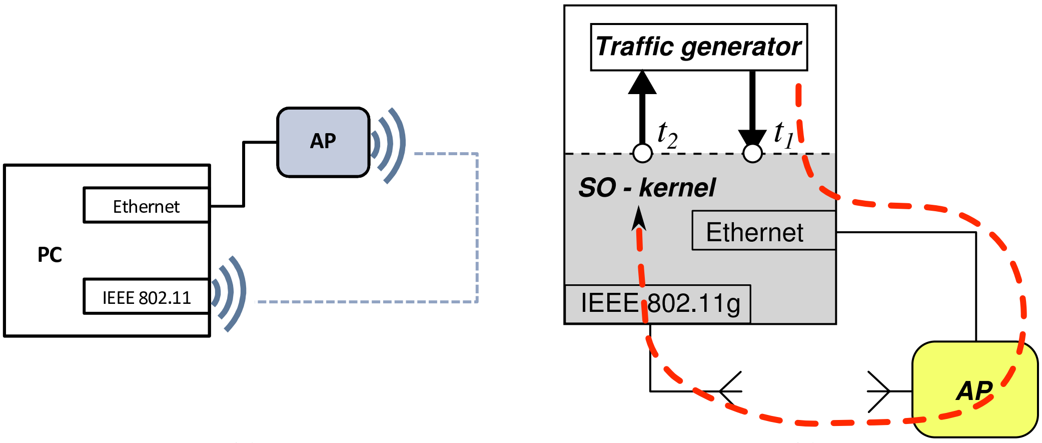
carried out by an AP. Moreover, considering the actual differences in the medium access strategies on the two different media and the wireless channel behavior, an AP is also typically provided with some queuing mechanism for incoming packets. Furthermore, one has to consider that in a complex device such as an AP, an operating system is implemented in order to provide a simple configuration interface, energy management schemes, etc. Therefore, it is easy to speculate that such a device will introduce some non-negligible delay, with some superimposed jitter. Nevertheless, the introduced delay will presumably have a random behavior hence being described by a precise stochastic process to be defined. In this light, the need for an accurate characterization of such communication devices emerges, and the obtained models will serve the scope of accurately simulate their behavior, which is an essential requirement for the analysis of industrial wireless communication networks.

The goal of the present chapter is to present the methods adopted to obtain the raw measurements of the introduced delay, along with the experimental set-up and the instrumentation employed. The outcomes of the measurement campaigns will serve as the basis for an original method to obtain the statistical distribution of the delays, based on a deconvolution process and a measurement by substitution approach. This method allows an accurate characterization of a network component through a common PC, while providing the delay contribution of only the Device Under Test (DUT) itself, and neglecting the contribution of the PC operating system and communication cards.

6.1 Measurement setup

In order to provide an accurate characterization of network communication devices, it is mandatory to realize a measurement system able to replicate the same environmental conditions as well as the same traffic profile. It is also needed that the system remains adequately simple in order to keep under control the high number of variables influencing the measurments.

A simple communication system has been found and realized, and it is shown in Fig. 6.4a. Namely, it is composed by a controller node and a single passive station. The former periodically queries (alternatively, *polls*) the latter with a packet of fixed length, and the slave reacts to such requests sending back some random data (*i.e.* representing the outcomes of some "virtual" measurements).



(a)

(b)

Figure 6.4 – Experimental setup for measurements through a PC.

While the controller node, as it often happens, is found on the wired link, the communication scheme allows for a slave that is connected to the system through a wireless link, thus forming an *hybrid network*. This has been implemented by means of two very popular communication systems. The wired segment is a switched Ethernet 100BASE-TX (*i.e.* full-duplex, 100 Mb/s); the segment connects directly the PC communication board to the AP. Therefore the access to the physical medium by the communication board is immediate and no additional delays are introduced by the IEEE 802.3 MAC protocol. The wireless segment is realized through an IEEE 802.11 WLAN, adopting the ERP–OFDM amendment of the standard [32], commonly known as IEEE 802.11g. In this case, the medium access is regulated by the well-known DCF which makes use of a CSMA/CA technique. Details about this protocol, and DCF in particular are found in 3.3.2 on page 57.

In this campaign of measurements, the measurand is represented by the time difference between the instant in which the polling request arrives at the slave and the instant in which the same packet leaved the master. It appears clear that such a measure would be influenced by misalignment between master and salve clocks. Therefore, in order to avoid this effect, thus achieving the best precision in the time measurement both the controller and the passive wireless station were implemented on the same PC, analogously to what described in [61, 11, 60].

The employed PC was running a GNU/Linux Operating System (OS)[1], running a Linux kernel version 2.6.39. The hardware subsystem was build on a professional workstation, equipped with a Intel® Core™2 Quad Q9400 2.66 GHz, and 4 GB of system memory. The communication boards was, namely, a *3Com 3c905C-TX/TX-M [Tornado]* Ethernet board and a *D-Link RT2561* IEEE 802.11 board. The IEEE 802.11 communication board was configured to avoid the execution of tasks that eventually could cause delays, such logging, power management, etc. Such tasks was therefore disabled via software. Unfortunately, in this case it was impossible to fix the communication channel for the board, which was continuously scanning for other active WLANs in the environment. This, however, seems to not had influenced the measurements.

Two different APs were alternatively employed in the experimental set-up. In particular, we used a general purpose device, namely the *3Com Office Connect 3CRWE454G75*, and a *Siemens SCALANCE W784-1* device, specifically tailored for industrial applications.

A traffic generator software, purposely designed to the scope of these measurements, was used to generate fixed-length packets and collects statistics and measurements. This software component directly communicates with the operating system kernel and uses low-level directives to access the Central Processing Unit (CPU) registers and counters. Actually, the measurements of time is performed by the software using the CPU Time–Stamp Counter (TSC) as reference, and converting the number of clock "ticks" to time using the processor frequency. The code used to get a TSC reading is implemented through direct calls to the CPU registers. The relevant fragment of code is:

```
static __inline__ unsigned long long rdtsc(void)
{
```

[1] For the sake of completeness we adopted the Ubuntu Desktop 11.04 distribution, freely available on the Web, at http://www.ubuntu.com.

```
  unsigned hi, lo;
  __asm__ __volatile__ ("rdtsc" : "=a"(lo), "=d"(hi));
  return ( (unsigned long long)lo)|( ((unsigned long long)hi)<<32 );
}
```

The traffic generator sends packet directly to the LLC layer, thus avoiding any high level protocols (UDP/TCP, IP, etc) overhead and delays. It is therefore independent of network addresses of devices, and sends and receives packets through the specified OS network interface (for example, in a common Linux system it could be eth0, wlan0, ...). Specifically, in the scenario realized for the experiments that will be presented in the following, the software delivers the packet in the outgoing direction from the Ethernet card, and waits for the packet arriving from the other side, back into the wireless card. Packets are therefore forwarded by the AP, in a communication loop. This is described by Fig. 6.4b on page 117. Finally, many communication parameters can be easily chosen in order to emulate different design solutions of a factory automation system, such as the length of packets, and the inter-arrival time between two consecutive packets. For instance, Fig. 6.5 shows the generated cyclic traffic pertaining to a typical automation network, where a new packet is sent after t_{cycle} seconds.

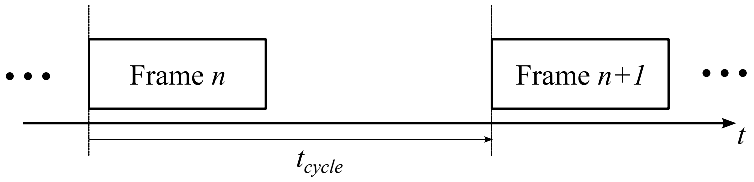


Figure 6.5 – The generated traffic profile.

Since in a typical real-time industrial communication system scenario a small amount of data is exchanged among nodes, the packets payload β was set to the size of a minimum Ethernet frames (β =46 Bytes long). They specifically contain a sequence number for tracking purposes, and are padded with random data to reach the desired length. The transmission follows a periodic profile, in which a packet is generated and sent each τ seconds.

During measurements, the AP was located very close (less than one meter) to the wireless slave, minimizing path loss effects and propagation delays. In each experiment the transmitting power was set to the maximum allowed value (*i.e.* 20 dBm) and the selected communication channel was the number 6, centered at 2.437 GHz, since it was not affected by transmissions of different (possibly interfering) communication systems already present in the laboratory. Therefore, it is assumed that the experiments were conducted in an interference-free environment, and this was experimentally verified through a spectrum analyzer continuously monitoring the radio channel. Given these almost ideal conditions, and considering the periodic basis onto which the query of the passive station was carried out, the backoff procedure typical of the IEEE 802.11 MAC DCF was never executed by the AP, since the medium was always free for (at least) a DIFS when it tried to access it.

Table 6.1 reports most of the communication parameters used in this paper. They are relevant to both Ethernet and IEEE 802.11 systems as well as to the specific application we implemented.

Table 6.1 – Communication parameters

Description	Value
IEEE 802.11g transmission rates	54 Mb/s
IEEE 802.11g transmission channel	6
Distributed Inter-Frame Space (DIFS)	28 µs
Short Inter-Frame Space (SIFS)	10 µs
Slot Time	9 µs
Max number of retransmissions (N_{max})	7
CTS protection	Switched off
Encryption	Open-system
Application sending periods (T_p)	3, 5, 10, 15, 20, 30 ms
Time to transmit the Ethernet frame	5.76 µs
Time to transmit the IEEE 802.11 frame @54 Mb/s	42 µs
Time to transmit the IEEE 802.11 ACK @24 Mb/s (t_{ack})	36 µs

6.2 Characterization of an AP: measurements through a simple PC

In a first phase of this analysis, the simple communication system just described, and shown schematically in Fig. 6.4a on page 117 has been directly adopted to characterize the delays introduced by the AP. The testbed allows an easy measurement of the one-trip delay experienced by a packet, from the master to the slave stations, which is, referring to Fig. 6.4b on page 117, equal to $D_{PC} = t_2 - t_1$. Actually, in this first set of experiments, the instants t_1, t_2 are read from the traffic generator output file, where each exchanged packet (transmitted or received) is registered along with the time instant of such operation. In this context, the expression[2] of the time employed by a frame to go from the controller (master) to the passive wireless station (slave) results

$$D_{PC} = t_{tx,e} + D_{AP} + t_{DIFS} + t_{tx,w} + D_{MS} \tag{6.1}$$

In Eq. (6.1) $t_{tx,e}$ represents the time necessary to transmit (on the cable) the Ethernet frame from the controller to the access point, considering the type of link and the characteristics of the standard IEEE 802.3 [29]. D_{AP} is the delay introduced by the access point, and is therefore the objective of the measurement, *i.e.* the measurand. t_{DIFS} is the time that has to be waited by an IEEE 802.11 station (the access point in our scenario) before transmitting and $t_{tx,w}$ is the transmission time (on the air) for the IEEE 802.11 frame, from the access point to the passive station. These two terms account for time related to the wireless section. Finally, $D_{MS} = D_{OS} + D_B$ accounts for the latencies introduced by both the controller and the passive station hardware. In particular, for the measurement system described, it includes the OS delays D_{OS} (task scheduler, processing time, etc) and those introduced by the communication boards D_B employed. It is worth remarking that the times $t_{e,tx}$, t_{DIFS} and $t_{w,tx}$ are constant (with a very limited uncertainty) and once specified the number β of data bytes (payload) carried by the frame, their values can be deduced from both the IEEE 802.3 [29] and the IEEE 802.11 [32] specifications. On the contrary, the latencies introduced by the devices D_{AP}, D_{OS} and D_B are random variables since their values may depend on events driven phenomena such as operating system latencies, buffer queues, conflicting tasks, etc. In this first stage, however, D_{OS} may be neglected since, as discussed in [11], it is mainly related to the PC operating system latencies that are, typically, in the range of a few microseconds. In Table 6.1 on the preceding page the deterministic contributions value appearing in Eq. (6.1) is found.

In order to describe the behavior of the considered access point, we carried out extensive measurements of the frame transmission time D_{PC}, under different traffic conditions. From the measured value of D_{PC}, a number of interesting considerations can be drawn, having in mind that the goal is to obtain the value of D_{AP}, and it is D_{AP} itself, as will be shown later, the principal contribution of the obtained measurement. In the following, therefore, any time D_{PC} is used and its properties commented, the same property will be reflected on the D_{AP} value itself.

The experiments were carried out varying both the sending period τ and the number of data bytes β transmitted. The outcomes of this experimental campaign allow for a thorough analysis of the measured delay D_{PC}. For example, the EPDF can be calculated in order to

[2] Capital letters are used for random variables.

give a clear picture of the random behavior of D_{PC}. This normalized histogram is shown in Fig. 6.6. It shows only a subset of all the chosen sending periods τ, and respectively, 5 ms, 10 ms, 20 ms and 30 ms, while keeping the frame length fixed to 46 bytes. To give a complete overview of the phenomena, the related basic statistics of the measured samples are summarized in Table 6.2 on the next page and also shown in Fig. 6.7 on the facing page. Here, the solid black line indicates the mean values of D_{PC}, while the dark and light gray rectangles display, respectively, the 25th–75th and the 10th–90th percentile intervals. The adoption of such intervals would highlight the multi-modal behavior of the D_{PC} statistic, as well as the presence of a quite long tail of samples.

A further set of measurements was carried out in order to evaluate the behavior of D_{PC} versus the number of data bytes transmitted. In this case, we set the sending period to 3 ms, and varied the frame payload length β. The outcomes of this second experiment are provided in Fig. 6.8 on page 124.

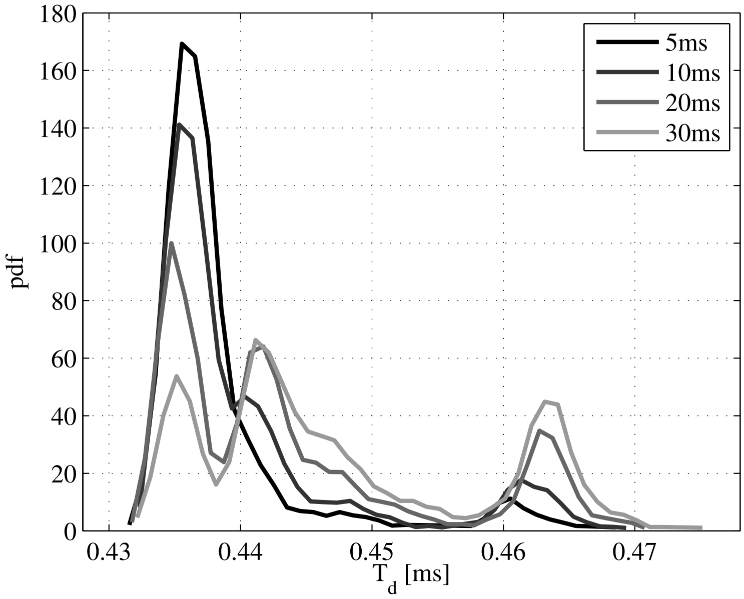


Figure 6.6 – Empirical pdfs of the delay introduced by the access point.

Some preliminary observation can be drawn considering the Figures 6.6 and 6.7 and Table 6.2 on the facing page.

1. *D_{PC} is definitely a random variable.*
   As expected, the delay introduced by the access point is event driven and the values it assumes are not only related to the frame length.

