Università
degli Studi
di Padova

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

REAL-TIME NETWORKS AND PROTOCOLS FOR INDUSTRIAL AUTOMATION

Ph.D. candidate
Lucia Seno

Advisor
Prof. Stefano Vitturi

Ph.D. School in
Information Engineering
2011

Contents

Abstract

Nowadays, networks are fundamental elements for developing factory automation systems. In this thesis, we focus industrial communication networks (ICNs) employed at the device level, for the fast data exchange between controller and sensors/actuators, that are required to satisfy tight reliability and real-time requirements. In this context, the scenario is rapidly evolving. Indeed on the one hand, high performance and low costs are required to cope with more and more demanding requirements, while, on the other hand, real-time characteristics are needed in an increasing number of industrial automation applications. In the last years, two new types of ICNs have been introduced for device-level industrial communication. Besides fieldbuses, traditionally employed, both real-time Ethernet (RTE) networks, based on IEEE 802.3 standard, and wireless networks have become available.

In this thesis we focus on Ethernet Powerlink, a popular RTE network. Firstly, we refer to the Ethernet Powerlink specifications and provide a description of the network and, in particular, of the data link layer protocol real-time features. We propose a modification to the protocol that allow the network to better manage real-time acyclic traffic and provide results of a relevant both theoretical and a simulative analysis. Subsequently we investigate possible wireless extensions of Ethernet Powerlink based on the IEEE 802.11 wlan and present results of both a theoretical and a simulative analysis relevant to two specific solutions. Finally we practically realize one of the two solutions and provide results of extensive measurements on the real system. We particularly focus on the comparison between theoretical, simulative and experimental results and on the influence of real components behavior on the overall performance of the system.

Sommario

Le reti di comunicazione sono diventate un elemento fondamentale dei sistemi di automazione industriale. In questa tesi si considerano, in particolare le reti di comunicazione impiegate per la comunicazione di livello dispositivo, per il veloce scambio di dati tra contollori e sensori/attuatori, e alle quali è solitamente richiesto un elevato grado di affidabilità e determinismo. Lo scenario della comunicazione industriale si sta rapidamente evolvendo. Infatti da una parte prestazioni sempre più elevate e costi contenuti devono sempre più conciliarsi con la soddisfazione di aspettative crescenti in termini di real-time, dall'altra sempre più applicazioni di automazione industriale richiedono uno scambio di dati preciso e veloce. Negli ultimi anni, in particolare, due nuovi tipi di reti di comunicazione industriale sono apparse sul mercato e si sono iniziate ad utilizzare nei sistemi di automazione. Oltra alle reti di comunicazione di campo, tradizionalmente utilizzate, reti real-time Ethernet, che si basano sullo standard 802.3, e reti wireless si sono cominciate ad diffondere negli ambienti industriali.

In questa tesi ci focalizziamo su Ethernet Powerlink, una diffusa rete real-time Ethernet. Inizialmente descriviamo la rete e in particolare le caratteristiche di real-time del protocollo di livello data-link da essa definito. Proponiamo, quindi una modifica al protocollo che consentirebbe alla rete di gestire in modo più efficiente il traffico real-time aciclico e presentiamo alcuni risultati, ottenuti sia analiticamente che per mezzo di simulazioni. In seguito, analizziamo possibili estensioni wireless di Ethernet Powerlink mediante la rete 802.11 e presentiamo risultati ottenuti per via teorica e simulativa relativi a due particolari soluzioni. Infine implementiamo praticamente una delle due soluzioni e presentiamo i risultati di una serie di misure effettuate sul sistema reale.

In particolare, mettiamo in evidenza differenze tra l'analisi teorica, simulativa e sperimentale e l'influenza del comportamento di componenti reali sulle prestazioni fornite dal sistema nel suo complesso.

Introduction

Nowadays, industrial communication networks (ICNs) are a key element at all levels of factory automation systems. In particular, ICNs are employed for communication at the lowest level, also known as *device level*, where the fast exchange of data between controller and sensors/actuators takes place. Clearly, the precision of the exchange of control/controlled variables between controller and sensors/actuators located on the factory plant/machine, deeply influences the overall performance of the system. As a consequence, differently from general purpose networks, ICNs employed for communication at the device level are typically required to satisfy tight deterministic[1] as well as real-time requirements. Moreover, these ICNs are often required to provide a high level of reliability.

In this thesis we will specifically refer to device level ICNs. In detail, device level traffic may be of two types: cyclic and acyclic. Cyclic traffic is usually given by control/controlled variables that have to be periodically transfered from the controller to the actuators or from the sensors to the controller (e.g. set point signal samples, process variables values, etc.). Acyclic traffic is usually event-driven and its generation is not predictable (e.g. alarms).

ICNs have evolved considerably over the years. The '80s have seen the development of several industrial communication protocols providing deterministic performance (*Token Bus*, *Token Ring*, etc.) and, adopted by ICNs like, for example, WorldFIP [45], Profibus [16], P-Net [8], Interbus [23], AS-Interface [2], SERCOS [61], LonWorks [6], MVB [7], DeviceNet [20], SDS [11], CAN [35]. All these ICNs, commonly known as *fieldbuses*, have been purposely

[1] With the term *deterministic* we intend that data have to be delivered from any source station to any destination station within a predefined (limited) time interval.

developed for device level industrial communication and have been traditionally employed in this context (several of these networks are still widely deployed in factory environments). Most of fieldbuses have also been encompassed by the IEC 61158 International Standard [22].

However, in some applications (e.g. motion control applications) the performance provided by fieldbuses may result not completely satisfactory due to their relatively low transmission rates (tens of Kb/s) as well as to the MAC protocols they adopt. As a consequence, in the 90's, the performance provided by fieldbuses have begun to be considered too limited when compared to the those provided by some well-known, general purpose networks such as IEEE 802.3 standard Ethernet [21] (Ethernet transmission rate can be up to 10 Gb/s). Moreover, the users begun to require the possibility of transmitting higher quantities of data.

Consequently, an attempt was made in order to upgrade the existing ICNs toward higher transmission rates and performance. This upgrade, however, revealed to be not be feasible in some cases due to limitations in the principles of operation (e.g. CAN), but more importantly the cost for upgrading the existing ICNs was a consistent issue for the vendors. On the other hand, Ethernet was becoming so widespread that the cost of its ICs was lowering and lowering. For these reasons, some vendors begun to develop ICNs based on commercially available Ethernet ICs, possibly introducing different protocols in order to improve determinism, maintaining high transmission rates (up to 100 Mb/s).

The newly developed Ethernet-based ICNs are known as *real-time Ethernet (RTE)* networks and their employment in factory automation systems rapidly increased. Nowadays, several commercial products by different vendors are available on the market (e.g. ProfiNet [27], EtherNet/IP [24], Ethernet Powerlink [43], EtherCAT [42], etc.). Most of these networks have been encompassed in both the IEC 61158 and IEC 61784 International Standards [29].

As a further enhancement, in the last years, besides fieldbuses and RTE networks, a new technology has become available for industrial communication. Indeed, *wireless technologies* have become particularly appealing for a wide range of industrial communication applications, relevant to all levels of factory automation systems.

In general, the availability of low-cost standard wireless networks is offering new opportunities in several application areas, such as, for example, personal

mobility, home networking, and office automation. One of the most popular wireless technologies used today is undoubtedly the IEEE 802.11 [19]. The adoption of such a family of wireless networks, also known as wireless lans (wlans), is suggested by some of their appealing features, such as, in particular, the high transmission rates (up to 54 Mb/s), which provides more than an adequate bandwidth for many applications, and the close resemblance with Ethernet. The last aspect means, that interoperability can be obtained through simple and inexpensive access points which transparently interface wireless stations to wired segments.

However, even though the technology behind wireless communications has evolved quickly in many traditional and emerging application fields, the same is true only up to a certain point for industrial and factory automation systems. Indeed, introducing wlans in industrial applications leads to face a larger number of challenges than those met in home or enterprise applications, the most severe being the fulfillment of tight requirements about reliable and real-time transmission typical of this field. For example, voice-over-IP applications can tolerate transmission latencies up to 150 ms and the corruption of up to 1% of the exchanged frames, thanks to the adoption of adaptive play-out control and error concealment algorithms. By contrast, factory automation systems usually require shorter cycle times, in the typical range between 1 and 10 ms, that may also fall below 1 ms in some particular applications (e.g. motion control applications). In addition, minimizing communication jitters is also of primary importance in many control systems based on cyclic operations. Another important issue is the demand for a frequently not negotiable deterministic behavior: for instance, runtime performance degradation is not a viable option for mission-critical industrial applications. These requirements apply also during device roaming, which leads to the need for a real-time handover.

It is worth pointing out that the majority of the currently defined wlans, as well as the relevant commercially available products, is generally considered unsuitable for implementing distributed control systems and applications, in particular, when real-time is one of the key issues [38]. This is mainly due to three reasons: communication over radio channels is usually very sensitive to electromagnetic interference (largely present in factory environments), that may cause excessively high transmission error rates; even in the case there is no electromagnetic noise, interferences might be generated by the presence of wireless networks nearby, including other wlans that are not under control of

the system administrator; and even in the case the first two could be neglected, the random access scheme wlans rely on, the carrier sense multiple access with collision avoidance (CSMA/CA) technique, prevents them from offering a deterministic behavior.

For the above reasons, there is no guarantee in a wlan that any given frame is eventually delivered to the intended destination(s). In particular, the system designer has to deal explicitly with two severe issues: no upper bound is ensured on transmission latencies and the possibility that a frame is completely lost cannot be neglected at all. Needless to say, this is usually considered unacceptable in most applications. As a direct consequence, in industrial environments, wlans are used at present mainly to enable simple and cost-effective maintenance and diagnostic functions. For instance, conventional access points are sometimes connected to the networks used at the factory shopfloor (in particular, when they are based on RTE networks) in order to enable management devices (laptops and handhelds) to be temporarily connected to the system for reconfiguring the control software or changing the operating parameters.