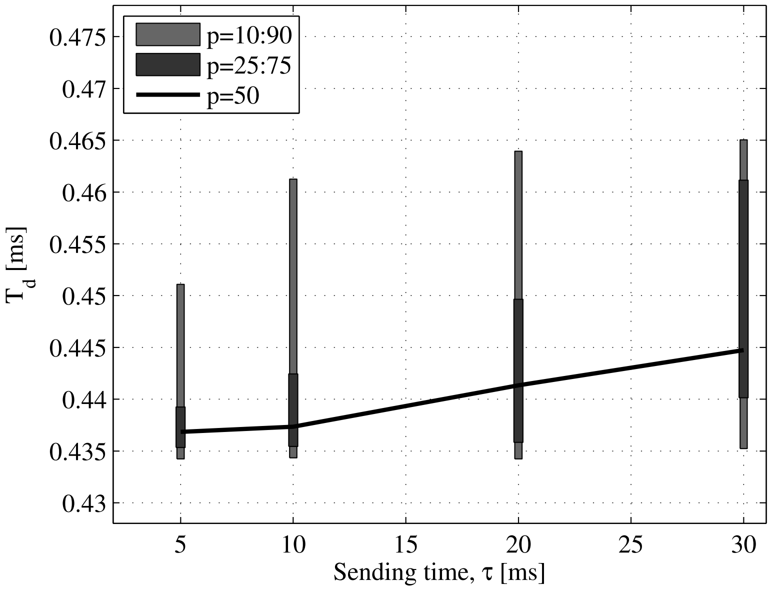


Figure 6.7 – Access point delay vs. sending period ($\beta = 46$ bytes).

Table 6.2 – Statistics of the access point delay

τ [ms]	**Mean** [µs]	**Std. Dev.** [μs^{-1}]
5	431.4	6.95
10	433.2	8.55
20	436.7	10.3
30	440.3	10.8

2. *The randomness of D_{PC} is reduced for lower values of τ.*
   As can be seen in Fig. 6.6 on the preceding page, the lower the sending period, the better the shape of the EPDFs, that tends to an almost deterministic delay. Such a behavior is particularly evident in Fig. 6.7 and is confirmed by the statistics of D_{PC} provided in Table 6.2. The presence of some side modes also stress this conclusion. In practice, it seems that some periodic process on the access point is triggered by the forwarding operation, especially if the sending period is not sufficiently low. It may be concluded that there is a threshold sending period, τ^*, above which the access point needs a greater and more random time to forward the frame toward the wireless passive station. The value of τ^*, clearly, can not be determined precisely. However, for the access point analyzed, looking at both Fig. 6.6 on the facing page

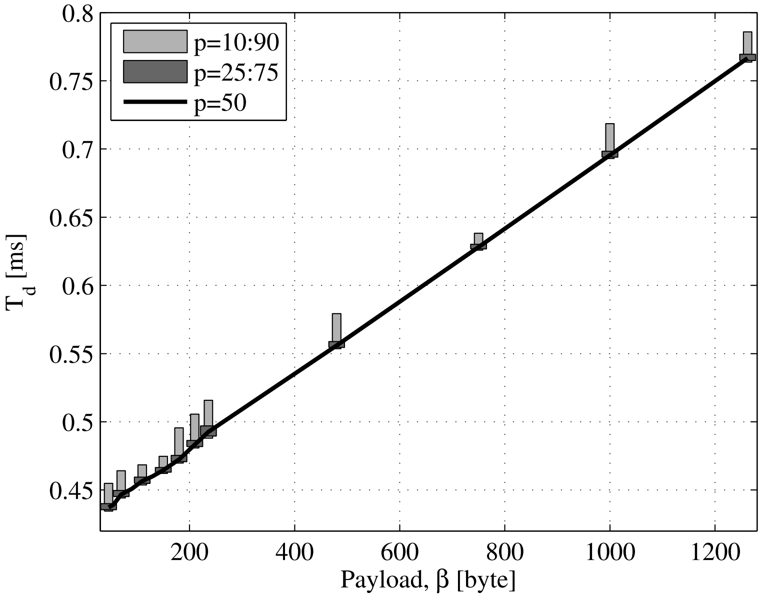


Figure 6.8 – Access point delay vs. number of data bytes transmitted ($\tau = 3$ ms).

and Table 6.2 on the previous page, it may be estimated as $\tau^* \sim 5$ ms.

3. *D_{PC} is lower bounded.*

   Fig. 6.6 on page 122 does not actually show the full EPDF, since the presence of a very long tail, due to some sporadic samples even for long delays (some ms), would have hindered the comprehension of the behavior of D_{PC}. Nonetheless, all the shapes indicate that the delay introduced by the access point is lower bounded. Moreover, the value of the lower bound of D_{PC} is almost the same in all the cases (around 430 µs) that likely represents the physical lowest limit of the access point delay.

4. *D_{AP} linearly depends on β.*

   This latter consideration seems quite obvious, since the management of larger amounts of data clearly may require larger time intervals. Nonetheless, it is interesting to observe the nearly perfect linear relationship between the amount of transmitted data bytes and the delay introduced by the access point. More importantly, it has been noticed that the shape of the pdf of D_{AP} does not change varying the packet length, but depends only on the sending period τ.

6.3 Characterization of an AP: measurements through an oscilloscope

A second step towards an accurate characterization of the AP behavior is represented by the direct delay measurement on the AP board itself. In this analysis we took into consideration the hybrid network of Fig. 6.11a, which is a modified version of the simpler one presented before.

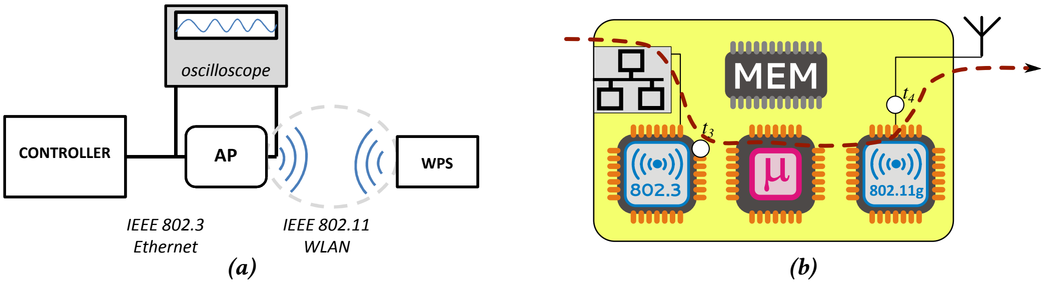


Figure 6.11 – Experimental setup for measurements through a PC.

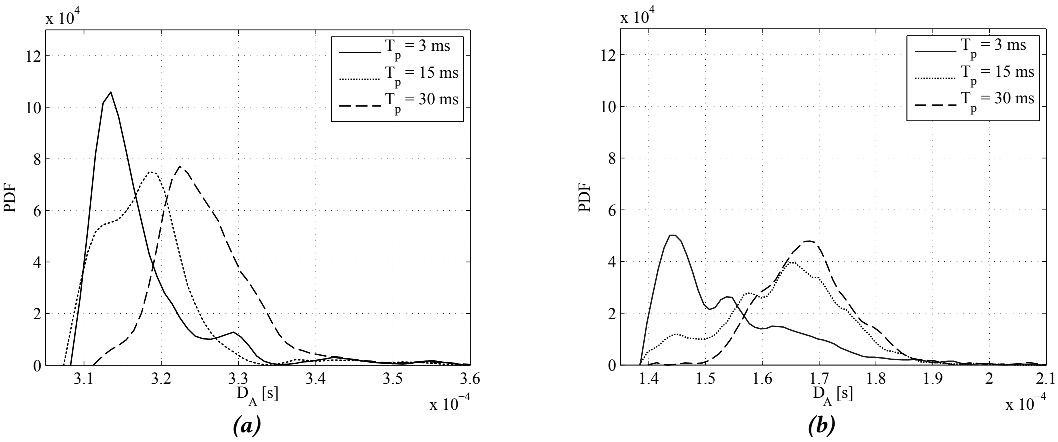
As shown in Fig. 6.11a, a high precision oscilloscope was used, namely an Agilent MSO6012A, characterized by 500 MHz bandwidth, 2 GS/s sampling rate, 8 MSample of acquisition memory for small-scale time measurements. The probes of such a device were connected to the AP internal pins that signal, respectively, the arrival of a frame from the wired segment (instant t_3) and the actual start of the consequent transmission on the wireless segment (instant t_4). The situation is better described by the sketch in Fig. 6.11b. The measurement has been automated via a LabView workstation. The function that returns the delay between two waveform, provided directly by the oscilloscope, was adopted, in this way decreasing the amount of data to be transferred to the workstation.

The use of the oscilloscope in the experimental set-up made possible the exact evaluation of the elaboration time required by the access point to forward the frames from the wired segment to the wireless one. In particular, looking at Eq. (6.1), it allows a direct measurement of $D_{AP}= t_4 - t_3$, avoiding any influence of the other terms, thus decreasing the uncertainty contributions.

The same network parameters described in the previous section has been adopted. The same experiment has been therefore reproduced. This time the results of this measurement campaign are both the output files produced by the traffic generator and the results collected through the LabView environment. Both APs have been measured in this session, again using different sending periods (τ) and payload sizes, following the parameters given in Table 6.1 on page 120. For each measurement cycle, more than 3000 samples from the oscilloscope has been collected.

The results we observed were actually in good agreement with those of the preliminary analysis presented in Sec. 6.2 on page 120. In particular, it has been verified the main result presented in that section, namely, an Access Point introduces a random delays in forwarding a packet from the wired to the wireless segment, even for small, fixed-length packets. Secondly, the delay introduced by an AP has been confirmed to be related to the transmission period, and roughly speaking, the longer the period, the higher the randomness of the delay D_{AP}.

The outcomes obtained with the two APs considered in these experiments can be again be summarized in the form of an EPDF. In the case of a fixed payload of 46 Bytes, and sending periods τ from 3 to 30 ms, the results are given in Fig. 6.12. Specifically, Fig. 6.12a refers to the 3Com AP, whereas Fig. 6.12b refers to the Siemens one.



(a) *(b)*

Figure 6.12 – PDFs of delay introduced by the APs (a): 3Com, (b): Siemens.

Some observations can be inferred from the measurements.

1. *D_{AP} is in the order of hundreds of μs*

   It may be noticed that the delays introduced by the APs are, on average, in the order of some hundreds of microseconds. This is actually a lower bound for any application requiring stringent deadlines, limiting therefore the applicability of wireless extensions in an industrial context to such applications tolerating this delay. Nonetheless, it has to be noted that such delay in some cases is not the critical issue in itself, but it could be the high level communication protocol that, not specifically designed for wireless medium, could suffer for the delay and behave differently from the expected behavior.

2. *D_{AP} for the Siemens AP is lower but less predictable*
   A second result is that the Siemens AP is faster with respect to the 3Com general purpose one. This was in some way expected considering that the first AP is specifically developed for industrial applications. However a result quite surprising is that the AP presents a wider EPDF, that is, the introduced randomness is higher. This is clearly visible comparing Fig. 6.12a on the preceding page with Fig. 6.12b on the facing page. In particular the 3Com AP introduces a delay in the order of 310-320 µs, with a span of 20 µs around the main lobe. The Siemens AP, instead, presents an average delay of 160 µs, with a 40 µs wide main lobe.

An exact model is difficult to obtain. Nonetheless, although a general rule can not be inferred, since an higher number of APs should be compared, there is a clear evidence that the forward operation carried out by these devices requires a non negligible elaboration time that, as such, may reveal dangerous for real–time applications.

The accurate measurement of the AP delays presented in this section allowed also an indirect estimation of the delays introduced by the two boards employed in the experimental set-up shown in Fig. 6.11a. In particular, the time to transmit a frame from the wired board to the wireless one is given by Eq. (6.1), which can be restated as[3]

$$D_{PC} = t_F + D_{AP} + D_B + D_{OS} \tag{6.2}$$

Here, t_F is the time actually necessary to transmit the frame, which is a deterministic value since the transmission strategy (periodic packets from a unique source) implies the absence of contention on the bus (hence the backoff procedure is never carried out); moreover, since the environment was interference free, and the signal to noise ratio was sufficiently high to avoid the corruption of frames (as experimentally verified), no retransmissions occurred on the wireless medium. Thus, t_F comprises both the time necessary to transmit the Ethernet frame (5.76 µs) and the time to transmit the IEEE 802.11 frame (70 µs, as can be derived from Table 6.1 on page 120, including inter-frame spaces).

D_{AP} is the delay introduced by the access point, that is the delay measured through the oscilloscope, whose EPDF is shown in Figures 6.12a or 6.12b.

D_B+D_{OS} accounts for the delays introduced by the measurement system itself, that are not under control and are unknown since the hardware used is not specifically realized for measurement purposes. In particular, D_B is the sum of the delays introduced by both the communication boards on the PC and, finally, D_{OS} accounts for the operating system

[3] As usual, capital letters are used for random variables.

latency. The latter (D_{OS}) may be neglected since, as already pointed out, in the experimental set-up of Fig. 6.11a it results in the order of a few microseconds.

Therefore some preliminary reasoning can be made on the basis of data available through the measurements performed. In order to evaluate the delay introduced by the communication boards, D_B in Eq. (6.2), all the other variables are known along with a good statistical description, since both D_{PC} and D_{AP} have been measured in several experimental sessions, with the above assumptions, for both the access points using different frame sending periods. A simple consideration could help estimate the behavior of the delay D_B: it is indeed clear that the minimum value measured for D_{PC} should correspond to a minimum for D_{AP}, in view of the fact that the high number of samples collected represents a significant set of data for the statistic of both delays. Consequently, looking at the difference between D_{PC} and D_{AP} for all the sending period τ in Table 6.1 on page 120, for which measurements have been taken, one could get a glimpse of the behavior of D_B.

A comparison of the EPDFs shown in Figg. 6.12 on page 126 and 6.6 on page 122 allows an estimate for D_B, which results in a nearly constant value with mean $\mu_{D_B} = 100\,\mu\text{s}$ and a the standard deviation around $4\,\mu\text{s}$.

Clearly, this is a raw estimation, made only in order to get a feeling of the order of magnitude of the delays introduced by the measurement system. However, a more precise estimation can be performed, starting from the available data, bearing in mind the properties of random variables. It will be described exhaustively in the following Section 6.4.

6.4 Method for the estimation of the elaboration delay

In the previous Sections the measurement of the delay introduced by an AP has been described. Both a direct measurement through a digital oscilloscope and an indirect one, using the CPU Time–Stamp Counter, produced a complete statistical description of D_{AP} and D_{PC}. However, specially referring to D_{AP} it is worth remarking that the method described in Section 6.3 on page 125 involves disassembling the AP in order to reach the pins of the integrated circuits on the internal board. Moreover, it requires an high performances oscilloscope and a instrumentation control environment to automate the measurements. In the goal of characterizing a set of communication devices, it would be more useful to perform such measurement through a simpler measurement system, without having to disassemble the AP. To this regard, moreover, some of the APs tested in the experiments were built using Field Programmable Gate Array (FPGA) in place of specific integrated circuit, for example the Ethernet controller etc. This makes useless the approach given because it is not known which are the pins, if any, that carry the control signals needed.

In this Section, an original method is proposed that uses a reference AP to obtain the characterization of the measurement system, and in particular of D_{MS}. Indeed, if the statistical description of D_{MS} is known, any other AP under test can be characterized through delay measurements provided directly by the traffic generator, as shown in Section 6.2 on page 120. This correspond to know the variable D_{PC}. A subsequent application of the algorithm proposed in the following results in a complete characterization of the delay introduced by the AP. One should note that while the initial characterization for the measurement system (D_{MS}) requires a "reference" AP, and a reasonable effort for its characterization, the estimation of the delay introduced by a given AP under investigation does not require specific instrumentation, nor opening the AP under test itself.

In the common scenario of Fig. 6.4a on page 117 the effect of delays are sketched in Fig. 6.13, that should be compared with Fig. 6.4b and Fig. 6.11b for a better understanding. In Fig. 6.13, D_B summarizes the delay contributions from both communication interfaces of the PC (*i.e.* Ethernet and Wireless card). For the sake of clarity, the results provided by this method will include also the delay introduced by the operating system as well. After t_3, which indicates the arrival of a packet at the Ethernet connector, the frame is passed up to the Data Link Layer (DLL). Here a protocol conversion is performed from the IEEE 802.3 format to the IEEE 802.11 one. The resulting frame is then used to generate the physical modulated signal to be transmitted, following the rules found in [32]. The time D_{PC} necessary to receive a frame back on the PC can be decomposed into the following components:

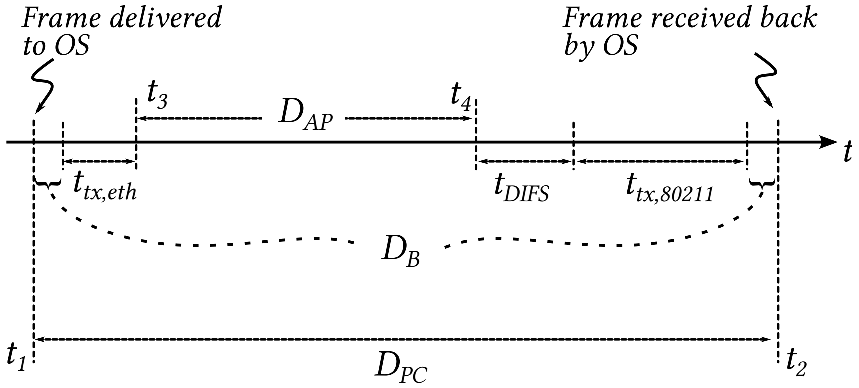


Figure 6.13 – A sketch of the various delay contributions that are measured through the software, and are included in D'_{PC}.

$$D_{PC} = t_{tx,eth} + D_{AP} + t_{tx,802.11} + t_{DIFS} + D_{MS} \tag{6.3}$$

where the one-trip delay is $D_{PC} = t_2 - t_1$, while the delay due to the AP is $D_{AP} = t_4 - t_3$. In Eq. (6.3), the propagation times of the signal both on the wired and the wireless segments are neglected since distances among components was very short. The terms $t_{tx,eth}$, $t_{tx,802.11}$ and t_{DIFS} represent deterministic delays and have already been discussed above, and are reported in Table 6.1.

Since deterministic contributions in Eq. (6.3) can be calculated, or measured [7], the model can be simplified into:

$$D'_{PC} = D_{AP} + D_{MS} \tag{6.4}$$

where $D'_{PC} = D_{PC} - t_{tx,eth} - t_{tx,802.11} - t_{DIFS}$ is a random variable that accounts only for the delays introduced by the AP and measurement system behaviors.

Therefore, we will use the EPDFs derived through the measurement presented in the previous sections. They are shown in Fig. 6.14. Please note that larger bins (5 µs) have been used to derive the EPDFs of this figure.

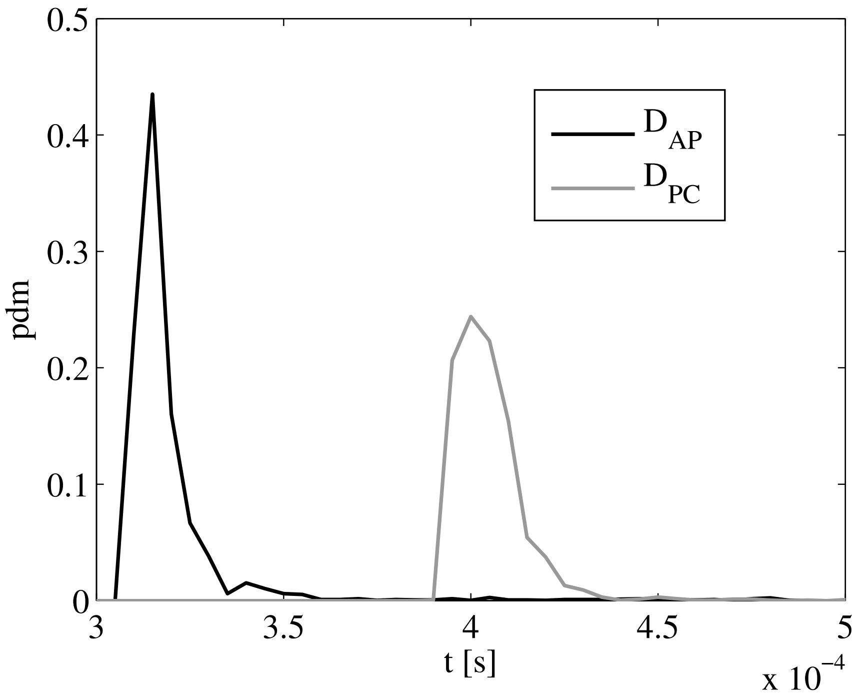


Figure 6.14 – PDFs of the two measured delays.

Considering the EPDFs of the random variables in Eq. (6.4), we obtain that the distribution for the delay measured through the PC should obey to

$$p_{D'_{PC}}(z) = p_{D_{AP}} \otimes p_{D_{MS}}(z) = \sum_{k=0}^{z} p_{D_{AP}}(k) p_{D_{MS}}(z-k) \tag{6.5}$$

where $\otimes$ denotes the convolution operator, and p_X is the PDF of a random variable X.

It is worth to observe that the EPDF of each random variable is estimated from measurements, building an histogram of data with a specific bin width. If the measurement procedure returns data with a granularity of 1 µs and EPDFs are built using bins 5 µs long, there are an implicit filtering on the acquired samples. Such implicit convolution could be described with:

$$p_{D_X}(z) = p_{\overline{D_X}} \otimes F(z) = \sum_{k=0}^{z} p_{\overline{D_X}}(k) F(z-k) \tag{6.6}$$

where $p_{\overline{D_X}}$ is the EPDF of the acquired data, without filtering, and $F(k)$ is defined as

$$F(k) = \begin{cases} \dfrac{1}{l} & 0 \leqslant k \leqslant l-1 \\ \\ 0 & \text{elsewhere} \end{cases}$$

l being the length of the filter.

Since an estimation of the EPDF of the delay $p_{D_{MS}}$ is the object of this method, the AP delay D_{AP} must be subtracted from the total delay D'_{PC}. In terms of EPDF, the corresponding operation is a deconvolution, which is again equivalent to cross-correlating the two EPDFs, namely:

$$\hat{p}_{D_{MS}}(z) = \sum_{k=0}^{z} p_{D'_{PC}}(k) p_{D_{AP}}(z+k) \tag{6.7}$$

where $\hat{p}_{D_{MS}}$ is the estimate of the statistic associated to the delay introduced by the network interface cards of the PC and the OS.

Such estimation provides a characterization of the test bed that can be used in order to infer the delay statistic of any new AP under test, called D_{AP_2}. For the purpose, it suffices to repeat the whole test that led to Eq. (6.7), substituting the access point used in the characterization with the digital sampling oscilloscope with the new one under investigation. It is further assumed that in the meantime the statistic of D_{MS} has not changed, as long as nothing else has been altered in the measurement set up, namely, operating system kernel and drivers, traffic generation software, etc.

After collecting a new set of delay values D_{PC_2} according to the scheme of Fig. 6.4a, from the PDF $p_{D_{PC_2}}$ of these delays, one has:

$$\hat{p}_{D_{AP_2}}(z) = \sum_{k=0}^{z} p_{D_{PC_2}}(k)\hat{p}_{D_{MS}}(z+k) \tag{6.8}$$

which is the required probabilistic description of the response time of the AP under investigation.

To understand the performances of the approach, tests have been made with APs that were characterized both according to the proposed black-box estimation approach, and by using the same direct measurement method used to characterize the reference AP, *i.e.* with a digital sampling oscilloscope.

An example of application of this method is shown here in Fig. 6.15 on the next page, where the estimated EPDF of an AP and the measured PDF of the same AP are compared. The measured AP EPDF is represented with the solid thin grey line in the middle, while for the estimated EPDF a dashed black line has been adopted. The leftmost PDF is the result of the application of Eq. (6.7), showing the estimation of $\hat{p}_{D_{MS}}$, while the rightmost one is the PDF of the delays measured through the software $p_{D'_{PC}}$.

The figure shows the good agreement between the estimated and the measured EPDFs. If one further calculates the normalized root mean square difference between the PDF estimated with the black-box approach and the directly measured one, a value below 1% is obtained, providing a raw figure of merit of the whole measurement procedure accuracy.

6.5 Summary

In this chapter the characterization of network components has been considered. This topic represents a core issue when dealing with real–time networks and distributed measurement systems in general, and when a wireless channel is chosen in particular.

A method for the characterization of the delays introduced by an access point has been presented, exploiting measurements both through a common personal computer (see Section 6.2 on page 120) and a high–performance digital oscilloscope (see Section 6.3 on page 125). The accuracy of measurement has been regarded throughout the experiments.

From the experimental sessions, the high randomness of the delay introduced by such devices has been highlighted. The delay is typically in the order of some hundreds of microseconds, and its behavior does not follow the statistical description of any known distribution (at least to the best of the author knowledge). Its behavior presents a long tail and, for longer packet interdeparture times, a relevant secondary mode (see Fig. 6.6

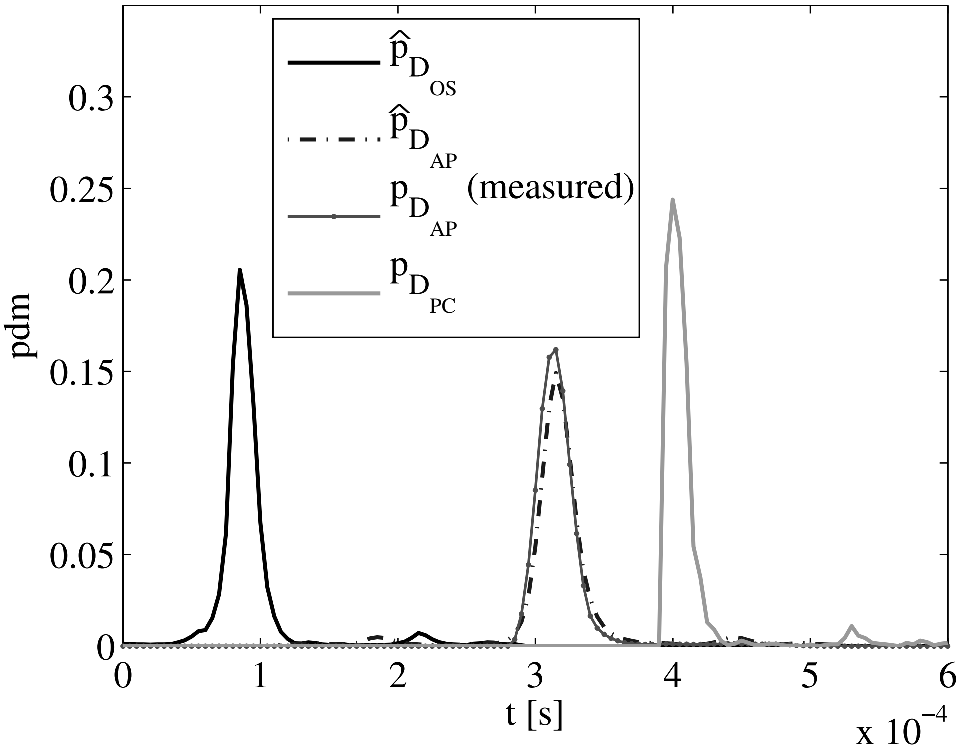


Figure 6.15 – The result of the estimation, and a comparison between the measured and estimated AP behaviour.

on page 122). This however seems to be due to the communication boards of the personal computer, since the delay measured with the oscilloscope does not present this secondary mode (see Fig. 6.12 on page 126).

An original method for the accurate estimation of the delay introduced by an AP is proposed. It exploits a measurement by substitution approach and deconvolution. It is based on a characterization of the measurement system realized through the personal computer by means of a "reference" AP, whose delay EPDF is known. Any other AP under test could be therefore measured through simple traffic flow analysis, avoiding "invasive" measurements on the printed circuit board of the AP itself, provided these can be done in the specific case. The outcomes obtained through this method are encouraging, as shown in Fig. 6.15, and confirmed also the raw estimation of the delays introduced by the measurement system made at the end of Section 6.3 on page 125.

Chapter 7 Rate adaptation strategies

In the framework of the creation of a wireless extension of a wired Real–Time Ethernet (RTE) various performance indexes can be defined, as already explained in Section 1 on page 6. One of the most relevant is the *service time*, which is defined as the time necessary to successfully complete a data transmission from the wired segment to the wireless one.

Considering the nature of the wireless channel, which cannot provide an absolutely reliable communication medium, and given the behavior of the mainly adopted wireless communication standards, which often adopts probabilistic methods to access the channel, it becomes very difficult to provide analytic expressions able to effectively describe the service time metric. In this scenario, a further source of complication is in automatic rate adaptation algorithms, which exploit the availability of different modulation schemes to increase the transmission robustness. The effect is a change in the transmission time of a packet on the basis of both the number of transmission attempts performed and on the "historical performances" of the channel.

In this chapter, mainly based on the work [60, 69], experimental measurements on a real life hybrid wireless/wired network are performed with the goal of characterizing the service time provided by an IEEE 802.11 based wireless segment, in the presence of interference. Rate adaptation effects have been considered, and the measurements have been conducted both enabling and disabling those algorithms. Moreover, original techniques for rate selection are proposed and described, and their performances are compared to those provided by a common choice, namely the ARF algorithm. To this aim, a comprehensive set of simulations is presented, showing the effectiveness of the proposed methods.

7.1 A relevant performance index: the *service time*

Chapter 6 on page 115 has been dedicated to methods for the characterization of network components used to deploy a wireless industrial communication network. In particular, the delay introduced by an Access Point (AP) has been measured and characterized, both through measurements via a personal computer, and through accurate measurements via an high performance digital oscilloscope. The AP was used to realize a hybrid network, in which the AP was used to realize a wireless extension of a wired Real–Time Ethernet (RTE). In this framework, various performance indexes can be defined, as already explained in Section 1 on page 6. One of the most relevant is the *service time*, which is defined as the time necessary to successfully complete a data transmission from the wired segment to the wireless one. Precisely, it is measured as the time that elapses between the delivery of the data packet by the traffic generator to the wired board, and its correct reception by the wireless board, including the time requested by the ACK transmission back to the AP, the time spent for any packet retransmission and that wasted in backoff periods.

In fact, the main wireless standards adopt random backoffs as a statistical method for access the channel, and as a solution to solve collisions between interfering nodes. Indeed, such a stochastic algorithm introduces random delays in order to decrease the probability that two stations decide to transmit at the same time, blocking the transmission of the one loosing the contention (*i.e.* choosing the longer backoff period). Probabilistic contention access algorithms, in addition, give definitely no guarantees about the packet delivery, leaving therefore a small fraction of packets not sent at all.

Since both the underlying physical channel and the physical layer of the chosen communication standard have to be considered as given, and also given that they are not reliable, it is therefore critical to quantify, both theoretically and experimentally, the service time in the case of industrial communication system wireless extensions. Nonetheless it appears pretty difficult to give a closed form solution for a mathematical model of the index. Anyway, measurements in the field could provide interesting figures of merit about the behavior of the service time, and also provide the bounds on the expected performances of such communication systems.

Indeed, industrial communication systems require real-time constraints, *i.e.* the capabilities of the system to react within the specified deadline. Moreover, determinism can be required, serving high priority and aperiodic task in within a specific time threshold, *e.g. alarms*. Clearly, a wireless system, as described above, is not able to definitely guarantee neither a (hard) real-time behavior nor the ability to serve alarms. However, if a soft real-time system would be sufficient, then a probability for the system to respect real-time capabilities could be given, provided accurate experimental measurements and theoretical analyses are performed.