For the sake of truth, a number of wireless technologies have been developed recently, that are aimed explicitly at connecting controller and sensor/actuator devices over the air (either directly or indirectly). InduraNET [28], for instance, relies on antenna diversity as well as an intelligent coexistence frequency management (CFM), that enables it to operate in the presence of other wireless systems in the 2.4 GHz ISM band. Wireless HART [12] utilizes IEEE 802.15.4 compatible DSSS radios with channel hopping on a packet by packet basis. Time-division multiple access (TDMA) is adopted to coordinate communications in order to avoid collisions. Phoenix Contact's Wireless Interface RAD-ISM-900-XD-BUS [1] is based on frequency-hopping spread spectrum (FHSS) in the 902 to 928 MHz ISM band. Often, the adoption of guaranteed time slots (GTSs) is envisaged in wireless sensor networks (WSNs) used in industrial applications so as to avoid collisions [59]. The above solutions, which are not based on IEEE 802.11 technology, mostly rely on improved physical layers that make them more resistant to environmental noise and envisage some kind of scheduling scheme for improving determinism in network access. Another example are WISA system [62] compliant products. It is worth noticing that the bit rate in these networks is noticeably lower than in wlans.

Summarizing, the advantages of the employment of wireless technologies in industrial communication would be manifold (support for mobility, reduced

deployment and maintenance costs, reduced risk of cable/connectors failures, enhanced flexibility, etc.). However, because of a number of reasons that have been extensively analyzed and debated in the scientific literature, wireless networks cannot be thought of as a complete replacement of wired ICNs in factory environments.

A more likely employment of wireless networks in industrial communication would be in order to implement wireless extensions of already deployed wired communication systems, realizing hybrid (wired/wireless) networks. These networks represent an effective solution to the problem of connecting to an already deployed wired communication system few stations that can not be reached (either easily or reliably) by means of a cable. The conjunction and inter-operation of a wired ICN with a wireless network may be actually a critical issue. Indeed wireless networks usually have transmission rates lower than those of ICNs, resulting in a lower throughput on the wireless segment (and, consequently, on the whole system). Analogously, the higher bit error rates and randomness typical of wireless networks may cause relevant, unpredictable communication delays on the wireless segment. Problems like these ones may be solved, for example, by limiting the number of stations and the traffic on the wireless segment, employing wireless networks with high transmission rate, using Quality of Service (QoS) techniques that allow to prioritize frames, etc.

The main goal of the research activity we carried out was the analysis (development/description, modeling and performance evaluation) of technologies, networks and protocols suitable for industrial automation. The methods we adopted to carry out such activities were mainly concerned with the theoretical and simulative analysis and experimental measurements on real systems. In particular, we focused on a specific RTE network, namely Ethernet Powerlink and investigate the possibility of realizing a wireless extension of the network based on the IEEE 802.11 wlan. We were interested in possible practical implementations and, over all, on the performance provided by the proposed hybrid systems in order to understand whether they would be suitable for device level communication applications.

In order to evaluate the performance provided by an industrial communication technology of any type (e.g. fieldbuses, RTE networks, wireless networks, hybrid networks, etc.) and, in particular, to compare the behaviors of different communication systems, it is very important to define some parameters,

referred as *performance indicators*, that if computed for a particular system allow to understand the actual performance it provides. To this regard, an interesting initiative is, for example, represented by the performance indicators defined by the IEC 61784 International Standard for RTE networks.

We carried out both theoretical and simulative analysis of the proposed wireless extension, particularly referring to some purposely defined performance indicators. We finally implemented one of the proposed extensions and experimentally evaluate its performance, comparing the results with those previously obtained and highlighting/analyzing the observed differences.

In detail, the thesis is structured as follows. In chapter 1, we focus on Ethernet Powerlink. We firstly provide a description of the network, as derived from the Ethernet Powerlink specifications. We particularly concentrate our attention on the handling of both cyclic and acyclic traffic provided and on Ethernet Powerlink reliability and real-time characteristics. Our contribution is the proposal of a change to the Ethernet Powerlink data link layer protocol which would allow an efficient handling of real-time acyclic traffic (e.g. alarms), maintaining full compatibility with the original specifications. We also present results obtained by extensive simulation of the behavior Ethernet Powerlink networks with different configuration, particularly focusing on typical values assumed by some performance indicators.

In chapter 2, we investigate possible extensions of Ethernet Powerlink based IEEE 802.11 wlan. In detail, we propose two types of extensions: one realized at the data link layer, based on a bridge device, and the other one implemented at the application layer, based on the implementation of a gateway device. After a discussion concerning the more convenient practical implementations of the two extensions, we present an evaluation of the performance provided by the hybrid networks. Since the reliability of wireless transmissions represents a very critical aspect, the analysis is carried out taking into account the presence of interference as well as fading on the radio channel. The results, obtained by both theoretical analysis and numerical simulations, allow to get some useful insights on the overall behavior of possible hybrid networks industrial applications.

In chapter 3 of this thesis, we define a set of performance indicators suitable for evaluating the behavior of polling-based wireless industrial communication systems. We compute such performance indicators, both analytically and experimentally for the wireless extension of Ethernet Powerlink at the data link

layer described in chapter 2 and practically implemented by means of commercially available IEEE 802.11 access points. The purpose is to show how the performance indicators may characterize the performance provided by an industrial communication system as well as how they may be positively influenced by a careful selection of some key parameters. Obtained results show relevant differences between theoretical and practical results, probably due to the random, implementation-dependent latencies introduced by employed real components, that can only be detected experimentally. Our conjecture is validated by the results obtained using access point devices from different vendors that cause relevant modification to the behavior of the overall system.

Finally, in chapter 4, starting from the observations in chapter 3, we focus our attention on the analysis of the behavior of a specific real wireless component (a general purpose IEEE 802.11 access point). By means of extensive measurement campaigns we deduce the empirical pdfs of the delay it introduced in communication. Then, we used the obtained results to implement more realistic simulation model of a possible device level industrial communication systems. In detail, we simulate the behavior of the considered system in presence of two different master/slave protocols in order to both understand how the presence of the real access points negatively influences the performance and which techniques allow to limit this influence and preserve the system efficiency.

1 Ethernet Powerlink for industrial communication

1.1 Motivation

Industrial communication solutions based on Ethernet technologies (Real-Time Ethernet, RTE, networks) are popular in factory environments and several competing products have been developed in the last years.

In this chapter, we focus on Ethernet Powerlink (EPL), a RTE network defined by Communication Profile (CP) 1 of Communication Profile Family (CPF) 13 of IEC 61784 International Standard and managed by the Ethernet Powerlink Standardization Group (EPSG). In particular, we present an analysis aimed at evaluating the real-time behavior of EPL for configurations typically employed in industrial communication applications. Starting from the network specifications, we investigate how both cyclic and acyclic traffic are managed and propose a modification of the EPL data link layer protocol (EPL protocol) which allows, in spite of no significant variations of the protocol, to handle real-time acyclic data (e.g. alarms). We also provide results of a performance analysis carried out both theoretically and through numerical simulation. In particular, we analyze the degradation of EPL performance in presence of transmission errors, as it is common in factory environments.

In detail, the chapter is organized as follows. In section 1.2 we provide a short description of EPL and particularly of the EPL protocol. In section 1.3, we describe the alternative technique to effectively manage real-time acyclic data and provide results obtained from numerical simulation of the behavior of a specific configuration implementing the new technique. In section 1.4 we present basic results, concerning both cyclic and acyclic real-time traffic management, that can be used as a first step for a better understanding of the behavior of EPL. Finally, in section 1.5 we draw some conclusions.

1.2 Ethernet Powerlink

Ethernet Powerlink (EPL) is a popular Real-Time Ethernet (RTE) network. The first version of EPL was developed in 2001 by B&R [3]. In 2003, the EPL Standardization Group (EPSG) [4], which is currently managing the EPL network development, published the EPL specifications as an open standard, *Ethernet Powerlink v. 2.0* [43]. This version of EPL has been included in the IEC 61784 International Standard [29] as Communication Profile (CP) 1 of Communication Profile Family (CPF) 13. A newer version of the EPL specifications [44] is also available on the EPSG website.

EPL uses IEEE 802.3 standard Ethernet [21], without any change of its characteristics and extends it in order to avoid collisions and achieve determinism in communication. In particular, the EPL extension of IEEE 802.3 standard Ethernet allows EPL to provide synchronization between stations, cyclic transmission of time-critical data characterized by low and predictable delays and asynchronous (i.e. on request) transmission of less time-critical data. These features make EPL actually suitable for industrial communication, since the network meets the tight timing requirements typical of industrial communication applications.

1.2.1 Network architecture

With reference to the ISO/OSI model [17], the EPL network architecture is shown in Fig. 1.1.

As can be observed, the EPL communication profile is based on the definition of a data link layer protocol (*EPL protocol*) placed on top of physical layer and MAC sublayer natively used by Ethernet. Moreover, an application layer protocol, based on the well-known CANopen standard [35], is placed on top of the EPL data link layer (for this reason EPL is also known as *"CANopen over Ethernet"*).

The use of IEEE 802.3 standard Ethernet at both physical layer and MAC sublayer allows EPL to be implemented on any Ethernet-compliant hardware device as well as it allows the user to employ standard Ethernet infrastructure components and test/measure/diagnostic devices. Analogously, the use of CANopen at the application layer guarantees the compatibility of the EPL network with a large number of already deployed industrial communication systems.

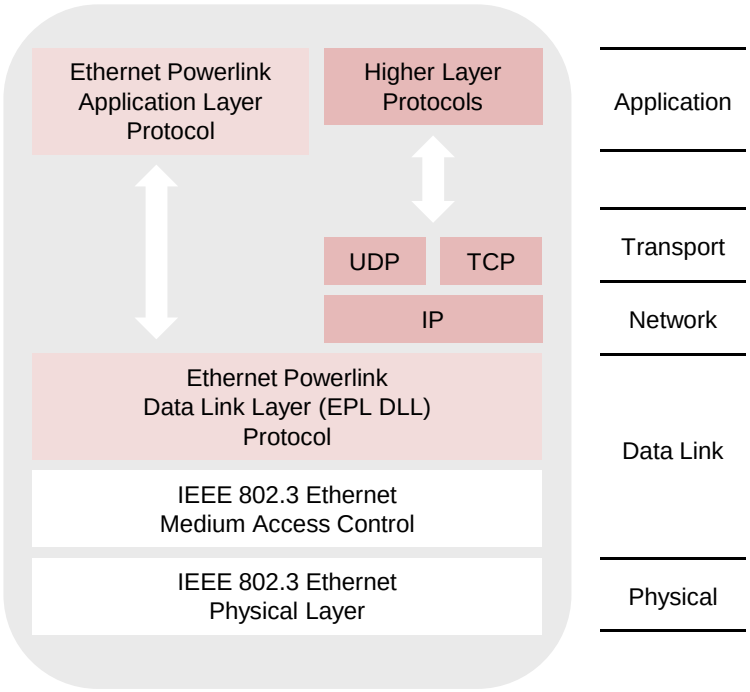


Figure 1.1: Simplified EPL network architecture with reference to the ISO/OSI model.