To this purpose it is possible to extend the measurement campaign described through Chapter 6 on page 115 in order to allow an estimation of the service time in the presence of interference. In order to obtain reproducible and effective results, all the aspects of the measurement systems should be defined accurately, and the interference should be controllable to some extent. To this aim, a specific experimental testbed has been realized in order to characterize the performances of such an index in the presence both of a wireless extension of a RTE wired network, and of interference. In particular, the measurement set up of Fig. 6.4a on page 117 can be modified accordingly to the requirements detailed above. In particular, a system to reproduce an interfering communication has been realized. An RF Signal Generator, model Agilent E4432B ESG, has been used as a source of wide-band AWGN, whose bandwidth has been set to 20 MHz so that a full IEEE 802.11g channel could be covered. The AWGN signal has been centered on the same transmission channel of the communication system, namely at 2.437 GHz (*i.e.* channel number 6). A directional log-periodic antenna has been used to selectively disturb the receiver radio. Such a new experimental set-up is described by the picture shown in Fig. 7.1.

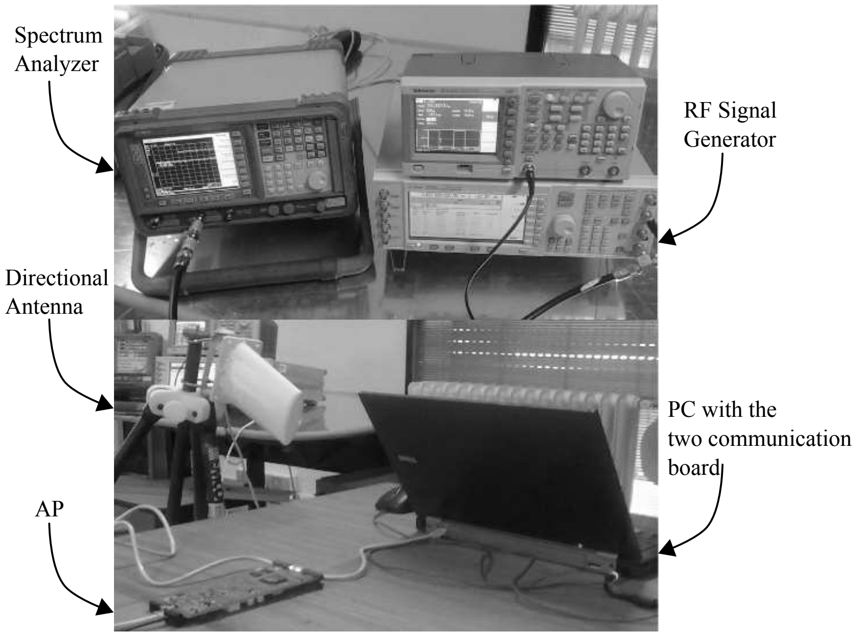
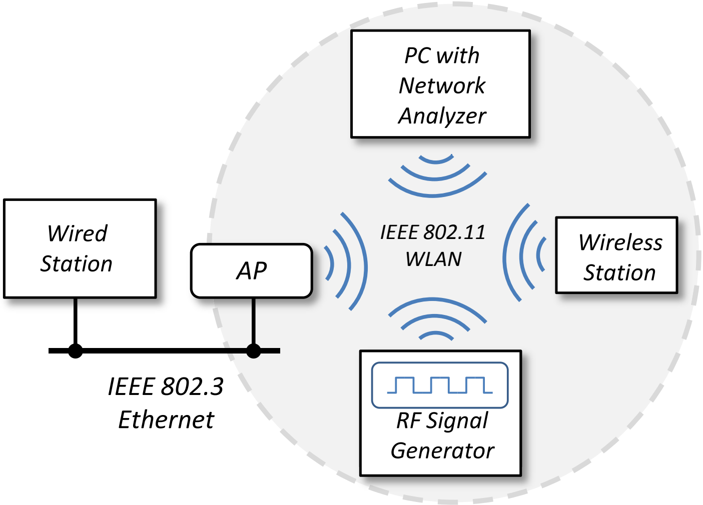


Figure 7.1 – Picture of the experimental set-up

Such a signal acts, with respect to the transmissions performed by the wireless stations, as an increase in the noise floor level, thus suddenly decreasing the SNR. The successful delivery of a frame is, at first, strongly related to this ratio. Indeed, as pointed out in Section 3.4 on page 63 in the case of the IEEE 802.11g standard the probability of a correct transmission of a packet shows the existence of a threshold value which represents a trade-off between two regions in which a frame is either delivered correctly or not. That



!htb

Figure 7.2 – Configuration of the hybrid network used for ARF evaluation

threshold is strongly dependent on the chosen transmission rate. In this light, a relevant influence of the SNR has to be expected on the behavior of the service time as well. Indeed, for SNR values in the range of the threshold of a given rate, the number of retransmissions will likely increase. The sending station will have to execute more backoff procedures, and each time retransmit the packet, leading to longer service times.

7.1.1 Service time in the presence of Interference

The analytic expression of the service time may be derived from Eq. (6.1) and Eq. (6.2) on page 127, and in particular is:

$$T_{ST} = D_{PC} + T_{rtx} + T_{backoff} + SIFS + t_{ack} \tag{7.1}$$

where T_{ST} is the service time, while D_{PC} has the same meaning as in Eq. (6.2), and in particular it represent the time interval between the instant in which the packet is generated and sent on the network and the instant in which the packet is correctly received on the wireless interface. Accounting for the presence of interference that could corrupt some packets, the terms T_{rtx} and $T_{backoff}$ have been considered, which are the time needed for, respectively, the retransmission of a frame and the backoff procedures. Both these times can be theoretically evaluated, since the standard states exactly which is the algorithm in case of retransmissions and the related times [24, 54]. Finally, since acknowledged transmissions are considered, after the correct reception of a frame the receiver has to wait for a SIFS period and then transmit the ACK. These time periods are accounted in the last two terms in Eq. (7.1), and the corresponding values are found in Table 6.1 on page 120. Therefore, in the case of a correct transmission at the first attempt, the service time will coincide by definition with D_{PC} a part from the final ACK transmission, which however has a fixed duration.

The service time T_{ST} was again measured for the transmission of minimum size Ethernet frames (corresponding to IEEE 802.11 frames with a 46 Bytes payload) and the sending period of the software application was set to 15 ms, since we were interested on the effects deriving only by the transmission rate for frame size typical of industrial applications. For each measurement session, 10000 samples of the service time were collected, so as to have a statistically significant set of data.

In a first session, the SNR threshold value has been determined, with the transmission rate fixed at 54 Mb/s, *i.e.* for that interferer power causing serious packet losses in the communication, nearly blocking the communication itself. Clearly, due to the configuration of the measurement system which has not been deployed in an shielded and anechoic environment, and for the characteristics of the wireless medium, a rigorous measurement of the SNR was not possible. Thus, the power of the signal generator has been progressively raised and, contemporaneously, it has been looked for a relevant increase of frame retransmissions, which is a clear indication the threshold SNR is approaching.

This approach served as a basis to identify the SNR threshold which highly impairs the 54 Mb/s transmission rate. It is therefore expected that lowering the transmission rate will increase the delivery probability for packets, hence decreasing the service time. Indeed, with the noise power set to the previously defined threshold value, the measurement of the service time has been carried out in several experimental sessions for different transmission rates foreseen by the IEEE 802.11 standard.

In order to achieve a correct interpretation of the results, they are given below in two different ways: by means of a table providing minimum, mean and maximum values and a plot reporting the Cumulative Distribution Function (CDF). Table 7.3 on the following page shows the measured values of the service time for different transmission rates relevant to the Siemens AP. Moreover, as a further way of comparison, the table shows the theoretical values for the service time, as derived for example from [24], as well as the values measured in the absence of the noise source. The notation "Min (noise)" refers to the measured minimum service time in the presence of noise, whereas, the notation "Min (ideal)" reports the same measurement in the absence of noise. Analogous notations are used for Mean and Maximum values. Moreover, maximum values are actually referred to the 90th percentiles of the measurements.

	Rate [Mb/s]				
	12	**24**	**36**	**48**	**54**
Theoretical	138	118	110	110	106
Min (noise)	371	340	328	324	329
Min (ideal)	364	338	327	325	325
Mean (noise)	407	372	360	1129	1928
Mean (ideal)	407	374	361	362	360
Max (noise)	433	397	385	2151	2610
Max (ideal)	426	391	379	383	384

Figure 7.3 – Measured service time values for the Siemens industrial AP. All values are expressed in µs.

Several considerations can be made with regard to Table 7.3, and they are detailed below:

✓ the difference between theoretical and minimum measured service time values (either in the presence of noise or not) is well explained by the delays introduced by both the APs and the communication boards on the PC. Indeed, as roughly derived in Section 6.3 on page 125, and confirmed through the deconvolution method, the cumulative delay of the two boards is 100 µs, whereas the minimum delay introduced by the Siemens AP is 140 µs, as evaluated with the oscilloscope. Actually, the sum of these (minimum) delays is very close to the aforementioned difference between the minimum and theoretical values.

✓ looking at the mean and maximum values the beneficial effect of reducing the transmission rate can be clearly observed. In particular, at both 48 and 54 Mb/s transmission rates, the service time assumes values that are dramatically greater than those measured in absence of interference

✓ again looking at the two rightmost columns, the values present a relevant randomness. This is caused by the retransmissions that occur as an effect of the corruption of frames. In practice, at high rates, the behavior of the protocol (which implies retransmissions and backoff procedures) has a predominant influence on the service time.

✓ for reduced rates, this effect disappears, leaving the internal behavior of the components (communication boards and access point) as the only source of delay and/or randomness

This is even more evident in Fig. 7.4 which reports the CDF of the service time in the presence of noise. Please, notice again that the curves are relevant to the Siemens AP. As can be seen, the slopes of both the curves at 48 and 54 Mb/s are very different from those of the other transmission rates meaning, in this case, that the service time assumes high values with non negligible probability.

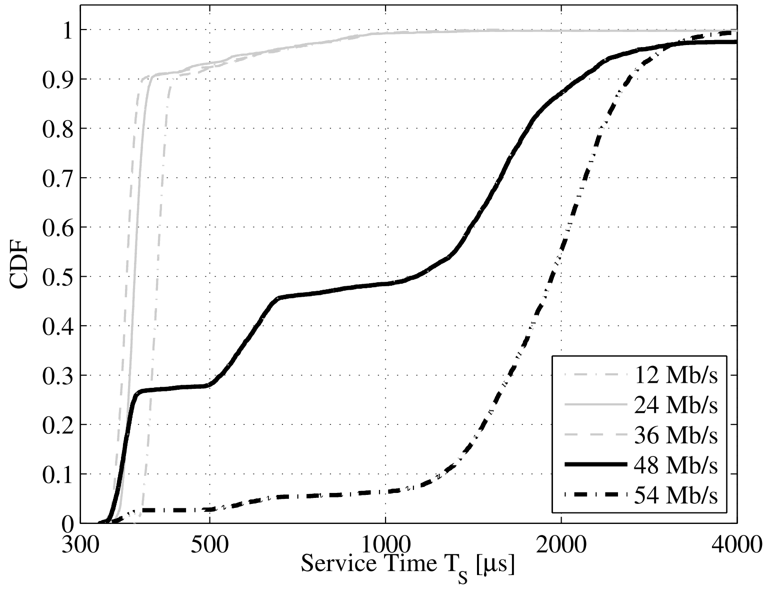


Figure 7.4 – CDFs of the service time (Siemens AP)

7.1.2 Service time with the Rate Adaptation algorithm enabled

The results described in the previous paragraph have been obtained with the AP transmitting at a fixed speed. This fixed a specific modulation and avoid the selection of a more suitable rate. Indeed, as described in the standard, IEEE 802.11 devices may adapt their transmission rates in order to ensure the successful delivery of frames even in environments characterized by relevant bit error rates. Unfortunately, the standard does not specify any rate adaptation technique, leaving the choice to the manufacturers. Thus, the behavior of devices in noisy environments can difficulty be predicted, since the algorithms according to which they vary the transmission rate are not known. This may be, clearly, a further source of difference between theoretical/simulative analysis and practical results. In order to get some helpful insights in this direction, some further measurements on both the 3Com and the Siemens APs have been realized. In particular, with regard to the same configuration shown in Fig. 7.2 on page 138, two different experimental sessions have been carried out. In the first one, no sources of noise were present, whereas in the second one the noise level was set to the previously determined threshold value. The service time was measured in both the sessions for the two APs. The transmission rate of the APs was set to “Automatic”, meaning that the choice of the rate was left to the RA mechanism implemented by the AP firmware. The obtained numerical results are presented in Table 7.5, where again the maximum values are referred to the 90-th percentiles. Moreover, in Fig. 7.6 on the next page the CDFs obtained from the outcomes of measurements are provided

AP-model	Siemens	3Com
Min (noise)	325	400
Min (ideal)	325	394
Mean (noise)	367	606
Mean (ideal)	360	421
Max (noise)	1166	1544
Max (ideal)	382	433

Figure 7.5 – Service time values using the rate adaptation mechanism. All values are expressed in µs.

Even in this case, some interesting observations can be detailed analyzing the obtained data:

- ✓ the values measured in the absence of noise (indicated as “Ideal”), for the Siemens AP in Table 7.5, are practically equal to the corresponding values measured at the rate of 54 Mb/s in Table 7.3 on page 140. For the sake of completeness, the 3Com AP has not been considered in this latter table, nonetheless the same correspondence occurred in this case too. This happened since the AP did not sense any noise and hence set the rate to the maximum possible value.

- ✓ the maximum values of the service time shown in Table 7.5 are significantly lower than those in Table 7.3 on page 140, and a rough comparison indicates that this value is more than halved. This confirms the beneficial effect of the adoption of a RA mechanism. Once again, however, it may be noticed that the presence of noise introduces a considerable level of randomness which becomes predominant on that due to the internal behavior of the APs.

- ✓ considering the CDFs shown in Fig. 7.6 on the facing page, it may be observed that the Siemens device behaves better than the 3Com AP in that it guarantees the service time maintains values close to the minimum with a probability almost up to 70%.

✓ comparing the CDFs shown in both Fig. 7.4 on page 141 and Fig. 7.6, it may be noticed that the use of a fixed transmission rate lower than 48 Mb/s always ensures to obtain shorter service time values with reduced randomness. In other words, using a fixed (relatively) low transmission rate reveals to be much more effective than the adoption of the RA mechanism.

This latter result can be explained, at least partially, considering that the techniques employed for rate adaptation are designed for typical telecommunication scenarios that are characterized by the continuous transmission of packets with considerable payloads (saturated networks). In these cases, usually, the more relevant performance issue to comply with is represented by the maximization of the throughput. This aspect leads to choose high transmission rates, even if some frame retransmission will necessarily occur. Conversely, in industrial applications very short payload packets are used. Hence, the transmission times depend only slightly on the transmission rate. Thus, the effect of the retransmissions becomes predominant on the service time whatever being the transmission rate. As a consequence, it is much more convenient to set the rate to the value that ensures the lowest number of retransmissions.

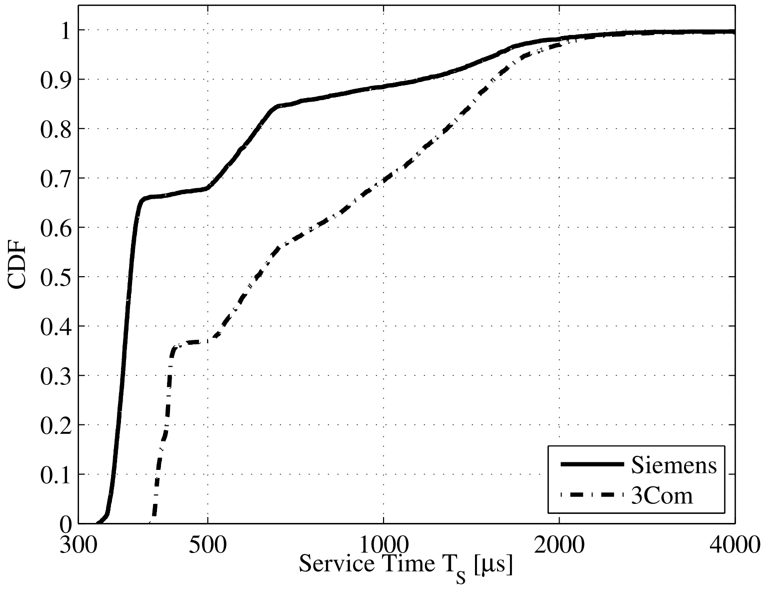


Figure 7.6 – CDFs of the service time using the rate adaptation mechanism

7.2 A common workbench: models for the Industrial Wireless Channel

In the light of the results described in the previous part of the chapter, it will be of interest to present a simple yet effective analysis of the behavior of the channel in an wireless industrial communication environment. Such a channel is generally, as already mentioned, negatively influenced by different phenomena that contribute to make it considerably error prone. Examples come from path loss, shadowing, multipath propagation, thermal noise [76] and interference [44, 23]. In industrial environments, these phenomena are further exacerbated by some peculiar aspects like the harsh environmental conditions, the movement of people and machineries, the possible presence of fixed obstacles to the signal propagation, etc.

The overall effect is basically represented by the degradation of the received signal that may result in uncorrect/missed reception of messages by the destination station(s). As an interesting result in this context, several studies derived from practical experiments [77], [10], [66] agree on the assumption that an industrial wireless channel may be effectively described by a two-state model. In practice, the temporal behavior of the channel is characterized by the alternation of "good" states, where the SNR is high (and, consequently, the probability of successful transmission is close to one) and "bad" states where, conversely, the SNR is low.

This indeed reflects effectively the behavior of typical communication devices adopted in the environment under consideration. As a matter of fact, focusing on IEEE 802.11, the probability of successful frame transmission has already been evaluated, for the traditional industrial traffic, in the previous Section 3.4 on page 63. A graph for the probability of a successful transmission in the case of a frame with payload of 46 Bytes, transmitted correctly at the first attempt, is shown in the following Fig. 7.7 on the facing page.

The results obtained clearly show that, for each transmission rate, the behavior of the success probability presents a very narrow threshold SNR range (which spans 1 or 2 dB of SNR) which discriminates between the values zero and one of this probability. In practice, starting from a state in which transmissions are successful with high probability, an even slight decrease of the SNR will immediately take this probability close to zero unless the transmission rate is reduced accordingly. Therefore it is likely to map the two-state channel model with the behavior of the modulations foreseen by the standard IEEE 802.11, since the occurrence of a bad state coincides with a sudden reduction of the SNR.

Therefore, in the situation described, a station using a RA technique will have to pro-

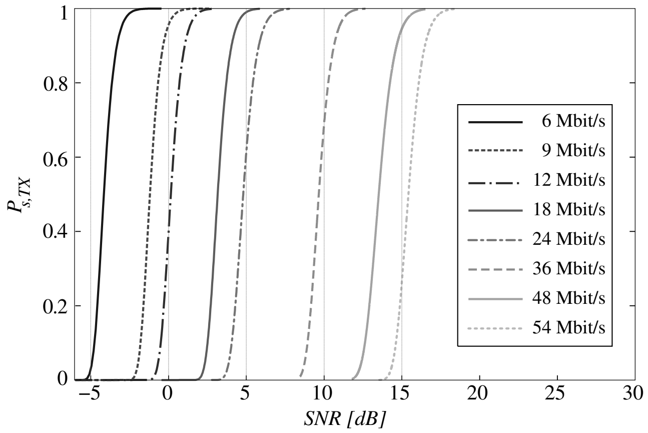


Figure 7.7 – Probability of a successful transmission at the first attempt, for all data rates *vs* SNR. Payload size of 46 bytes.

gressively reduce its transmission rate in order to ensure the probability of successful transmission returns close to one.

Based on the above considerations, different channel models have been considered in this work, based on their characteristics and ability to highlight specific performance issues of a RA technique.

static channel it is characterized by a constant value of the SNR. Although this model is difficult to encounter in practice, it allows to effectively analyze the performance of the RA techniques in steady state conditions.

dynamic channel it emulates the occurrence of an error burst. In this case, it is supposed that the SNR rapidly switches from a high value to a low one in order to force the state of the channel from good to bad. This model allows to evaluate the capability of the RA techniques to effectively adapt the transmission rate to a sudden variation of the SNR.

In the following the different channel models will be considered for the measurement and simulation sessions.

7.2.1 Experimental Set-Up and Simulation Model

With regards to the experimental sessions, a hybrid (wired/wireless) network has been used in order to investigate the behavior of the RA techniques. The structure of the network highly reflects that already described in Section 6.1 on page 117, Section 7.1 on page 135 [60]. Fig. 7.8 on the next page describes such structure.

As already described in Section 7.1 on page 135, two stations are connected to different segments based, respectively, on Ethernet and on IEEE 802.11g interconnected by an AP. As commonly assumed throughout the present work, the network of Fig. 7.8 has been practically implemented locating both the stations on the same PC, running a Linux-based operating system. The already described purposely developed traffic generator has been adopted to cyclically send packets from one station to the other with a specific period. The interference (a wide-band AWGN signal) has been generated following the aforementioned models through a RF signal generator. A further PC, equipped with a network traffic analyzer software was used to collect additional data necessary to complete the performance evaluation.

The analysis carried out in Chapter 6 on page 115 allowed to estimate all the latencies involved in the experimental set-up, ensuring in this way the execution of a very accurate set of measurements. Although it was possible, without distinction, to investigate the behavior of the RA technique for both the wireless devices employed in the experimental set-up (*i.e.* the AP and the communication board of the wireless station) in the following, without loss of generality, results refer exclusively to the AP performances, since for this type of device a configuration tool was available that allowed to handle it in a quite comfortable way. Thus, the software application was configured to send frames from the wired station to the wireless one. Table 7.1 on the next page summarizes all the components used in the experimental set-up.

The wireless segment was implemented as a pure IEEE 802.11g network, where the

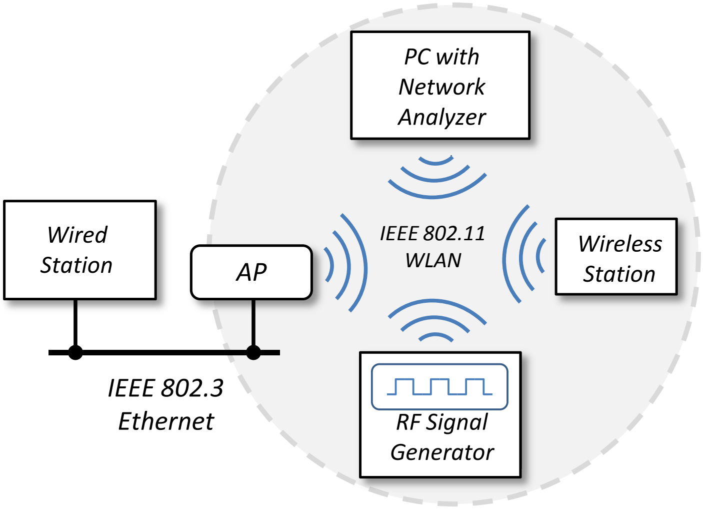


Figure 7.8 – Configuration of the hybrid network used for ARF evaluation

Table 7.1 – Components of the experimental set-up

Description	Component
Access Point #1	3Com Office Connect
Access Point #2	Siemens SCALANCE W784-1
Ethernet board (wired station)	Intel 82567LM Ethernet
IEEE 802.11 g board (wireless station)	Intel WiFi Link 5100 AGN
Network Traffic Analyzer	WIRESHARK 1.4.6
RF Signal Generator	Agilent E4432B ESG

stations adopted the DCF to access the physical medium. The most relevant parameters of the wireless segment are summarized in Table 7.2.

Table 7.2 – IEEE 802.11g parameters

Description	Value
Transmission rates	6, 12, 24, 36, 48, 54 Mbps
Distributed Inter-Frame Space (DIFS)	28 µs
Short Inter-Frame Space (SIFS)	10 µs
Slot Time	9 µs
Max number of retransmissions	7
Payload size (minimum size Eth. frame)	46 Bytes

The simulator presented in Chapter 4 on page 67, characterized and employed throughout this work has been adopted as a mean of performances evaluation for the the hybrid network of Fig. 7.8, in order to test the proposed RA techniques.

7.3 The Automatic Rate Fallback Technique

Automatic Rate Fallback (ARF) was firstly proposed in [39] as a rate adaptation algorithm to improve the performance of WaveLAN-II devices, i.e. those compliant with the first version of the IEEE 802.11 standard, supporting, as basic transmission rates, 1 and 2 Mbps, but also able to reach 10 Mbps. In order to transmit at the highest allowed transmission rate in each operational condition, the devices were equipped with a transmission rate selection algorithm.

More in detail, ARF specifies that, given the set of supported transmission rates $R = \{r_1, r_2, \dots, r_Q\}$, a station decreases its transmission rate from r_i to rate r_{i-1} after K consecutive failed (unacknowledged) transmissions while it increases the transmission rate from rate r_i to rate r_{i+1} after N consecutive successful transmissions. However, if after the transmission rate has just been increased the first transmission attempt (*probing* transmission) fails, then the station will immediately switch back to the lower rate without performing the remaining $K-1$ attempts at the higher rate. In the original version of the ARF technique, K and N were 2 and 10 respectively.

As a further feature, the ARF algorithm proposed in [39] included a timer function that a station starts when it decreases the transmission rate and whose expiration allows the station to switch up to an higher rate even if it has not performed the requested N consecutive successful transmissions.

Finally, it is worth remarking, as pointed out in [41], that ARF is not well optimized neither for very fast nor very slow changing channel conditions. Indeed in the first case, if the channel conditions change too fast, ARF is not able to adapt, since it takes at least K or N transmission attempts to change rate. Conversely in the second case, even if the conditions do not change at all, ARF tries to use a higher rate after every N successful transmissions, likely (and regularly) incurring in a transmission error.

7.3.1 Model of a Station Implementing the ARF Technique

A station implementing the ARF technique has been modeled by means of the discrete time homogeneous Markov chain shown in Fig. 7.9 on page 150. It is assumed that the station always have a packet to transmit and the slot time duration of the chain to be the mean duration of a single packet transmission (including possible interframe times and backoffs). The choice of a discrete time Markov chain since, according to ARF, the transmission rate is decided before each packet transmission. Clearly, the packet transmission time (and hence the slot time of the chain) is not fixed since, in general, it depends on both the transmission rate and the possible backoff delays. As a consequence, the slot time has been selected as an average packet transmission time. Even though this assumption is clearly an approximation, it is commonly used in rate adaptation techniques performance evaluation [46]. As can be seen from the state diagram, before each transmission, a station can be in one of the Q sets of states (each one corresponding to a specific transmission rate)

$$R_i = \left\{ r_i^0, r_i^{1S}, r_i^{2S}, \dots, r_i^{(N-1)S}, r_i^{1F}, r_i^{2F}, \dots, r_i^{(K-1)F} \right\}$$

where, r_i^{jS} means that the station has already carried out j consecutive successful transmissions at rate r_i, whereas r_i^{jF} has the same meaning for failed transmissions. The term p_i in Fig. 7.9 on page 150 represents the packet error probability at rate r_i, as derived in

Section 3.4 on page 63 as a function of the SNR.

Since there are Q possible rates, the probabilities p_i can be arranged in a structure called *packet error probability vector* $V_p = [p_Q \quad p_{Q-1} \quad \dots \quad p_1]$ which is a function of the SNR.

If, before starting a transmission, a station is is in state r_i^{jS}, $(j \leqslant N-2)$, then that transmission will be performed at rate r_i and the previous one has been the last of j consecutive successful transmissions. If this transmission is successful, then the station will switch to state $r_i^{(j+1)S}$. Conversely, if the transmission fails then the station will switch to state r_i^{1F}. The same for states r_i^{jF}, $(j \leqslant K-2)$. After each sequence of N consecutive successful transmissions the station switches from state $r_i^{(N-1)S}$ to state $r_{i+1}^{(K-1)F}$ characterized by rate $r_{i+1} > r_i$. Analogously, after each sequence of K consecutive failed transmissions the station switches from state $r_i^{(K-1)F}$ to state r_{i-1}^0 characterized by rate $r_{i-1} < r_i$. Moreover, after the station has just switched from rate r_i to rate r_{i+1} then it is in state $r_{i+1}^{(K-1)F}$ and if the transmission fails then it switches back to state r_i^0. Summarizing, the state transition probabilities of the Markov chain result:

$$\begin{cases} P[r_i^{jS} \rightarrow r_i^{(j+1)S}] = 1 - p_i \quad i = 1, \dots, Q-1, \quad j = 0, \dots, N-2 \\ P[r_i^{jS} \rightarrow r_i^{1F}] = p_i \quad i = 1, \dots, Q-1, \quad j = 0, \dots, N-1 \\ P[r_i^{jF} \rightarrow r_i^{(j+1)F}] = p_i \quad i = 2, \dots, Q, \quad j = 0, \dots, K-2 \\ P[r_i^{jF} \rightarrow r_i^{1S}] = 1 - p_i \quad i = 2, \dots, Q, \quad j = 0, \dots, K-1 \\ P[r_i^{(N-1)S} \rightarrow r_{i+1}^{(K-1)F}] = 1 - p_i \quad i = 1, \dots, Q-1 \\ P[r_i^{(K-1)F} \rightarrow r_{i-1}^{0}] = p_i \quad i = 2, \dots, Q \\ P[r_1^0 \rightarrow r_1^0] = p_1 \\ P[r_Q^0 \rightarrow r_Q^0] = 1 - p_Q \end{cases}$$

7.4 Performance of available RA Techniques

This section provides the results of the test sessions executed on the theoretical model of the ARF technique as well as on the RA techniques adopted by the two APs alternatively employed in the experimental testbed of Fig. 7.8 on page 146.