1.2.2 Physical layer

As shown in Fig. 1.1, the EPL physical layer is defined as IEEE 802.3 standard Ethernet physical layer. In particular, the EPL specifications indicate 100BASE-X (copper and fiber), half-duplex as transmission mode and standard patch cables (twisted pair, S/UTP, AWG26) with either RJ-45 or M12 connectors for the connection of EPL devices. Finally, the EPL specifications recommend to refer to the IAONA standard [5] installation guidelines.

1.2.3 Network topology

Connections between EPL stations are obtained through either traditional Ethernet hubs or switches. It is worth mentioning that, at present, the EPL specifications encourage the use of hubs since they guarantee limited latencies (less or equal to 460 ns) and jitters (less or equal to 70 ns) and discourage the use of switches since they may introduce additional non deterministic delays. However, due to the evolution of Ethernet technology, hubs are rapidly disappearing from the market and, consequently, their costs are increasing. Moreover, in many applications (e.g. factory automation systems and production plants),

switches that are already deployed for supporting other communication tasks could be used by the EPL network as well. It has also to be considered that, the announced (enhanced) version of EPL, running at 1 Gbit/s, will make use of switches. For the above reasons, we will not limit our analysis to network configurations based on hubs, but we will consider switched Ethernet networks as well. In this latter case, additional latencies and jitters have to be considered during the off-line network configuration phase.

Concerning network topology, since collisions are avoided, the IEEE 802.3 standard Ethernet constraint of 5120 ns as maximum round trip time has no more to be satisfy by EPL networks. As a consequence, line topologies with a large number of nodes, typical of industrial applications, are allowed. Other common topologies like star, tree, or hybrid tree-line configurations are permitted as well, see for example Fig. 1.2.

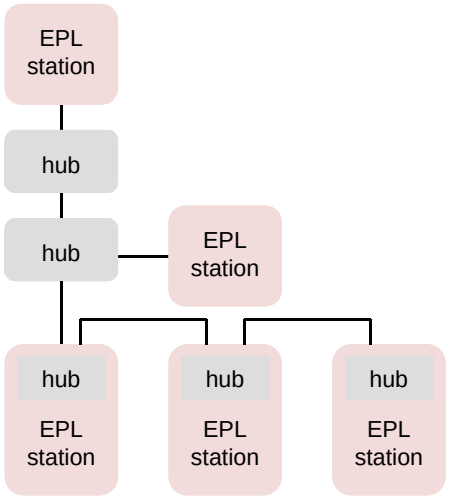


Figure 1.2: Example of EPL network topology.

1.2.4 Data Link layer

EPL makes use, without applying any change, of Ethernet MAC layer; as a consequence, the technique used by EPL devices to access the transmission medium is the well-known carrier sense multiple access/collision detection (CSMA/CD) technique.

The EPL specifications define two different operating modes for an EPL network:

- *EPL mode*

  When an EPL network operates in the *EPL mode* the network DLL

behavior is defined by the EPL protocol. As it will be clarified in the next subsection, the EPL protocol defines a transmission medium access managed by a master station (Managing Node, MN), unique within an EPL network. Only the MN can give a slave station (Controlled Node, CN) the right to transmit. This centralized transmission medium access technique allows the EPL network to avoid collisions. Consequently, the collision-resolving mechanism defined by the CSMA/CD, responsible for the randomness of Ethernet transmissions, is no longer employed and the the communication results deterministic.
In the EPL operating mode the communication takes place mainly through EPL frames, nonetheless an asynchronous time slot is dedicated to non-EPL frames exchange. The most used higher layer protocol in the asynchronous time slot is UDP/IP [13, 14], however any other protocol can be employed.

- *Basic Ethernet mode*
  When an EPL networks operates in the *Basic Ethernet mode*, the network DLL behavior is defined by the IEEE 802.3 standard. The employed transmission medium access technique is therefore CSMA/CD without any modification. In this case, as it happens for general purpose Ethernet networks, collisions are possible and, as a consequence, the communication is non-deterministic.
  Any higher layers protocol can be employed on top of Ethernet (i.e. UDP/IP, TCP/IP [15], etc.).

Clearly, we are interested in industrial communication applications of EPL and therefore we will specifically focus on the EPL operating mode.

1.2.5 EPL protocol

The EPL protocol specifies a transmission medium access based on a time division multiple access (TDMA) technique which guarantees each EPL station the exclusive access to the transmission medium, avoiding frame collisions. In the EPL specifications this transmission medium access technique is referred as *Slot Communication Network Management (SCNM)* and is managed by a specific station, referred as Managing Node. Indeed, two different types of stations are specified for an EPL network: Managing Node (MN) and Controlled Nodes (CNs). Each EPL network contains exactly one MN and several (up to

239) CNs. The MN operates as a master station and is allowed to transmit frames independently, while the CNs operate as slave stations and are only allowed to transmit frames when requested by the MN.

In detail, the communication between the stations of an EPL network occurs on the basis of a cycle, the *EPL cycle*, managed by the MN and continuously repeated. The duration of the EPL cycle, the *EPL cycle time* τ_{EPL}[1], is defined by the user during an off-line network configuration phase and is maintained constant, that is τ_{EPL} does not change during the network operation phase.

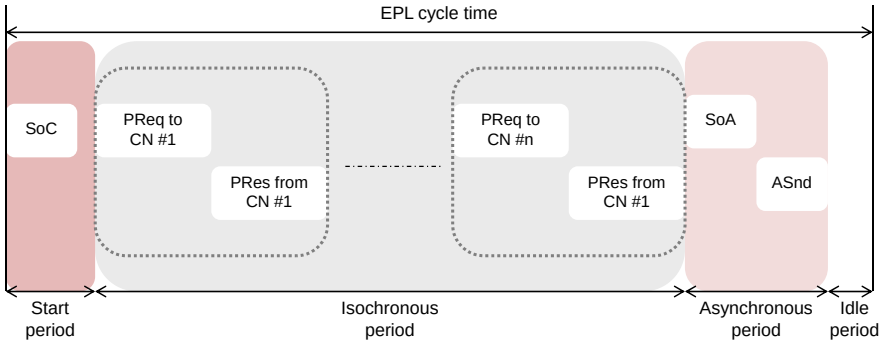


Figure 1.3: EPL cycle.

Each EPL cycle consists of four different periods, as shown in Fig. 1.3:

1. *Start period*

   The *start period* is the beginning of the EPL cycle. At the beginning of each cycle, the MN sends (broadcasts via Ethernet multicast) a Start of Cyclic (*SoC*) frame in order to synchronize the CNs.

   The *SoC* frame is the only periodic EPL frame, that is it is the only frame independently generated and transmitted over the network every τ_{EPL}, all the other EPL frames are event-driven frames, that is they are generated and transmitted over the network in response to particular events (e.g. the reception of a frame, the expiration of a time interval, etc.). The start period ends with the end of the *SoC* frame transmission.

2. *Isochronous period*

   The *isochronous period* is a time slot dedicated to the real-time data exchange between the EPL stations.

   After the SoC frame has been transmitted and the synchronization between the EPL stations has been reached over the network, the cyclic

[1] Greek letters are used to indicate values defined by the user.

real-time data exchange is realized by means of a sequential polling. In detail, the MN polls each CN by sending it a Poll Request (*PReq*) frame and waiting for a Poll Response (*PRes*) frame before moving to the next CN. The reception of a *PReq* frame by a particular CN indicates the beginning of the time slot dedicated to the real-time data transmission from that CN.

A *PReq* frame is always a unicast Ethernet frame since only the target CN has to receive it and it always contains data addressed by the MN to the target CN. Conversely, *PRes* frames are always multicast Ethernet frames, so that all the CNs can receive the real-time data sent over the network by a particular CN. This procedure allows producer/consumer relationships between the EPL stations.

In order to avoid undesired (and potentially dangerous) delays, the EPL protocol states that the polling of each CN has to be concluded within a fixed time interval, the *timeout time* τ_{to}, specified by the user for each CN during the off-line network configuration phase. If τ_{to} expires before the reception of the *PRes* frame from the polled CN, then the MN moves on to the next CN.

Once that all the CNs have been polled, the MN may broadcast a *PRes* frame in order to transmit data relevant for groups of CNs (note that not all the MNs support the transmission of *PRes* frame).

3. *Asynchronous period*

   The *asynchronous period* is a time slot dedicated to the non real-time data exchange between the EPL stations.

   After the isochronous period has ended, the *asynchronous period* is entered and the MN sends to one of the CNs, that has made request for it during the isochronous period, a Start of Asynchronous (*SoA*) frame to allow the CN to transmit one single frame containing non time-critical data. Two types of asynchronous frame are allowed: Asynchronous Send (*ASnd*) frames, that use the EPL addressing scheme and can be either unicast or broadcast frames, or standard Ethernet frames.

   The *SoA* frame indicates to the CNs that the isochronous period has been concluded and the asynchronous period has started. If no CN has made request for an asynchronous transmission, the MN sends a *SoA* frame without assignment of the right to transmit to any CN.

4. *Idle period*

Finally, the *idle period* is the time interval between the end of the asynchronous transmission and the beginning of the next EPL cycle. During the idle period all the EPL stations simply wait for the next *SoC* frame.

The EPL cycle time, τ_{EPL}, represents the minimum sampling period for the real-time data that are collected by the CNs and transmitted every cycle to the MN as well as for the real-time control data transmitted every cycle by the MN to the CNs. Consequently, it is important to keep the start time of an EPL cycle as precise and without jitter as possible. During the off-line network configuration phase, the user can set the value of τ_{EPL}; in order to correctly configure an EPL network, the EPL cycle time has to be chosen carefully, taking into account the times necessary to send the *SoC* and the *PReq* and *PRes* to all the CNs, as well as the elaboration delays typical of each node, so that τ_{EPL} would not be exceeded during the network operation phase.

The EPL cycle is given by[2]:

$$\tau_{EPL} = t_{st} + T_{is} + T_{as} + T_{id} \tag{1.1}$$

where t_{st}, T_{is}, T_{as} and T_{id} are, respectively, the duration of the start, isochronous, asynchronous and idle period. The value of t_{st} is constant and can be computed as the time necessary to transmit the *SoC* frame plus a safety margin necessary to guarantee that all the stations are synchronized. The EPL specifications mention 45 μs as a typical value.