The purpose is to verify the performances that would be obtained adopting ARF as a rate adaptation technique in an industrial automation networks perspective. Another goal is to characterize the RA algorithm implemented in the two APs. AP manufacturers very often, and in this case apply, do not provide adequate technical specifications of their

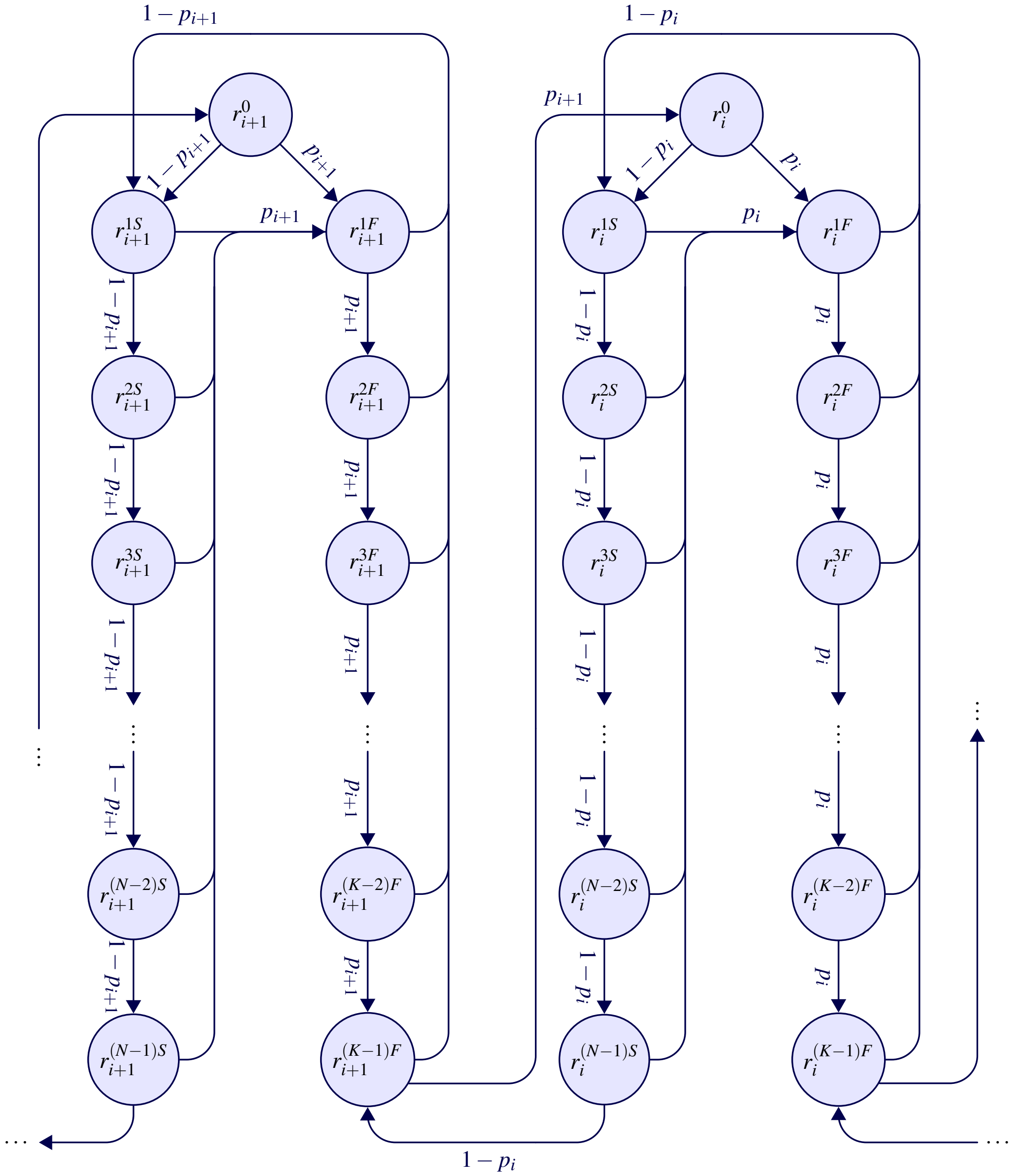


Figure 7.9 – ARF algorithm model

product, in the sense that various communication related parameters result undeclared. In this specific case, the implemented RA algorithm is not specified. An accurate experimental session, carefully designed to the scope, could give interesting insights on the choices the manufacturers applied in designing their products.

In every test session a typical example of industrial traffic is assumed, in which the wired station sent 10,000 frames to the wireless one with a period of 15 ms.

7.4.1 Tests with a static channel model

This preliminary analysis has been therefore carried out in order to investigate the behavior of the transmission rate as selected by the ARF algorithm. In this direction, the model just described is considered and a constant SNR value set, ensuring a high successful transmission probability at 36 Mbps. The resulting packet error probabilities vector was

$$V_p = [0.9999 \quad 0.4542 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

meaning that the single packet error probabilities were, respectively, 0.9999 at 54 Mbps, 0.4542 at 48 Mbps, and zero for all rates $\leqslant$ 36 Mbps.

Plain ARF performances

As already said, the model for ARF has been implemented in the OMNeT++-based simulator presented in this work. It has been therefore in the situation described, using the proposed (classic) values $N = 10$, $K = 2$. The results obtained are shown in Fig. 7.10 on the following page where the y-axis shows the transmission rate (as selected by the ARF technique) and the x-axis reports the progressive transmission number. To this regard, it is worth observing that the progressive transmission number does not necessarily coincide with the number of the packet delivered by the wired station, since the same packet may undergo some retransmission attempts. Moreover, dots represent successful transmissions that took place at the rate indicated on the correspondent y-axis, whereas x-marks represent failed transmissions. In order to ensure an adequate readability, the figure is relevant to a limited number of transmissions that, however, are sufficient to explain the behavior.

Fig. 7.10 on the next page shows, as expected, that the station implementing the ARF technique switches effectively from different transmission rates. For example, it may be noticed that the rate regularly increases from 36 Mbps to 48 Mbps after 10 consecutive successful transmission attempts, whereas it decreases at the immediately lower value either

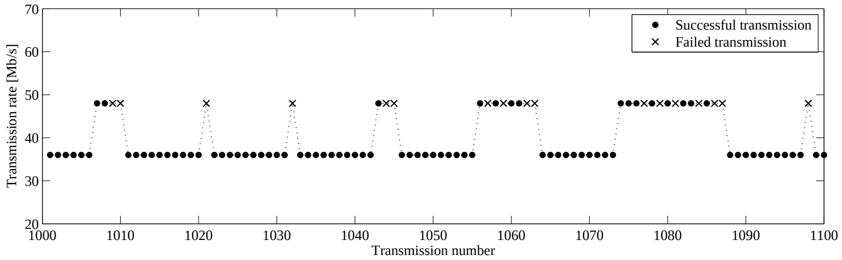


Figure 7.10 – Transmission rate behavior for the ARF technique

after 2 consecutive failed transmissions or after one failure if the station has just increased its transmission rate. On the other hand, it has to be observed that the number of failed transmission attempts (and hence of the consequent retransmissions) is not negligible at all. This result, that at a first glance could seem surprising (since the SNR is maintained constant) is well explained considering that the increase of the transmission rate, that takes place after *N* successful attempts, has the effect of reducing the successful probability transmission, as described by Fig. 3.22 on page 66, with the consequent increase of the retransmission probability.

Analysis of the RA algorithm implemented on the analyzed APs

Subsequently, having in mind a clear picture of the behavior of the ARF technique in the designed case, experiments on the real APs have been made. The aforementioned environmental conditions have been emulated through the RF signal generator, manually adjusting the noise power level to a value that ensured a high successful transmission probability at 36 Mbps. The condition has been verified through a real-time analysis of packet flow, looking at the rate used for the transmission of each packet.

The obtained behavior of the transmission rate for both the APs used in the practical experiments is provided, respectively, in Fig. 7.11 on the next page for the 3Com AP and in Fig. 7.12 on the facing page for the Siemens one.

An immediate comparison with Fig. 7.10 leads to some interesting considerations:

1. both the APs behavior deviates consistently from that of a station implementing a pure ARF technique.

2. *for the 3Com AP*, it may be noticed in Fig. 7.11 on the facing page that the RA technique employed always tries to transmit at high rates, since a single successful transmission at 36 Mbps is sufficient to increase the rate to 48 Mbps

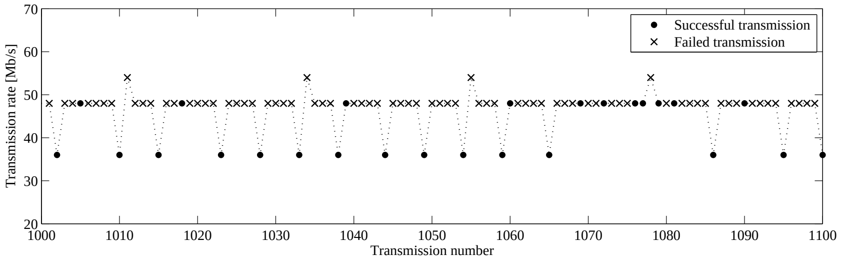


Figure 7.11 – Transmission rate behavior for the 3com AP

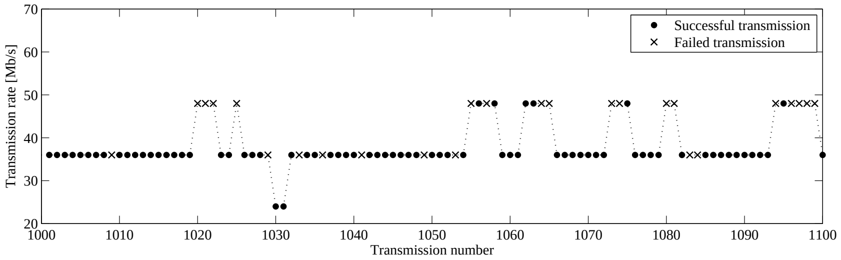


Figure 7.12 – Transmission rate behavior for the Siemens AP

3. *for the Siemens AP*, the behavior looks more stable, since it seems the adopted RA technique aims at selecting an almost constant rate (36 Mbps in this case) where the probability of successful transmission is high. However, the rationale behind the chosen algorithm is absolutely not clear.

About the second point, several other transmission sequences have been analyzed, often finding them similar to the one shown in the figure. However, the rule stated evidently does not apply for the transition to 54 Mbps, as can be noticed in the figure, where the rate in one case increases from 48 Mbps to 54 Mbps after two successful transmissions whereas, in some other cases, it switches directly from 36 Mbps to 54 Mbps after just one successful transmission. Moreover, the reduction of the rate takes place after four consecutive failed transmissions, suggesting in this way the RA technique adopted the value $K = 4$. Conversely, in some sporadic cases unexpected (and not explainable) behaviors were detected.

For the case of the Siemens AP, as it may be noticed, for the interval of transmissions shown in Fig. 7.12, the rate increase takes place several times after different numbers of consecutive successful transmissions. The same behavior can be observed for rate reduction. Moreover in the considered transmission interval there are two (unexplainable) rate reduction cases that are consequent to successful transmissions.

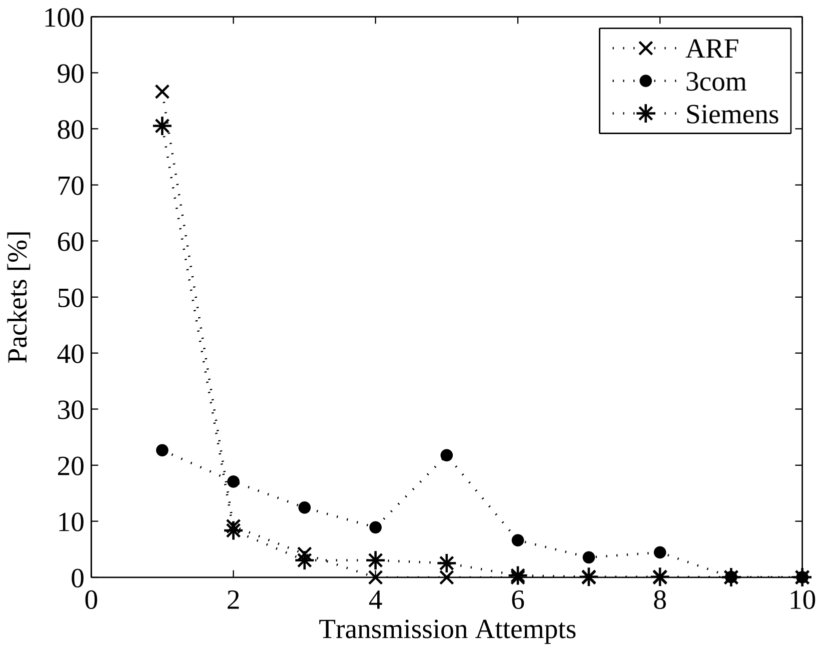


Figure 7.13 – PDFs of the transmission attempts with a static channel

Number of transmission attempts for a correct delivery

In a second step of the performance analysis, the focus has been specifically directed on the number of transmission attempts necessary for the correct delivery of a packet. This represents a meaningful index of the randomness of the service time, as described in Section 7.1 on page 135, since the more the transmission attempts, the higher the randomness, due to the random backoffs waited before each retransmission.

Thus, analyzing the outcomes of the previous measurements, the EPDF of the total number of transmission attempts that occurred in a session test has been obtained. This measure is given in Fig. 7.13.

As can be seen, both a station implementing ARF and the Siemens AP perform quite similarly, in that they transmit most of the packets at the first attempt. However, as already pointed out, the number of retransmissions is not negligible at all (more than 10% of the packets are not transmitted at the first attempt by the ARF station, whereas this percentage raises up to almost 20% for the Siemens AP). On the other hand, the 3Com AP presents a much more distributed PDF revealing a high number of transmission attempts, clearly due to the strategy of transmitting at high rates it adopts.

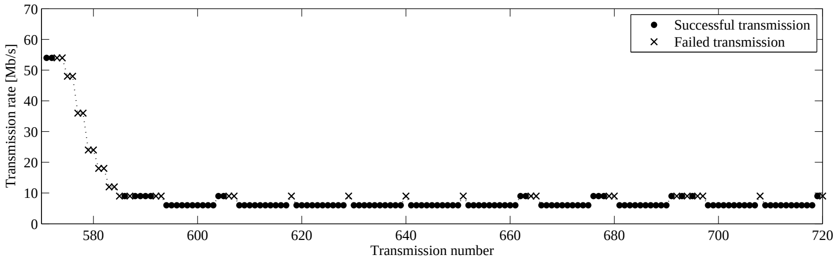


Figure 7.14 – Transmission rate behavior for the ARF technique with dynamic channel

7.4.2 Tests with a dynamic channel model

In this session of tests, the reaction of the RA techniques to a noise burst that causes a sudden reduction of the SNR is investigated. In particular, the analysis was focused on the transitory period in which the techniques adjust the transmission rate to cope with the SNR reduction. In order to emulate a worst case situation, the noise power level has been chosen in such a way that the transmission rate, to ensure a good successful probability transmission, had to be lowered from 54 Mbps to 6 Mbps.

In the theoretical model of the ARF technique, and observing at Fig. 7.7 on page 145 such a transition has been achieved changing the SNR from 17 dB to -1 dB. The resulting packet error probabilities vectors were respectively

$$V_{p,good} = [0.0649 \quad 0.0010 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$V_{p,bad} = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0.9999 \quad 0.4454 \quad 0.0001]$$

The behavior of the transmission rate is shown in Fig. 7.14, where the occurrence of the noise burst takes place at transmission #575.

It may be noticed that the transmission rate is progressively reduced until the value of 6 Mbps is reached. This transition implies several retransmissions, since the ARF technique requires that all the rates are “crossed”, with the consequent increase of the randomness in packet delivery. Particularly, in the worst case, a packet whose transmission starts exactly at the beginning of the noise burst, could require up to 14 transmission attempts before reaching the final rate. Considering that the default maximum number of retransmissions is 7, the packet might be definitely discarded.

In the experimental testbed the SNR variation has been emulated manually. Unfortunately, although several attempts have been made, it was not possible to reach a SNR value capable of forcing the RA techniques to steadily adopt the rate of 6 Mbps. In particular, an

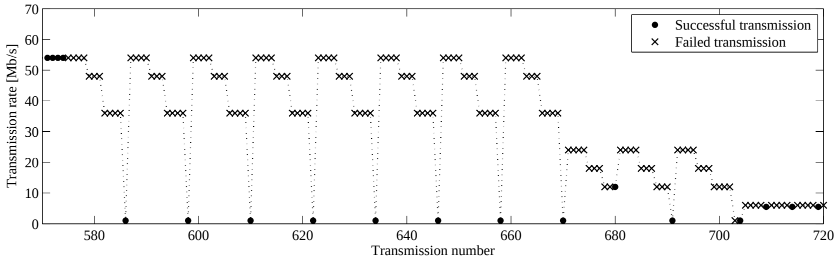


Figure 7.15 – Transmission rate behavior for the Siemens AP with dynamic channel

unexpected behavior of the 3Com AP is worth of observation: the lowest rate selected by the RA technique was 12 Mbps; further decreasing the SNR led the AP to (unexpectedly) enter the IEEE 802.11b mode, where it transmitted at 1 Mbps. However, this transmission mode was definitely excluded through the configuration settings of the AP.

The behavior of the Siemens AP is shown in Fig. 7.15.