The value of T_{is} within an EPL cycle depends, in general, on:

- the number of CNs,
- the elaboration delays introduced by the MN, the CNs and the network interconnection devices
- the amount of time-critical data transmitted by each CN and by the MN
- the transmission/propagation delays (that depends on the network topology, on the used interconnection devices, etc.)

A general expression of T_{is} is the following

$$T_{is} = t_{tx} + t_{prop} + D_{el} \tag{1.2}$$

[2]In this chapter, capital letters are used to indicate random variables.

where t_{tx} is constant and indicates the sum of transmission times of all *PReq* and *PRes* frames, D_{el} is a random variable that accounts for delays introduced by all the network components, and t_{prop} is constant and accounts for propagation delays on cables.

The expression of T_{as} is analogous to that of T_{is}, while T_{id} clearly depends on the particular EPL cycle considered.

In order to correctly choose the EPL cycle time, it is important to correctly estimate the value of the sum in Eq. (1.1) for the considered network configuration and add a safety margin. The EPL specifications give some indications about the values typically assumed by some of the parameters of Eqs. (1.1) and (1.2). A possible way to compute the margin could be evaluating the maximum allowed value of T_{is}, which is trivially obtained by imposing that no CN responds to the MN within the timeout τ_{to}:

$$T_{is,max} = n \times \tau_{to}.$$

Unfortunately, typical timeout values are not specified by the EPL specifications.

Computing the time necessary to poll each CN of the EPL networks is also a key step in order to correctly configure the corresponding timeout time τ_{to}. The timeouts technique is an effective procedure that belongs to the class of cyclic polling protocols [48]. Nonetheless, it cannot avoid the occurrence of jitters in case frames are delayed or even lost, as pointed out in the EPL specifications. Indeed, when a *PRes* frame sent by a CN is either lost or delayed beyond τ_{to}, then the duration of the relevant isochronous period becomes longer than the expected one (i.e. the time necessary to send both the *PReq* frame from the MN and the *PRes* from the CN, for all CNs). Moreover, a late *PRes* frame transmitted while the MN is polling a different CN, might collide with other transmitted frames. Consequently, the isochronous period may be affected by a jitter which influences the duration of the whole EPL cycle as well. Each time the jitter on the EPL cycle exceeds a threshold value, the CNs increment a counter and, when the count limit is reached, the state of the whole network is switched from "operational" to "pre-operational" and the cyclic data exchange is stopped. It turns out that maintaining jitters as low as possible is fundamental for the effective performance of the whole network.

1.2.6 Communication classes

The EPL specifications define different communication classes in order to provide the possibility of polling a CN with frequencies other than every EPL cycle. In particular, a CN may belong to one out of two communication classes, corresponding to two different polling frequencies:

- *Continuous*: the CN is polled by the MN every cycle (the polling frequency is $1/\tau_{EPL}$) ;
- *Multiplexed*: the CN is polled by the MN every η cycles, $\eta \in \mathbb{N}$ (the polling frequency is $1/\eta\tau_{EPL}$).

The multiplexed communication class allows, in principle, the user to implement EPL networks with a large number of CNs maintaining low values of τ_{EPL}. Moreover, it is worth observing that, although the multiplexed CNs are not polled every EPL cycle, they can still cyclically receive data from the other CNs via the *PRes* frames. An example of multiplexing is provided in Fig. 1.4.

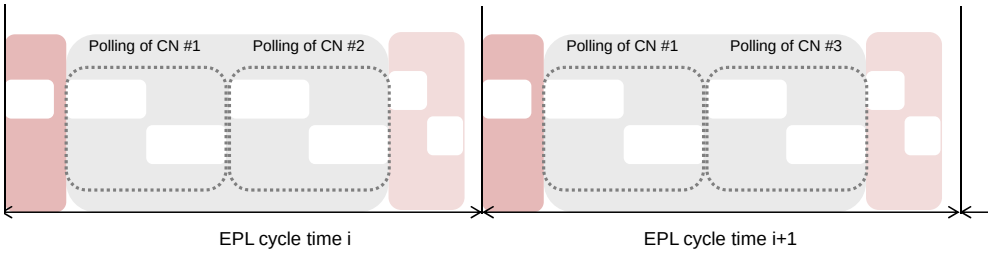


Figure 1.4: Example of multiplexed EPL cycle: CN #1 is continuously polled, while CNs #2 and #3 are multiplexed.

1.2.7 EPL frames

EPL frames are encapsulated and transmitted in the Data field of IEEE 802.3 standard Ethernet frames (see Fig. 1.5). The Ethernet Type field of the Ethernet frame in case it contains an EPL frame assumes the hexadecimal value $88AB_h$. As can be seen in Fig. 1.5, an EPL frame consists of five fields: the first octet, namely Message Type, specifies the type of the EPL frame. To this regard, Table 1.1 illustrates the correspondence between the hexadecimal value of the Message Type field and the different types of EPL frames. The second and third octets specify, respectively, the source and the destination EPL address, while the remaining octets are reserved to the data to be exchanged.

It is worth observing that, since the EPL protocol does not modify the Ethernet frame structure, communication via standard TCP/IP services is possible as well, with consequent straightforward integration with all factory communication systems.

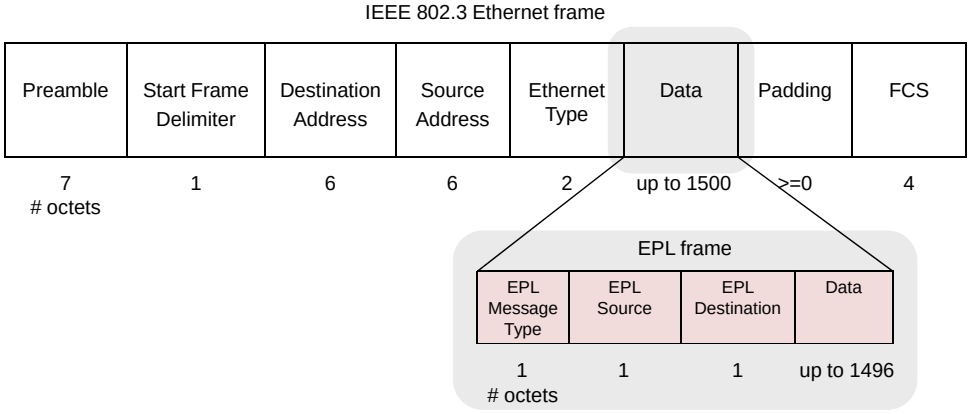


Figure 1.5: EPL frame structure.

EPL frame type	**Message Type field**
Start of Cyclic (*Soc*)	01_h
Poll Request (*PReq*)	03_h
Poll Response (*PRes*)	04_h
Start of Asynchronous (*SoA*)	05_h
Asynchronous Send (*ASnd*)	06_h

Table 1.1: Value of Message Type field of an EPL frame.

1.2.8 EPL addressing

Each EPL station (either MN or CN) of an EPL network has a unique EPL address (Node ID). The EPL address 240 is always assigned to the MN, while EPL Node IDs in the range 1-239 are assigned to the CNs. To this regard Table 1.2 illustrates the complete EPL Node ID assignment.

1.2.9 Application layer

As described in Fig. 1.1, the EPL specifications define an application layer protocol based on the CANopen standard which defines a set of communication objects to be exchanged over the network as well as a set of communication

EPL Node ID	Description
0	Invalid
1-239	CN
240	MN
241-252	Reserved
253	Diagnostic device
254	EPL to legacy Ethernet router
255	EPL broadcast

Table 1.2: EPL node ID assignment

services. Each EPL station contains a specific object, namely the Object Dictionary, which is responsible of the interface between the application within the EPL station itself and the definition of the communication objects. In particular, the Object Dictionary contains the list of the objects belonging to an EPL station.

Data relevant to the application are stored and transferred via Process Data Objects (PDOs). The EPL specifications distinguish between PDOs that are transmitted (TPDOs) and received (RPDOs) by the EPL station. Thus, looking at the isochronous period of the EPL cycle of Fig. 1.3, the MN sends one or more TPDOs via a *PReq* frame to the addressed CN which, in turn, stores the data in one or more RPDOs and responds with a *PRes* frame carrying one or more TPDOs.

Although the analysis of the EPL application layer protocol may be interesting, it will not be considered further since we are mainly interested in the EPL data link layer, which is largely responsible for the real-time behavior of the whole communication protocol suite.

1.3 Real-time acyclic traffic handling

Although the EPL cycle presents a specific asynchronous period, the handling of real-time acyclic traffic such as that generated, for example, by process alarms is not explicitly mentioned by the EPL specifications. Indeed, the asynchronous period is dedicated to generic requests, typically deriving from TCP/IP traffic. However, alarms are often critical information which need to be transferred within specified deadlines [36].

Thus, looking at the EPL cycle, a suitable way to handle alarms is to transmit them in the isochronous period using, for example, the *current* technique described in [66]. In practice, when one or more alarms occur at a CN, the CN replaces the cyclic data in the TPDOs with the alarms and transmit them to the MN. Obviously, with such a procedure, input data are not updated for one cycle and, consequently, this technique is not applicable in case of frequent occurrences of alarms. As an alternative, a fixed space could be allocated by each CN in the cyclically transmitted TPDOs and used to send alarms at their occurrence. This technique, however, may cause a consistent wasting of bandwidth, since alarms are typically rare events and hence their transmission is sporadic.

Another, more immediate, choice is to reserve a portion of the asynchronous period for time-critical acyclic traffic. This resembles the *late* technique described in [66] and employed, for example, by Profibus DP [18]. The implementation of such a technique for EPL is relatively simple. Indeed, during the isochronous period each CN may issue a request to the MN to transmit a real-time acyclic data. Then, the MN, by means of a purposely defined additional frame, may grant the requesting CNs with the access to the network for the transmission of alarm messages.

It is a task of the MN to determine the portion of the asynchronous period to be reserved to real-time acyclic messages. In principle, such an assignment could be dynamic, depending on the specific network load. In any case, a correct bandwidth assignment, ensures that the EPL cycle is maintained constant. It is worth remarking that the implementation of the aforementioned technique guarantees total compatibility with the existing EPL protocol. In particular, enhanced CNs (i.e. those capable of handling alarms) can coexist on the same network with those compliant with the original version of EPL.

1.3.1 Simulation results

We simulated the behavior of the EPL network by means of a commercially available tool, namely Opnet Modeler [54]. In this context, we developed simulation models for the MN, the CNs and the interconnection devices that may be employed by an EPL network. In particular, we used standard models provided by Opnet for the interconnection devices (Ethernet hubs, switches, etc.). Conversely, specific models have been developed for the EPL stations starting from IEEE 802.3 standard Ethernet stations provided by Opnet and

implementing the EPL protocol on the top of IEEE 802.3 standard Ethernet MAC. Moreover, we implemented the real-time acyclic traffic handling technique previously described, reserving a portion of the asynchronous period for the transmission of alarms, if any.

We focused on the specific network configuration shown in Fig. 1.6, comprising one MN connected to 7 CNs through a maximum of three levels of hubs. It is obvious that different configurations behave in different ways, so that the results presented are clearly pertinent to the specific configuration we adopted. Nonetheless, as it will be shown, the simulation model we developed is suitable to evaluate, at least preliminary, the effectiveness of the proposed alarm handling technique and also the behavior of more complex configurations. Using Eq. (1.1) and supposing that only minimum size Ethernet frames are exchanged over the network, an indicative value for the cycle time of the network configuration of Fig. 1.6 results 0.4 ms. Thus, as a preliminary test, in agreement with some worst case values cited in the EPL specifications, we set $\tau_{EPL} = 0.4$ ms in our simulation set-up and we reserved 0.09 ms for the asynchronous period.

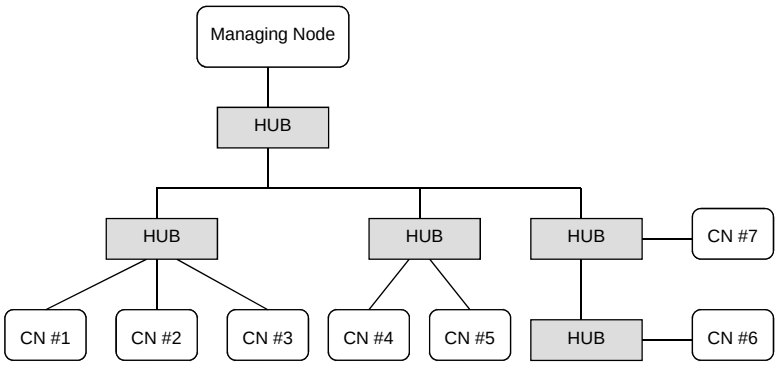


Figure 1.6: Network configuration considered for simulation.

The CNs generated alarms according to a Poisson point process with different averages. The intensities of the Poisson processes have been chosen in order to generate, on average, two alarms out of 7 CNs per cycle. Without loss of generality, we reserved all the duration of the asynchronous period to handling of alarms (i.e. the TCP/IP traffic has not been considered at this stage). Moreover, a maximum of two alarms were allowed to be served during the asynchronous period. The behavior of the EPL cycle obtained from the simulation is shown in Fig. 1.7.

As can be seen, there is a good accord with the theoretical data. In par-

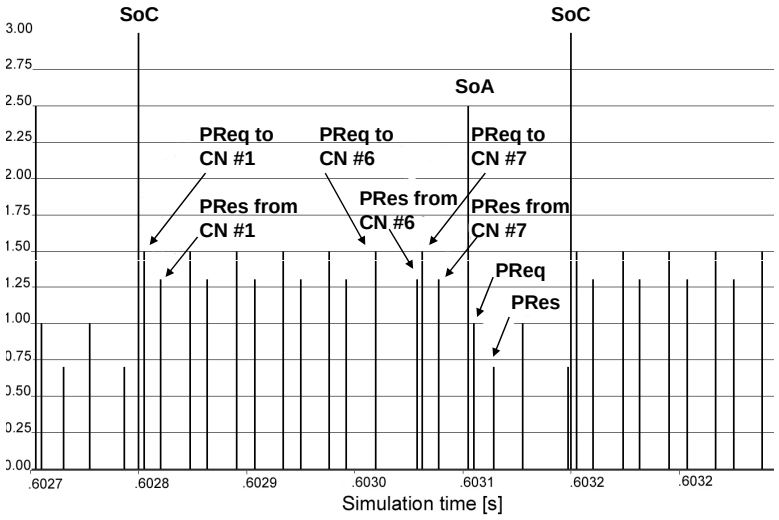


Figure 1.7: Simulation of the EPL cycle.

ticular, the cycle takes exactly 0.4 ms (evaluated as the difference between the two *SoC* frames). Moreover, all the *PReq* frames issued by the MN to the CNs in the isochronous period are correctly followed by a *PRes* frame. The times elapsed between *Preq* and *Pres* are sometimes different (in particular for CN #6) due to the different paths covered by the frames exchanged between MN and CNs. The same phenomenon has been observed for the alarms (i.e. those coming from CN #6 have the greatest latency).

Focusing on the asynchronous period, we observed that, as expected, two alarms are served per each cycle (actually, Fig. 1.7 refers only to one cycle and does not completely justify such a statement). However, this is more evidently confirmed by the mean number of CNs, in the overall network, waiting to transmit an alarm, which is shown in Fig. 1.8. It may be observed that, after an initial simulation transient, the mean number of nodes waiting to transmit an alarm rapidly converges to 2, which is the expected value (and also the maximum allowed by the simulation). This is a confirm that the duration of the asynchronous period is able to cope with the alarm rates we set. Of course, different alarm rates could be handled by modifying both the duration of the asynchronous period and the maximum number of alarms managed per EPL cycle. Moreover, depending on the requirements, in agreement with the original EPL specifications, an adequate bandwidth portion may always be dedicated to the TCP/IP traffic.

As a last performance index, we evaluated the mean alarm latency, which is shown in Fig. 1.9 (the different grey scales are referred to different CNs). As

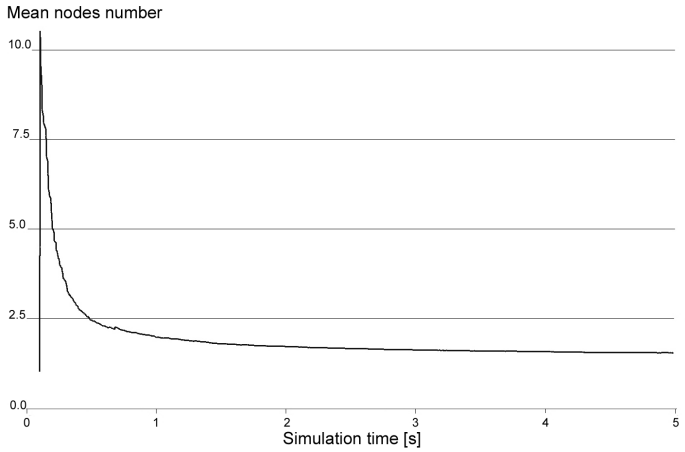


Figure 1.8: Mean number of CNs waiting for sending an alarm to the MN.

can be seen, after the initial transient, very low latencies may be observed for all the CNs. Indeed, values are comprised about in the range between 0.6 and 1 ms. This is, roughly speaking, an indication that alarms generated during one EPL cycle are, on average, served in the immediately following one. In practice, since the MN uses a FIFO queue, when an alarm occurs, the CN generates the request for an acyclic transmission. This is typically queued after the two requests (on average) occurred during the previous EPL cycle, which are going to be served at the end of the current cycle.

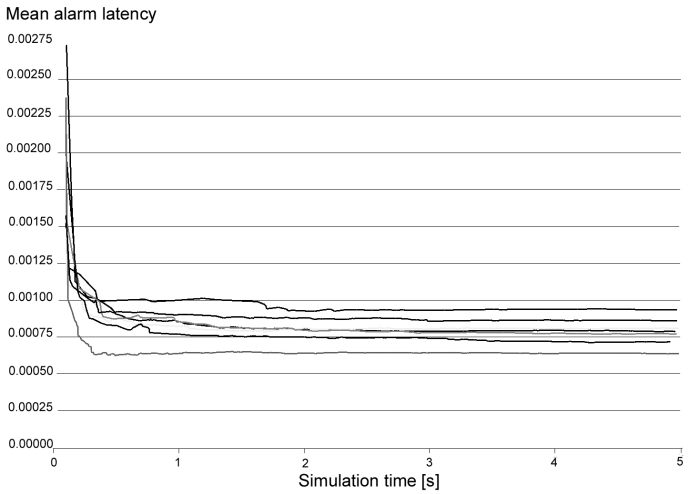


Figure 1.9: Mean number of alarms waiting at each CN.

1.4 Performance analysis of EPL networks

We carried out some examples of performance analysis for the two network configurations shown in both Fig. 1.10 and Fig. 1.11. Clearly, several other network configurations could have been considered. Nonetheless, the ones we chose are good representatives of a large number of industrial communication applications. Indeed, as outlined in [51], the one-level configuration of Fig. 1.10, based on a single Ethernet hub, is typically employed at the device level of automation systems and/or by networked control systems. Conversely, the configuration of Fig. 1.11, a three-level switched Ethernet network, refers to more complex network architectures such as, for example, Computer Integrated Manufacturing (CIM) systems [68]. In this case, from a conceptual point of view, it is possible to suppose that switches are employed to interconnect devices operating at factory, cell and field level, respectively ,of the plant.

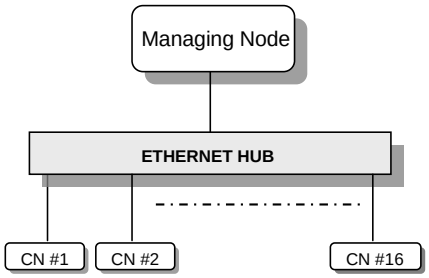


Figure 1.10: One-level configuration.

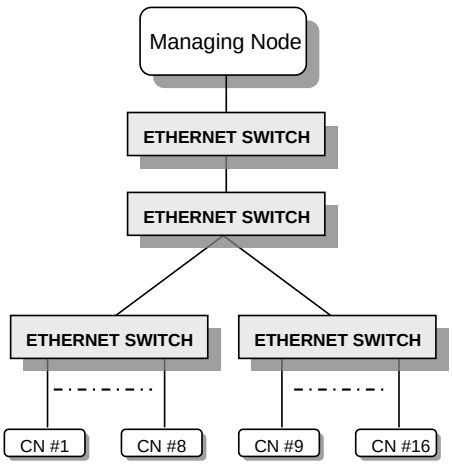


Figure 1.11: Three-level configuration.

Considering the importance of alarms handling in real industrial communication applications, the performance analysis we carried out refers exclusively to the modified version of the EPL network proposed in section 1.3. For both configurations, we supposed that only minimum size Ethernet frames (64 bytes) are exchanged, as it is usual for industrial communication applications [64]. Indeed, the size of process data is typically small (few bytes) and the EPL protocol adds only 3 bytes on its own. On the other hand, small payloads for Ethernet frames are padded so that the total Ethernet frame length is never shorter than 64 bytes.