It may be noticed that during the transient, the transmission rate was reduced every four consecutive failures, whereas a single success was sufficient to increase it to a very high value. Strangely, in the first part of the transient the RA technique switched only among three high rates (54, 48 and 36 Mbps) suddenly moving, after some failures, to 1 Mbps (IEEE 802.11b mode) where a single successful transmission was sufficient to restore the highest rate. Subsequently (roughly from transmission #670, three lower rates were alternatively selected (24, 18 and 12 Mbps) and the same behavior observed, that is a successful transmission that took place at 1 Mbps was sufficient to increase the rate. Then, once the new stable condition was eventually reached, the AP actually selected the rate of 6 Mbps but it was not able to maintain it in that it returned to the IEEE 802.11b mode, transmitting at 5.5 Mbps. For the sake of clarity, this behavior may be also due to some uncertainty of the measurement system that did not allow to set a SNR value capable of forcing the RA technique to steadily select the rate of 6 Mbps. Nonetheless, even if a finely tune of the noise power level has been tried, the overall result was that the rate selected was either 5.5 Mbps or 12 Mbps even for very limited variations of the noise amplitude.

It is worth to observe that both the APs temporarily entered a sort of non operational state in reaction to the noise burst, and this time period lasted for up to 5 s. In this time interval, the APs were not able to accept any service request from the upper layer, with the consequent loss of several packets whose transmission requests were issued by the test application we used. Since the internal structure of the APs, and hence the RA techniques they use, are not exactly known, it results difficult to comprehensively explain the described behaviors. However, based on the practical experience derived from the experimental tests, the following hypothesis can be formulated: the RA techniques of the chosen APs "prefer" selecting the IEEE 802.11b mode in reaction to a sudden decrease of the SNR, since the modulation techniques used in this mode are much more robust than those of IEEE 802.11g. This ensures the best probability of successful transmission.

Concerning the non operational period, it seems that, at first, the APs "interpret" the noise burst as an anomalous event that forces them in an idle state from which they leave only after a time-out is expired. Then, some sort of SNR estimation seems to be performed and on this basis the devices select the more appropriate transmission rate as specified by the implemented RA technique.

7.5 Proposed RA Techniques for Industrial Applications

The most relevant drawback of the RA techniques analyzed in the previous section, when employed for industrial applications, is represented by the potentially high number of retransmissions a packet may undergo. This has a negative effect on the performance of IEEE 802.11 networks employed for industrial communication systems since it increases both the mean value and the randomness of the packet service time. Consequently, there is the need for alternative RA techniques designed in the direction of reducing the number of retransmissions that, however, at the same time do not lead to an increase of the service time. Thus, to this regard, the conservative choice to always set the transmission rate to the lowest value often does not represent a good solution, as pointed out in [60, 69]. Unfortunately, the techniques proposed in the literature are not conceived for industrial communications, thus resulting inadequate for these applications. For example, the Adaptive Automatic Rate Fallback (AARF) outlined in [41] is a variant of ARF that, in case of failures, progressively increases the number of consecutive successful transmissions necessary to move from a rate to the upper one. In practice, if a failure occurs at rate r_{i+1} after N successful transmissions carried out at rate r_i, then the number of successes necessary to switch again to rate r_{i+1} will be increased to $2N$ and so forth. AARF, intuitively, ensures good performance for slow varying channels but, on the other hand, it results not adequate for fast varying channels. Moreover, in these cases, the restoring of the rate to high values might require considerable time.

In this perspective, two original and effective RA techniques are here proposed and the analysis of their performance obtained through numerical simulations is given. Basically, the techniques represent two variants of ARF specifically designed in order to meet the aforementioned requirements.

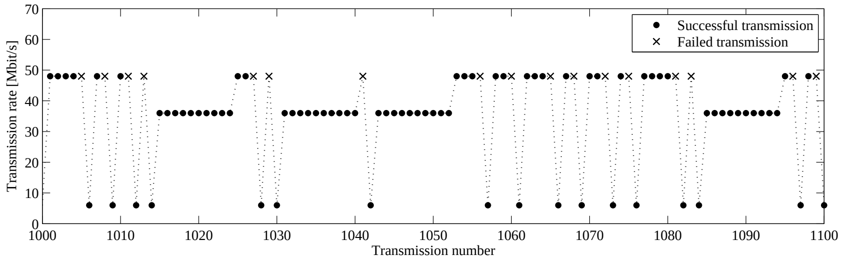


Figure 7.16 – Transmission rate behavior for the SARF technique with static channel

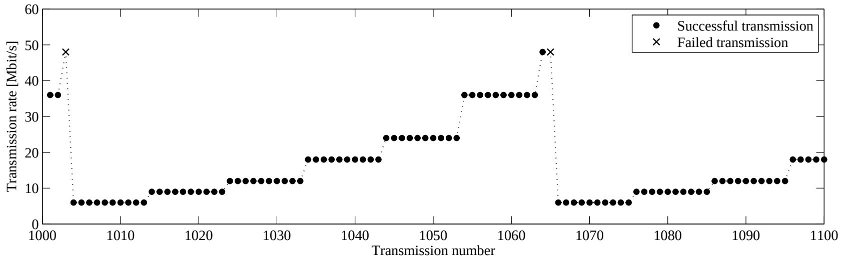


Figure 7.17 – Transmission rate behavior for the FARF technique with static channel

Static retransmission rate ARF (SARF) The SARF algorithm behaves like a plain ARF but, in order to limit the randomness in packet delivery, it specifies that each retransmission takes place at the lowest rate (6 Mbps), ensuring in this way the highest successful probability. Moreover, the positive result of a retransmission is not considered as an event that resets the number of consecutive failures K. In other words, after two consecutive failed transmissions at rate r_i interleaved by a successful retransmission at 6 Mbps, the SARF technique selects the rate r_{i-1} for the transmission of the next packet.

Fast rate reduction ARF (FARF) this is a modified version of ARF that, at the occurrence of a failure always selects 6 Mbps as the next rate. Moreover, this technique specifies that the new rate is entered just after one failure ($K = 1$).

7.5.1 Analysis of SARF and FARF for a Static Channel

The behavior of the transmission rate as selected by the two techniques is shown, respectively, in Fig. 7.16 for SARF and in Fig. 7.17 for FARF. It may be noticed that both algorithms have the characteristic of dramatically reducing the number of transmission attempts, since every retransmission of a packet is carried out adopting a very robust BPSK modulation,*i.e.* 6 Mbps.

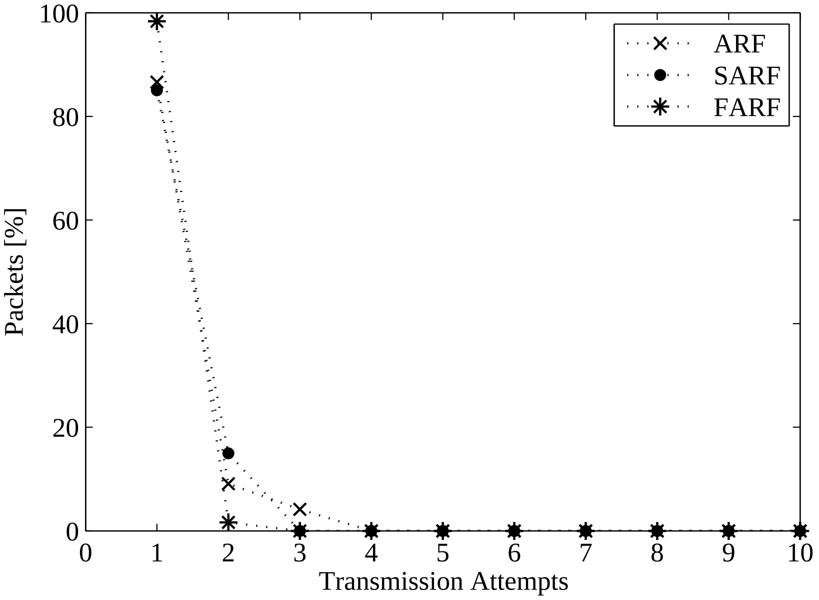


Figure 7.18 – PDFs of the transmission attempts for ARF, SARF and FARF, with a static channel

This feature is even more evident looking at Fig. 7.18, which reports the PDFs of the transmission attempts.

As can be seen, both SARF and FARF never required more than one retransmission, performing in this way, better than ARF. Also, good results are provided by the two techniques in terms of service time, whose statistics is given in Table 7.3.

Table 7.3 – Service time for ARF, SARF and FARF with a static channel

RA Technique	Mean	Std. Dev.
ARF	498 µs	156 µs
SARF	500 µs	132 µs
FARF	488 µs	56 µs

It may be observed that, while the mean values are very close for the three techniques, both SARF and FARF have a lower standard deviation, accounting for the reduced randomness. Particularly, the service time obtained with the FARF technique exhibits a very limited variability, since this technique ensures that the transmissions are successful at the first attempt with very high probability. The residual randomness is due to the internal behavior of the components as well as to the different transmission rates adopted.

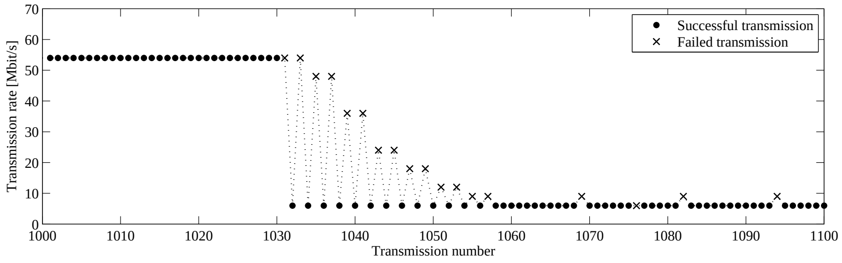


Figure 7.19 – Transmission rate behavior for the SARF technique with dynamic channel

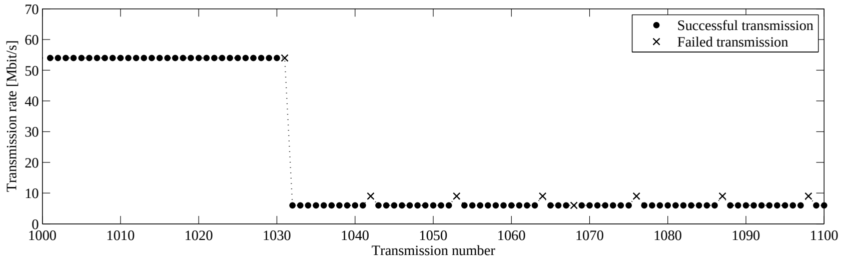


Figure 7.20 – Transmission rate behavior for the FARF technique with dynamic channel

7.5.2 Analysis of SARF and FARF for a Dynamic Channel

The behavior of the transmission rate as selected by the two techniques at the occurrence of a noise burst is shown, respectively, in Fig. 7.19 for SARF and in Fig. 7.20 for FARF. In both the figures, the error burst occurs exactly after 10 seconds of simulation, which correspond to transmission #1030.

Clearly, both the techniques perform better than ARF since, provided that packets are successfully transmitted at 6 Mbps, the techniques ensure that a packet never requires more than one retransmission in order to be correctly delivered. Moreover, in this specific situation FARF performs considerably better than SARF in terms of retransmissions. Indeed, as can be noticed in both Fig. 7.19 and Fig. 7.20, at the occurrence of the burst, FARF immediately reaches the rate of 6 Mbps, whereas SARF needs to pass through all the other rates, considerably increasing the number of transmission attempts.

On the other hand, if the noise burst were of lower amplitude (*i.e.* if the SNR allowed to safely set a rate greater than 6 Mbps), then SARF would reach the new state faster than FARF (even if still requiring a greater number of transmission attempts). This is shown in both Fig. 7.21 on the next page and Fig. 7.22 on the facing page, where it is supposed that the noise burst reduces the SNR to a value for which the rate of 24 Mbps ensures a high transmission successful probability.

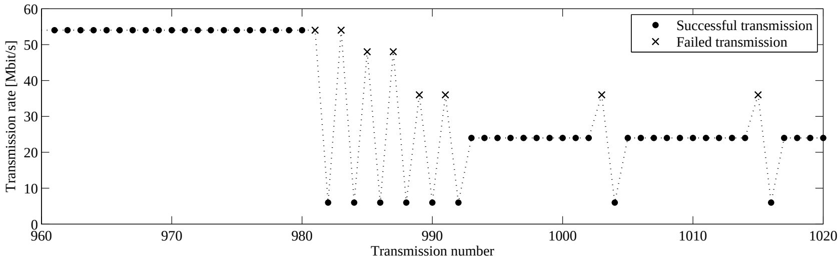


Figure 7.21 – Transmission bate behavior for the SARF Technique with Reduced Noise Burst

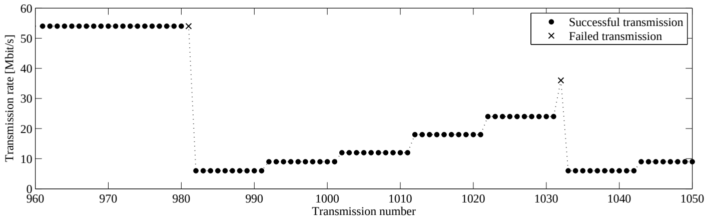


Figure 7.22 – Transmission bate behavior for the FARF Technique with Reduced Noise Burst

7.5.3 Analysis of SARF and FARF for Different Traffic Profiles

In order to investigate the effectiveness of the proposed RA techniques in a wider range of fields, their behavior for applications that imply the transmission of packets with high payload size has also been investigated. This scenario fits particularly well for the industrial multimedia traffic, where each (high payload size) packet could be used to carry, for example, either the complete set of data relevant to a single image, or a fragment of a more complex video file. Similarly, packets of this type could be employed at the higher levels of distributed automation systems, where relevant amounts of data needs to be exchanged with more relaxed time constraints [72], [58].

The first simulations have been executed for a static channel, using the parameters of the previous session tests, but with a packet payload size of 500 bytes. The analysis of the results showed that the EPDF of the transmission attempts has the same behavior of Fig. 7.18 on page 159, where it can be seen that both SARF and FARF perform better than ARF. Conversely, the statistics of the service time shows better (lower) values for ARF with respect to the other two RA techniques, as can be seen in Table 7.4 on the following page.

This result, actually, is not surprising since for these payloads the difference between the transmission times at, respectively, 54 Mbps and 6 Mbps is very relevant (638 µs) and, moreover, the backoff procedure usually has a minor impact on the service time because,

Table 7.4 – Service time for 500-byte payload packets, static channel

RA Technique	Mean	Std. Dev.
ARF	618 µs	113 µs
SARF	670 µs	271 µs
FARF	820 µs	211 µs

in this case, the packet transmission times are considerably greater than the delays introduced by the backoff intervals. In such a condition, the strategy adopted by both SARF and FARF of always retransmitting at 6 Mbps reveals not effective for a static channel since it leads to greater values of both the mean and the standard deviation of the service time than those obtained by ARF, even if these two RA techniques require a lower number of transmission attempts.

Conversely, when a dynamic channel is considered, the scenario changes again. This context has been investigated through the use of a channel in which the noise bursts are repeated with a period of 500 ms. Clearly this behavior does not reflect exactly any specific physical channel. Nevertheless, it represents an effective benchmark for the RA techniques in the context of industrial applications. The statistics of the service time obtained by the simulation is shown in Table 7.5.

Table 7.5 – Service time for 500-byte payload packets, dynamic channel with repetition of the noise bursts

RA Technique	Mean	Std. Dev.
ARF	1164 µs	1216 µs
SARF	1070 µs	369 µs
FARF	1012 µs	316 µs

Clearly, both SARF and FARF, in this case, perform better than ARF since the higher number of transmission attempts employed by this latter technique to reach the final transmission rate when a noise burst occurs becomes prominent in determining the service time. Such a relevant number of retransmissions, particularly, has a strong negative effect on the standard deviation that results very high accounting for a considerable randomness.