In order to evaluate the performance provided by the two network configurations, we focused on two parameters: the duration of the isochronous period, T_{Is}[3] and the alarm latency, T_a. The duration of the isochronous period is the time actually employed by the MN to execute a complete polling of all the CNs and it is measured as the time elapsing between the sending of the *SoC* frame and the arrival of the last *PRes* frame from the CNs. The alarm latency is defined as the time elapsing between the generation of an alarm at a CN and the arrival of the corresponding *ARes* frame to the MN during the asynchronous period.

1.4.1 Ideal operating conditions

We firstly carried out the performance analysis of the two EPL networks shown in Figs. 1.10 and 1.11 under ideal operating conditions, that is we supposed the complete absence of transmission errors so that each transmitted frame was received correctly by the intended destination(s).

- *One-level configuration*
  According to Eq. (1.2), the duration of the isochronous period for the one-level configuration, neglecting cable propagation delays, is given by:

$$T_{is}^{1l} = t_{st} + n \times (D_{MN} + D_{CN}) + 2n \times (b \times t_{tx}) \tag{1.3}$$

  where n is the number of CNs of the EPL network. D_{MN} is the time elapsing between the reception of a *PRes* frame by the MN and the instant the *PReq* frame is issued to the next CN. Similarly, D_{CN} is the time employed by a CN to generate a *PRes* frame after the reception

[3] In this Section, with a (slight) abuse of notation, we refer with isochronous period T_{Is} to the time interval that is the sum of the start period T_{St} and the actual isochronous period T_{Is}.

of the *PReq* frame from the MN. We set $D_{MN} = 8$ μs and $D_{CN} = 8$ μs respectively, since these are specified as typical values by the EPL standard. b is the length of frames (64 bytes in our case) and, finally, t_{tx} is the time necessary to transmit one byte (0.08 μs at a transmission rate of 100 Mbit/s).

Eq. (1.3) can be used to compute T_{is}^{1l} when the number of CNs varies, in order to set the value of the EPL cycle time. However, since T_{is}^{1l} may be influenced by transmission delays, in order to ensure the correct operation of the protocol, an appropriate safety margin has to be added to the value of T_{is}^{1l} obtained from Eq. (1.3). We considered as the maximum $T_{is,max}^{1l}$, the value obtained from Eq. (1.3) increased by 20%, which seems to be a reasonable safety margin.

We used the Opnet models of the EPL stations described in section 1.3 and run extensive numerical simulation sessions. The behavior obtained for T_{is}^{1l} versus the number of CNs is shown in Fig. 1.12 where the maximum theoretical value $T_{is,max}^{1l}$ is plotted as well.

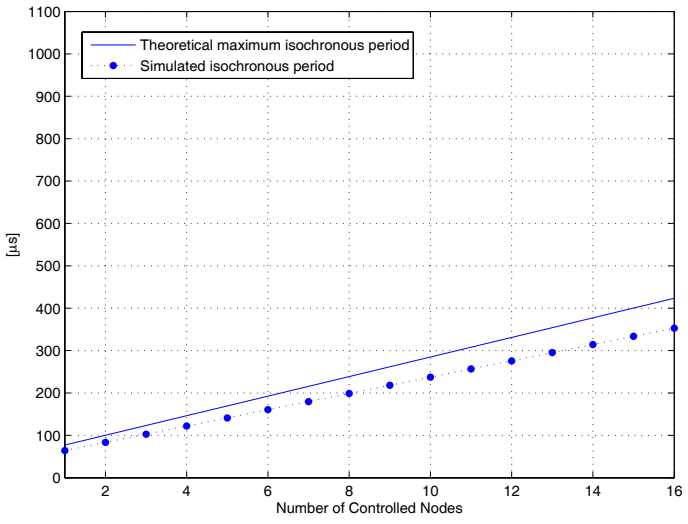


Figure 1.12: Duration of the isochronous period for the one-level configuration.

The sequence of actions relevant to the transmission of an alarm is shown in Fig. 1.13. Consequently, the alarm latency can be expressed as:

$$T_a^{1l} = T_g + T_w + D_{CN} + 2 \times (b \times t_{tx})$$

where T_g is the time elapsed between the actual occurrence of the alarm at the CN and the notification of its presence to the MN (this takes

place when the CN is polled by the MN): if the alarm occurs after the *SoC* and before the CN is queried, then it is notified in the current EPL cycle; conversely, the notification takes place in the following cycle. Hence, T_g is a random variable and its stochastic description depends on the distribution of the alarm generation process as well as on the value of τ_{EPL}. T_w is the time elapsing between the arrival of an alarm notification to the MN and the issuing of the related *AReq* frame, which actually requests the CN to transmit the alarm data via the *ARes* frame. T_w is a random variable as well and its stochastic description is determined by both the processes of alarm arrival and departure. In general, T_w represents the waiting time of the queuing system, resident on the MN, which handles the real-time acyclic traffic. D_{CN} has the same meaning as in Eq. (1.3), whereas the term $2 \times (b \times t_{tx})$ accounts for the transmission of both the *AReq* and *ARes* frames whose length is supposed to be the same of *PReq* and *PRes* frames.

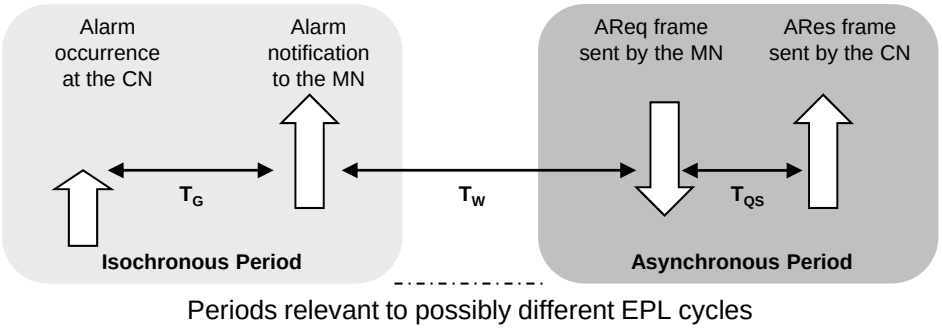


Figure 1.13: Sequence of actions relevant to an alarm transmission.

The alarm latency for the network shown in Fig. 1.10, with 16 CNs, has been obtained via simulation. According to Eq. (1.3), we set $T^{1l}_{is,max} = 423.4$ μs. The duration of the asynchronous period was set to 57.7 μs, which represents the time necessary to transmit 2 alarms (two pairs of *AReq*-*ARes* frames) plus a safety margin. We added a 15% idle period to the sum of the durations of both the isochronous and asynchronous periods (also in this case, the EPL standard does not specify a typical value) so that the EPL period was equal to 553.3 μs. We assumed that alarms were generated by CNs as independent Poisson counting processes with an overall intensity $\Lambda = 1$ alarm/EPL cycle. Consequently, the ratio between the mean alarm intensities of the arrival and departure

processes is $\rho = 0.5$, which ensures the stability of the aforementioned queuing system [49].

Results of the simulation are shown in Fig. 1.14, which reports the instantaneous alarm latencies for all the CNs; the mean value of latencies (not shown in Fig. 1.14), after a short initial transient, settles to 460 μs. This means that most alarms are transmitted within one or two EPL cycles since their occurrence. Clearly, higher values of ρ lead to worse performance; in this sense, the knowledge of the alarm generation rate is of significant importance, since it affects the correct dimensioning of the asynchronous period. It may be concluded that, in the context of the simulations carried out, the technique we have proposed for handling the real-time acyclic traffic proves to be effective since, besides allowing for the transmission of alarms with low latencies, it does not influence the duration of the EPL cycle.

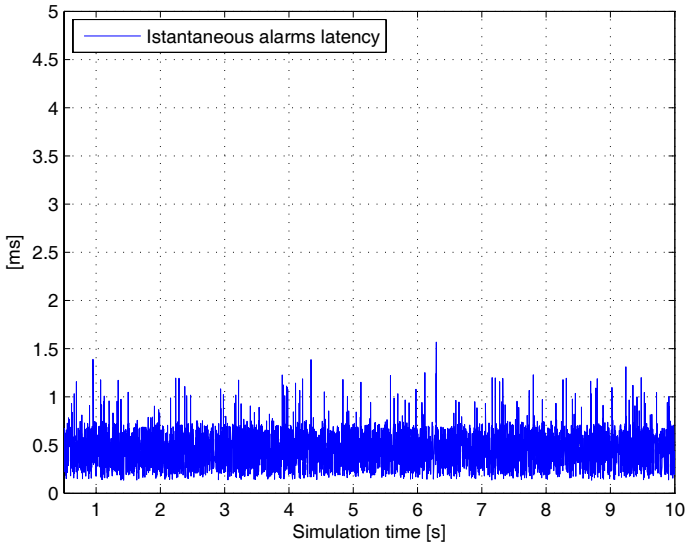


Figure 1.14: Instantaneous alarm latency for the one-level configuration.

- *Three-level configuration*

  Switches used in the configuration of Fig. 1.11 introduce additional delays which have to be taken into consideration in evaluating the considered parameters. Indeed, a frame that propagates from one switch to another, up to the destination, has to be re-transmitted by each switch it traverses. Thus, the duration of the isochronous period may be obtained

by modifying the last term of Eq. (1.3) as follows:

$$T_{is}^{3l} = t_{st} + n \times (D_{MN} + D_{CN}) + 2n \times (4 \times b \times t_{tx}) + D_s \qquad (1.4)$$

where the term $(4 \times b \times t_{tx})$ takes into account the frame transmission times (indeed, both the *PReq* and the *PRes* frames have to be transmitted four times before reaching the destination). To be fair this is true only when store-and-forward switches are used, however our choice appears to be reasonable for two main reasons: first cut-through switches are seldom used in industrial environments and they are likely to be discarded in the future. Second, the duration of the isochronous cycle computed by means of Eq. (1.4) represents the worst-case value and can be considered a sort of upper bound for this kind of performance index. Finally, D_s is the total latency introduced by the switches. According to [51], D_s mainly depends on the queuing delay, which, for each port of a switch, is determined by the number of frames waiting to be transmitted on that port. However, because of the characteristics of the EPL protocol, queuing of frames in a switch can never occur, since a frame (e.g. *PReq*) can only be transmitted if either the previous one (*PRes* , in this example) has been correctly received or the time out has expired. For such a reason, the contribution of D_s can be neglected in Eq. (1.4).We simulated the behavior of the isochronous period when the number of CNs is varied. Similarly to the one-level configuration, we set $T_{is,max}^{3l}$ to the value obtained from Eq. (1.4) plus a 20% safety margin. Results are reported in Fig. 1.15, where the delay introduced by the switched three-level configuration is shown.

It may be noticed that T_{is}^{3l} is below 1 ms even when a significant number of controlled nodes is considered and this is an interesting result for real-time distributed automation and control applications.

The alarm latency has also been investigated for the three-level architecture and, in particular, for the configuration of Fig. 1.11 with 16 CNs. We set the duration of the asynchronous period to 149 μs which, analogously to the previous example, allows for the transmission of two alarms (a safety margin has been included in this case too). The resulting EPL cycle is 1337 μs (including, also in this case, a 15% idle period). The behavior of the alarm latency is shown in Fig. 1.16. It is worth noticing that instantaneous values are mostly below 3.5 ms, whereas the mean

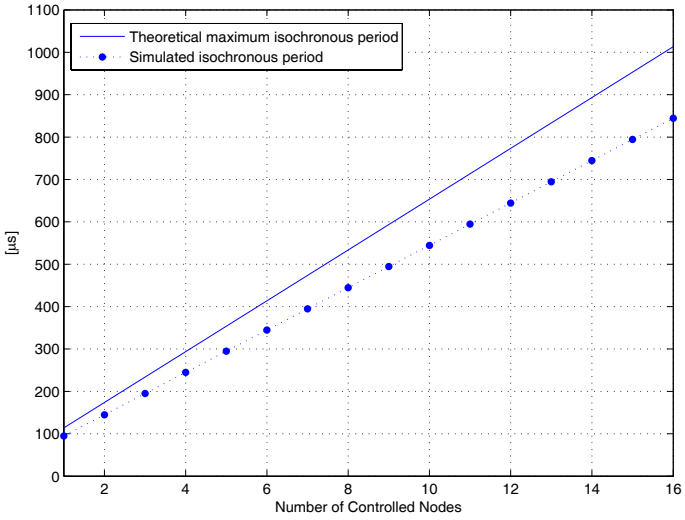


Figure 1.15: Duration of the isochronous period for the three-level configuration.

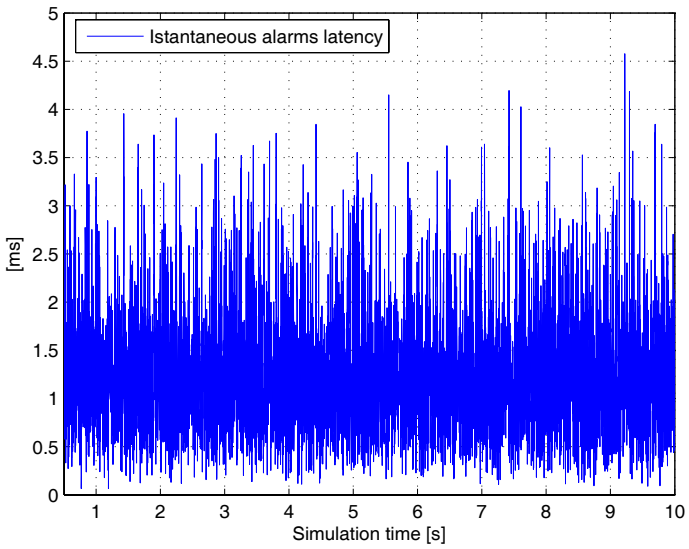


Figure 1.16: Instantaneous alarm latency for the three-level configuration.

value is 1.3 ms. These results confirm the effectiveness of the protocol and particularly of the alarm handling technique.

1.4.2 Non-ideal operating conditions

Industrial environments are often characterized by the presence of several sources of electromagnetic noise. Consequently, networks deployed in these environments are influenced by transmission errors that may have dangerous effects on their performance and hence on those parts of the control systems where networks are employed. In general, the behavior of communication networks in hostile and potentially error prone environments is a topic of significant importance. However, strange enough to say, it has not yet been considered adequately in the scientific literature, as pointed out in [41][4] For such a reason, it is quite difficult to find theoretical models suitable to describe communication channels for industrial networks. Nevertheless, some useful insights may be obtained from analysis carried out mainly for wireless systems but that can be applied to wired networks as well.

Indeed, both papers [71] and [34] show that interference signals appear typically as bursts of variable duration and intensity. Thus, in principle, the communication channel may assume two states: "good" and "bad". In the good state, the error probability is very low and, consequently, transmissions are not affected by errors. Conversely, the bad state is characterized by a higher error probability which may cause corruption, loss and duplication of frames. Such a two-state channel is effectively described by the well-known Gilbert-Elliot model [47].

The effects of transmission errors clearly depend on the specific protocol adopted. Focusing on Ethernet, the integrity of a frame is checked by the station that receives it, and it is based on the Cyclic Redundancy Check (CRC) field inspection. In case of corruption, the frame is simply discarded. Since the EPL protocol uses Ethernet frames, this results in the unsuccessful transmission and/or reception of the EPL frames. For example, if a *PReq* frame sent by the MN is corrupted, then the addressed CN does not receive it and, consequently, it cannot issue the *PRes* frame. Thus, the MN after the time out expiration, moves on to the next CN. Similarly, if a *PRes* frame sent by a

[4] As a remarkable exception, in [70] the behavior of the PROFIBUS data link layer has been analyzed under these conditions; the results presented in such a study confirm the possibility of causing serious drawbacks to the protocol operation.

CN is missed, then the MN considers the polling of such a CN as failed. Even worse effects are caused by the corruption of the *SoC* frame: in this case the whole EPL cycle is aborted since the CNs could not be synchronized.

We simulated the behavior of the EPL configuration of 1.11, with 16 CNs, for a two-state channel. The operational parameters of the network were the same as for the previous example ($\tau_{EPL} = 1337$ μs and up to 2 alarms served in the asynchronous period). We assumed that in the good state frames are transmitted correctly, whereas in the bad state each frame is corrupted and hence the communication is inhibited. Error bursts are generated according to a Poisson counting process and their duration is a random variable uniformly distributed in the EPL cycle. Since the simulation model currently implemented is only able to generate a burst within an EPL cycle, the results we are presenting do not include situations in which the *SoC* frame is not received by one or more CNs and, in this sense, they do not show the worst case behavior that could be actually expected. Nonetheless, we believe that, in any case, such results provide useful insights that may be confirmed by further enhancements of the simulation model. In the simulation, we assumed a mean occurrence of one burst every five EPL cycles. Clearly, this represents a specific example, different generation rates could be adopted, depending on the knowledge of the environment characteristics.

As a first important result from the simulation, we observed that the EPL protocol was able to recover from error bursts. In practice, when the channel state switches from bad to good, then the protocol resumes its operation correctly. This means that the timeout technique revealed to be effective in that a failed polling of a CN did not influence the following ones. Moreover, as far as the acyclic traffic is concerned, when the *SoA* frame is not received by a CN because of an error burst, the CN is not able to transmit the *ARes* frame in the same cycle. As a consequence, the alarm latency increases.

The duration of the isochronous period resulting from the simulation session is shown in Fig. 1.17. It can be seen that the expected value (844μs in this case) is affected by the jitter deriving from Timeout expirations triggered by the failed pollings which occur in presence of error bursts. However, as Fig. 1.17 clearly shows, the use of timeouts has the beneficial effect of introducing an upper bound to the isochronous period and hence to the jitter.
Similarly, error bursts also influence alarm latencies, as shown in Fig. 1.18. Values plotted in the figure are considerably higher than those obtained for

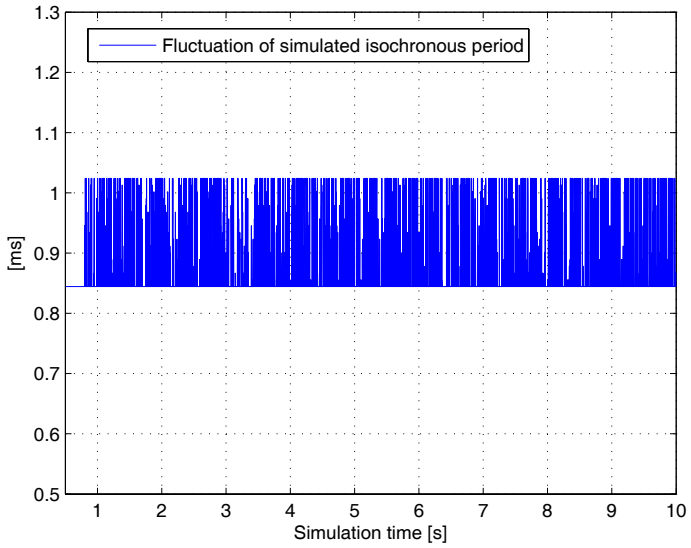


Figure 1.17: Duration of the isochronous period in presence of transmission errors.

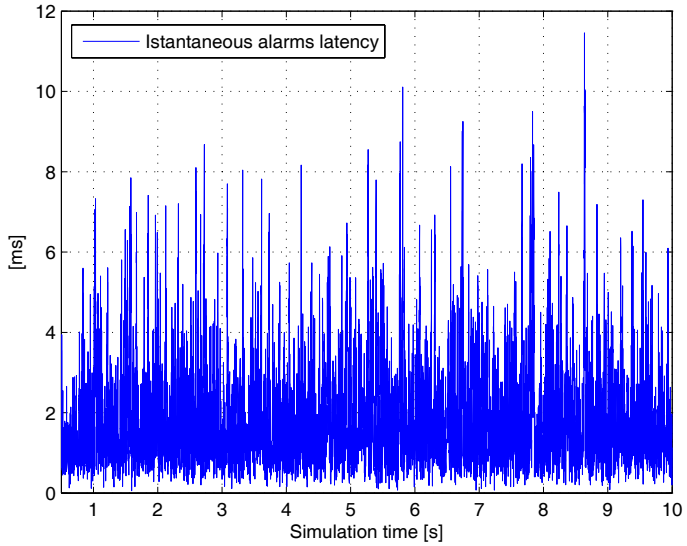


Figure 1.18: Instantaneous alarm latency in presence of transmission errors.

the ideal communication channel shown in Fig. 1.16.