7.6 Summary

The effect of interference on a real–time wireless communication system based on IEEE 802.11 standard has been analyzed in this Chapter. In particular the behavior of the physical layer has been considered in Section 7.1 on page 135, with particular reference to the provided rate adaptation mechanisms.

A relevant delays is introduced by the algorithms found in some commercial devices, as well as in the common literature solution called ARF. The latter has been in–depth described and simulations of its behavior have been compared with the outcomes of experimental measurements of the implemented rate adaptation techniques on two commercial APs, (see Section 7.3 on page 147 and 7.4).

Two original and effective new algorithms for rate adaptation in the case of real–time constraints on the wireless network has been proposed in Section 7.5 on page 157. Simulations through the software presented in this thesis have confirmed their very good performances in terms of delays and retransmissions. Indeed, these techniques are conceived to reduce considerably the number of retransmission attempts, thus decreasing sensibly the time wasted in backoff periods, which also increases the randomness of the network behavior. Simulations have been performed both in the case of typical industrial traffic (small payloads) and emulating the behavior of the forthcoming real–time multimedia (industrial) traffic, characterized by higher payloads.

Bibliography

[1] L. Angrisani, M. Bertocco, D. Fortin, and A. Sona, "Experimental study of coexistence issues between ieee 802.11b and ieee 802.15.4 wireless networks," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 8, pp. 1514 –1523, aug. 2008.

[2] N. Benvenuto and G. Cherubini, *Algorithms for Communications Systems and Their Applications.* John Wiley & Sons, Ltd, 2002.

[3] F. Benzi, G. Buja, and M. Felser, "Communication Architectures for Electrical Drives," *IEEE Trans. on Industrial Informatics*, vol. 1, no. 1, pp. 47–53, February 2005.

[4] M. Bertocco, G. Gamba, and A. Sona, "Is CSMA/CA really efficient against interference in a wireless control system? An experimental answer," in *Proc. of IEEE ETFA*, Hamburg, Germany, September 2008.

[5] M. Bertocco, G. Gamba, A. Sona, and F. Tramarin, "Investigating wireless networks coexistence issues through an interference aware simulator," in *Proc. IEEE Int. Conf. Emerging Technologies and Factory Automation ETFA 2008*, 2008, pp. 1153–1156.

[6] M. Bertocco, A. Sona, and F. Tramarin, "Design of experiments for the assessment of coexistence between wireless networks," in *Proc. IEEE Instrumentation and Measurement Technology Conf. (I2MTC)*, 2010, pp. 928–932.

[7] M. Bertocco and F. Tramarin, "A cross-layer simulator for industrial wireless communication systems," in *Proc. IEEE International Workshop on Measurement&Networking (M&N2011)*, 2011.

[8] P. Bhagwat, P. Bhattacharya, A. Krishna, and S. K. Tripathi, "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs," *Wireless Networks*, vol. 3, no. 1, pp. 91–102, Mar. 1997.

[9] C. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, “The Impact of Temperature on Outdoor Industrial Sensornet Applications,” *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 3, pp. 451–459, august 2010.

[10] D. Brevi, D. Mazzocchi, R. Scopigno, A. Bonivento, R. Calcagno, and F. Rusina’, “A methodology for the analysis of 802.11a links in industrial environments,” in *Proc. of WFCS*, Torino, Italy, 2006, pp. 165–174.

[11] G. Cena, I. B. Cibrario, A. Valenzano, and C. Zunino, “Evaluation of response times in industrial WLANs,” *IEEE Trans. on Industrial Informatics*, vol. 1, no. 3, pp. 202–214, May 2007.

[12] G. Cena, A. Valenzano, and S. Vitturi, “Hybrid wired/wireless networks for real-time industrial communications,” *IEEE Industrial Electronic Magazine*, vol. 2, no. 1, pp. 8–20, March 2008.

[13] C. M. De Dominicis, P. Ferrari, A. Flammini, E. Sisinni, M. Bertocco, G. Giorgi, C. Narduzzi, and F. Tramarin, “Investigating wirelesshart coexistence issues through a specifically designed simulator,” in *Proc. IEEE Instrumentation and Measurement Technology Conf. I2MTC ’09*, 2009, pp. 1085–1090.

[14] J. D. Decotignie, “Ethernet–Based Real–time and Industrial Communications,” *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1102–1117, June 2005.

[15] ——, “The Many Faces of Industrial Ethernet [Past and Present],” *IEEE Industrial Electronic Magazine*, vol. 3, no. 1, pp. 8–19, March 2009.

[16] *Pythagor Simulator*, Department of Information and Communication Systems Engineering – University of Aegean. [Online]. Available: http://www.icsd.aegean.gr/telecom/Pythagor

[17] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia, “On the Fidelity of IEEE 802.11 Commercial Cards,” in *Proceedings of the 1st IEEE WICON*, Budapest, Hungary, july 2005.

[18] J. P. Espanha, P. Naghshtabrizi, and Y. Xu, “A Survey of Recent Results in Networked Control Systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, January 2007.

[19] M. Felser, “Quality of Profibus Installations,” in *Proc. of the 6th IEEE WFCS*, Torino, Italy, June 2006, pp. 113–118.

[20] L. L. Feng, J. Moyne, and D. M. Tilbury, “Performance evaluation of control networks,” *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 66–83, February 2001.

[21] P. Frenger, P. Orten, and T. Ottosson, “Convolutional codes with optimum distance spectrum,” *Communications Letters, IEEE*, vol. 3, no. 11, pp. 317 –319, nov 1999.

[22] G. Gamba, L. Seno, and S. Vitturi, "Performance Indicators for Wireless Industrial Communication Networks," in *Proc. of the 8th IEEE WFCS*, Nancy, France, May 2010, pp. 3–12.

[23] G. Gamba, F. Tramarin, and A. Willig, "Retransmission strategies for cyclic polling over wireless channels in the presence of interference," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 405–415, 2010.

[24] G. Gamba and S. Vitturi, "Statistical Evaluation of the Service Time for IEEE 802.11g Networks in Industrial Applications," in *Proc. of the 8th IEEE WFCS*, Nancy, France, May 2010, pp. 67–70.

[25] G. Gamba, F. Tramarin, and A. Willig, "Retransmission strategies for cyclic polling over wireless channels in the presence of Interference," in *Proc. 14th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2009*, Mallorca, Spain, Sep. 2009.

[26] A. Goldsmith, *Wireless Communications.* Cambridge University Press, 2005.

[27] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Aspects," *IEEE Trans. on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, October 2009.

[28] I. Howitt, "WLAN and WPAN coexistence in UL band," *Vehicular Technology, IEEE Transactions on*, vol. 50, no. 4, pp. 1114–1124, Jul 2001.

[29] *IEEE 802.3 standard: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*, IEEE Std., October 2000.

[30] *IEEE Recommended Practice for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements Part 15.2: Coexistence of Wireless Personal Area Networks With Other Wireless Devices Operating in Unlicensed Frequency Bands*, IEEE Std., 2003.

[31] *IEEE Standard 802.15.4: Wireless Medium Access Control (MAC)and Physical Layer (PHY) Specifications for Low–rate Personal Area Networks (LR–WPANs)*, IEEE Std., September 2003.

[32] *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std., June 2007.

[33] *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, Institute of Electrical and Electronic Engineers Std., July 2008.

[34] *IEC61784 International Standard: Digital data communications for measurement and control. Part 1: Profile sets for continuous and discrete manufacturing relative to fieldbus use in industrial control systems. Part 2: Additional profiles for ISO/IEC8802–3 based communication networks in real–time applications*, International Electrotechnical Commission Std., November 2007.

[35] *ISA-100.11a Wireless systems for industrial automation: Process control and related applications*, ISA Std., 2009.

[36] J. Jasperneite, J. Imtiaz, M. Schumacher, and K. Weber, "A Proposal for a Generic Real-Time Ethernet System," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 2, pp. 75 −85, may 2009.

[37] J. Jasperneite and P. Neumann, "Switched Ethernet for Factory Communication," in *Emerging Technologies and Factory Automation, 2001. Proceedings. 2001 8th IEEE International Conference on*, October 2001, pp. 205 −212 vol.1.

[38] H. Kaghazchi, H. Li, and M. Ulrich, "Influence of Token Rotation Time in Multi Master PROFIBUS Networks," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, Dresden, Germany, may 2008, pp. 189–197.

[39] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, pp. 118–133, August 1997.

[40] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in OMNeT++: The MiXiM vision," in *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops*, ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 71:1–71:8. [Online]. Available: http://portal.acm.org/citation.cfm?id=1416222.1416302

[41] M. Lacage, M. H. Manshaei, and T. Turletti, "Ieee 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '04. New York, NY, USA: ACM, 2004, pp. 126–134.

[42] K. C. Lee, S. Lee, and M. H. Lee, "Worst Case Communication Delay of Real-Time Industrial Switched Ethernet with Multiple Levels," *IEEE Trans. on Ind. Electr.*, vol. 53, no. 5, pp. 1669–1676, October 2006.

[43] F. Li, M. Li, R. Lu, H. Wu, C. Mark, and K. Robert, "Measuring queue capacities of ieee 802.11 wireless access points," in *Proc. of the Fourth International Conference on Broadband Communications, Networks and Systems. BROADNETS 2007.*, Sept. 2007, pp. 846–853.

[44] L. Lo Bello and E. Toscano, “Coexistence Issues of Multiple Co-Located IEEE 802.15.4/Zigbee Networks Running on Adjacent Radio Channels in Industrial Environments,” *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 2, pp. 157 −167, may 2009.

[45] *MATLAB*, MathWorks. [Online]. Available: http://www.mathworks.com

[46] A. Min and K. Shin, “An Optimal Transmission Strategy for IEEE 802.11 Wireless LANs: Stochastic Control Approach,” in *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON ’08. 5th Annual IEEE Communications Society Conference on*, June 2008, pp. 251 −259.

[47] G. Mulligan, “The 6lowpan architecture,” in *EmNets ’07: Proceedings of the 4th workshop on Embedded networked sensors.* New York, NY, USA: ACM, 2007, pp. 78–82.

[48] T. Nandagopal and X. Gao, “Fair scheduling in wireless packet data networks,” in *Handbook of Wireless Networks and Mobile Computing*, I. Stojmenovic, Ed. New York: John Wiley & Sons, 2002, pp. 171–194.

[49] Omnet++. INET Framework. [Online]. Available: http://inet.omnetpp.org/

[50] *OMNeT++ Network Simulation Framework*, OMNeT++ Community. [Online]. Available: http://www.omnetpp.org

[51] *Opnet Modeler*, OPNET Technologies, Inc. [Online]. Available: http://www.opnet.com

[52] J. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[53] M. Pursley and D. Taipale, “Error probabilities for spread-spectrum packet radio with convolutional codes and viterbi decoding,” *Communications, IEEE Transactions on*, vol. 35, no. 1, pp. 1 – 12, jan 1987.

[54] D. Qiao, S. Choi, and K. Shin, “Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs,” *Mobile Computing, IEEE Transactions on*, vol. 1, no. 4, pp. 278 – 292, oct-dec 2002.

[55] T. S. Rappaport, *Wireless Communications – Principles and Practice.* Upper Saddle River, NJ, USA: Prentice Hall, 2002.

[56] T. Sauter, “The continuing evolution of integration in factory automation,” *IEEE Industrial Electronic Magazine*, vol. 1, no. 1, pp. 10–19, Spring 2007.

[57] ——, “The Three Generations of Field-Level Networks–Evolution and Compatibility Issues,” *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 11, pp. 3585 −3595, nov. 2010.

[58] T. Sauter and M. Lobashov, "How to Access Factory Floor Information Using Internet Technologies and Gateways," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 699 –712, nov. 2011.

[59] V. M. Sempere and J. Silvestre, "Multimedia Applications in Industrial Networks: Integration of Image Processing in Profibus," *IEEE Trans. on Ind. Electr.*, vol. 50, no. 3, pp. 440–449, June 2003.

[60] L. Seno, F. Tramarin, and S. Vitturi, "Experimental evaluation of the service time for industrial hybrid (wired/wireless) networks under non-ideal environmental conditions," in *Proc. of IEEE Conf. Emerging Technologies & Factory Automation ETFA 2011*, Toulouse, France, September 5-9 2011.

[61] ——, "Influence of Real Components Behavior on the Performance of Wireless Industrial Communication Systems," in *Proc. of the 20th IEEE ISIE*, Gdansk, Poland, June 2011.

[62] ——, "Performance of industrial communication systems in real application contexts," *IEEE Industrial Electronics Magazine*, 2011, accepted. In press.

[63] J. Silvestre-Blanes, L. Almeida, R. Marau, and P. Pedreiras, "Online QoS Management for Multimedia Real-Time Transmission in Industrial Networks," *IEEE Trans. on Ind. Electr.*, vol. 58, no. 3, pp. 1061–1071, March 2011.

[64] S. Sooyeon, K. Taekyoung, J. Gil-Young, Y. P., and H. Rhy, "An Experimental Study of Hierarchical Intrusion Detection for Wireless Industrial Sensor Networks," *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 4, pp. 744–757, november 2010.

[65] S. T., J. J., and L. B. L., "Towards new hybrid networks for industrial automation," in *Proc. of the 14th International Conference on Emerging Technologies and Factory Automation, ETFA 2009*, Palma de Mallorca, Spain, September 2009.

[66] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme, "The industrial indoor channel: Large–scale and temporal fading at 900, 2400 and 5200 mhz," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, July 2008.

[67] J. P. Thomesse, "Fieldbus Technologies in Industrial Automation," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1073–1101, June 2005.

[68] J.-P. Thomesse, "Fieldbus Technology in Industrial Automation," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1073–1101, Jun. 2005.

[69] F. Tramarin, L. Seno, S. Vitturi, and M. Bertocco, "On the rate adaptation techniques of ieee 802.11 networks for industrial applications," *IEEE Trans. Ind. Informat.*, 2011, accepted for publication.

[70] A. Viterbi, "Convolutional codes and their performance in communication systems," *Communication Technology, IEEE Transactions on*, vol. 19, no. 5, pp. 751 −772, october 1971.

[71] S. Vitturi, "Stochastic Model of the Profibus DP Cycle Time," *IEE Proceedings - Science, Measurement & Technology*, vol. 151, no. 05, pp. 335–342, September 2004.

[72] V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review," *Industrial Informatics, IEEE Transactions on*, vol. 7, no. 4, pp. 768 −781, nov. 2011.

[73] X. Wang, G. B. Giannakis, and A. G. Marques, "A Unified Approach to QoS-Guaranteed Scheduling for Channel-Adaptive Wireless Networks," *Proceedings of the IEEE*, vol. 95, no. 12, pp. 2410–2431, Dec. 2007.

[74] K. Wessel, M. Swigulski, A. Köpke, and D. Willkomm, "Mixim: the physical layer an architecture overview," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009, pp. 78:1–78:8. [Online]. Available: http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5555

[75] "HART Communication Foundation," http://www.hartcomm.org/.

[76] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, May 2008.

[77] 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," *IEEE Trans. on Ind. Electr.*, vol. 49, no. 6, pp. 1265–1282, December 2002.

[78] A. Willig, K. Matheus, and A. Wolisz, "Wireless technologies in industrial networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1150, June 2005.

[79] A. Willig and A. Wolisz, "Ring stability of the profibus token-passing protocol over error-prone links," *IEEE Trans. on Ind. Electr.*, vol. 48, no. 5, pp. 1025–1033, October 2001.

[80] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communications: A Selection," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, May 2008.

[81] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1130–1151, Jun. 2005.

[82] ZigBee Alliance. [Online]. Available: http://www.zigbee.org

[83] "Industrial communication systems," in *The Industrial Information Technology Handbook*, R. Zurawski, Ed. CRC Press, 2005, pp. 37.1–47.16.