1.5 Conclusion

Ethernet-based communication solutions have begun to be introduced in the factory and automation environments since about a decade. At present, many networks/subnetworks based on Ethernet technologies have been deployed in a lot of industrial scenarios to support distributed (non real-time) applications and functions such as remote monitoring, supervision and maintenance.

Real-time communications, such as those demanded by distributed control systems are, however, still a challenge that is currently being tackled by a number of competing (and incompatible) proposals by different manufacturers and consortia.

In this chapter we focused on the Ethernet Powerlink standard, which is one of the Ethernet-based technology whose popularity is rapidly increasing. In particular we were interested in the behavior of EPL with respect to its capability of managing real-time traffic. To this purpose two different kinds of real-time data exchanges have been considered, that is cyclic and acyclic. Our analysis has been aimed at obtaining some insights concerning the isochronous cycle time and the latency experienced by urgent (alarm) messages in two specific, but significant, system configurations, by means of a simple theoretical analysis and extensive simulation. Obtained results are encouraging, since they show that Ethernet Powerlink can be successfully employed also for applications with tight real-time requirements; moreover the behavior of the protocol, with regard to the alarm handling, can probably be further enhanced, by means of a slight modification.

2

Ethernet Powerlink extension based on the IEEE 802.11 wlan

2.1 Motivation

In the last years, the industrial communication scenario has been evolving thanks to the availability, at all levels of factory automation systems, of increasingly performing networks. In particular, focusing on the device level communication, besides fieldbuses [64] (purposely developed and traditionally employed) and real-time Ethernet networks [64] (purposely developed starting from Ethernet technology and increasingly employed), also wireless technologies are nowadays available and appealing. Clearly, wireless technologies offer several valuable advantages deriving from cabling avoidance such as: the possibility of connecting components that can not be reached (easily or reliably) by means of a cable (e.g. mobile components), decreasing costs for cabling, installation and maintenance, decreasing risk of cable/connectors failures, etc. Nonetheless, on the other hand, radio channels are intrinsically error prone and, as such, they are characterized by high bit error rates ($[(10^{-3}, 10^{-2}]$), and since errors may be caused by several different phenomena (path loss, fast and slow fading, noise and interference), by non-stationary channel error characteristic.

For this reason it seems unrealistic that wireless networks will completely replace wired industrial networks. Conversely, an immediate employment of wireless technology for IC is to realize wireless extensions of already deployed wired communication systems [39], in order to realize hybrid (wired/wireless) networks [72], tailored to the specific applications. The resulting configurations allow devices that can not (or are difficult to) be reached by means of

a cable to be connected to a wired communication system. Typical examples of applications are mobile devices (robots, cranes, etc.). Although hybrid systems may have, in general, various configurations, the aforementioned wireless extensions of wired networks, due to their nature, are usually characterized by the following peculiar features:

1. the controller (i.e. the station which is responsible of the traffic scheduling) is connected to a wired segment;

2. there is only one wireless segment and the stations connected to it are considerably less than those connected to the wired section(s);

3. the wireless segment has limited geographic extension (some tens of meters).

In this chapter we focus on the wireless extension of Ethernet Powerlink (EPL) [43], described in chapter 1, implemented through the IEEE 802.11 wlan citeieee80211std. The use of the IEEE 802.11 wlan is suggested by its features and, in particular, by the high transmission rate (54 Mb/s) that, in general, allows to maintain (or at least not significantly worsen) the real-time behavior of EPL. Moreover, the possibility of exploiting frame prioritization, recently made available by IEEE 802.11e standard, represents a further valuable opportunity in such a direction. Industrial applications of the IEEE 802.11 wlan have been yet extensively considered in the scientific literature; to this regard, some interesting results are provided for example in [71], [34] and the references therein.

The analysis we provide refers specifically to two different types of extensions: the first one, implemented at the data link layer, exploits the full compatibility of EPL with legacy Ethernet. Conversely, the second extension we propose is based on a gateway which represents one of the classical ways of interconnecting heterogeneous systems and, as such, may be used for whatever type of networks. After describing the possible implementation of the wireless extensions, we provide an analysis aimed at evaluating the performance they provide through the analytical and simulative computation of some performance indexes. Moreover, since the reliability of wireless networks may represent a critical aspect, the analysis is carried out taking into consideration the possible presence of interference and fading in the wireless segment.

In detail, the chapter is organized as follows. Section 2.2 illustrates some basic features of IEEE 802.11 wlan. Section 2.3 describes in detail the two types

of wireless extensions we propose as well as the specific network configuration considered. Section 2.4 focuses on the characteristics of the wireless channel, while section 2.5 provide some general considerations on the performance evaluation we carried out. Section 2.6 and 2.7 presents the actual performance evaluation of the hybrid networks based on a theoretical analysis as well as on numerical simulations. Finally, section 2.8 concludes the chapter.

2.2 IEEE 802.11 WLAN

The IEEE 802.11 wlan standard actually specifies a family of standards for wireless local area communication.

At the physical layer, IEEE 802.11 compliant devices may work in two different bands, either centered around the 2.45 GHz ISM band (802.11b, 802.11g, 802.11e) or centered around the 5 GHz band (802.11a), and may use different modulation techniques. The maximum transmission rate is 54 Mb/s (802.11a, 802.11g, 802.11e). At the MAC layer, two different transmission medium access methods are specified, namely distributed coordination function (DCF) (mandatory) and point coordination function (PCF) (optional). The PCF is based on a polling procedure executed by a station referred as Point Coordinator. The features of PCF, in particular the ordered access of the stations to the transmission medium, would make it particularly appealing for device level industrial communication but, unfortunately, this function is not implemented by the majority of IEEE 802.11 commercially available devices.

The DCF is based on the carrier sense multiple access/collision avoidance (CSMA/CA) technique with binary exponential backoff. The basic DCF transmission medium access method is summarized as follows. A station wishing to transmit listens to the wireless channel for a fixed time interval (distributed inter-frame spacing, DIFS). The time following an idle DIFS is slotted, and the station can start transmit only at the beginning of a slot time. The duration of a slot time, t_{slot}, is the time needed by any station of the network to detect a transmission from any other station. For this reason, t_{slot} is the sum of the propagation delay, the time needed by a station to switch from the receiving to the transmitting state (referred as RX_TX_Turnaround_Time in the standard), and the time to notify to the MAC layer the state of the channel (referred as busy detect time in the standard). If the wireless channel is free for a DIFS, then the station transmits. Conversely, a random backoff time is calculated by the station before transmitting.

The backoff counter is decremented every time slot the channel is idle, is stopped when the channel is busy and decremented again every time slot the channel is idle after being idle for a DIFS. The station transmits when the backoff counter is zero, provided that the medium remains free for a DIFS. At each transmission attempt the backoff is uniformly chosen as

$$B \in [0, w-1]$$

were w is the actual contention window. At the first transmission attempt $w = CW_{min}$, were CW_{min} is referred as minimum contention window, while after each failed transmission w is doubled, up to the maximum contention window $2^m CW_{min}$. The values of t_{slot}, CW_{min} and m are MAC layer related and are specified by the standard.

Since the CSMA/CA can not rely on the capability of the stations to detect a collision while transmitting, each transmitted frame has to be explicitly acknowledged by the receiver (except for broadcast transmissions). Thus, when a station successfully receives a packet, after a short inter-frame spacing (SIFS), it sends an acknowledgment frame back to the source. If the acknowledgment frame is not received within a specified time interval (referred as ACK_Timeout in the standard), then the transmission is considered as failed by the source which doubles its contention window and, after the expiration of the backoff counter, tries to transmit again. The contention window is restored to CW_{min} after every successful transmission.

It is worth pointing out that, as specified by the standard, a station executes the backoff procedure even at the end of a successful transmission of a payload of either "Data" or "Control" type. This means that after the completion of an acknowledged transmission, both the source and the destination stations wait for a backoff time before accessing again the physical medium, even if no additional transmissions are currently queued.

The IEEE 802.11 wlan standard also provides the possibility of adopting a quality-of-service (QoS) mechanism in data transmission[1]. The QoS mechanism is based on the definition of an additional channel access function, namely the hybrid coordination function (HCF) which includes two different access methods: contention-based, referred as enhanced distributed channel access (EDCA), and controlled channel access (HCF controlled channel access, HCCA). In particular, EDCA defines four access categories (ACs) that may

[1] Originally, QoS was introduced by Amendment 8 (IEEE 802.11e) of the wlan specification. Now, such a feature is explicitly encompassed by the final version of the standard [30].

be assigned to different types of traffic, and hence to frames. In other words, frames may be prioritized according to four different categories of traffic ranging from *Background* (the lowest priority traffic, belonging to Access Category 0, AC[0]) to *Voice* (the highest priority, belonging to Access Category 3, AC[3]). The actual prioritization of the frames is implemented assigning both shorter contention windows and reduced inter-frame spaces to higher priority frames. Thus, when employing EDCA, the value $CW_{min}[AC]$ replaces the CW_{min} introduced by DCF and new inter-frame spaces (arbitration inter-frame spaces, AIFS[AC]) are used instead of DIFS.

For commodity, Table 2.1 reports the most relevant IEEE 802.11 parameters used throughout this Chapter. Further details on the IEEE 802.11 wlan standard and the defined transmission medium access method can be found in [46] and references therein.

Parameter	**Description**	**Value**
r	Transmission rate	54 Mb/s
t_{slot}	Slot time	9 μs
DIFS	Distributed Inter-Frame Space	28 μs
SIFS	Short Inter-Frame Space	10 μs
CW_{min}	Minimum contention window	15
$t_{ack,out}$	ACK_Timeout	45 μs

Table 2.1: IEEE 802.11 parameters

2.2.1 IEEE 802.11 frames transmission time

In this subsection we give an example of computation of IEEE 802.11 frames deterministic transmission times. An IEEE 802.11 frame comprises an initial block of fields namely physical preamble and header, which are transmitted at the lowest rate (6 Mb/s) followed by the MAC Protocol Data Unit (MPDU) which is of variable length for data frames and of fixed length for acknowledgment frames. The MPDU may be transmitted at different rates. Supposing to adopt the maximum available rates which correspond to, respectively, 54 Mb/s for data frames and 24 Mb/s for acknowledgment frames, in agreement with the analysis performed in [56], the times necessary to transmit data and

acknowledgment frames, namely, t_{data} and t_{ack}, are given by

$$t_{data} = 20\ \mu s + \left\lceil \frac{38.75 + l}{BpS_d} \right\rceil \cdot 4\ \mu s + 6\ \mu s \tag{2.1}$$

$$t_{ack} = 20\ \mu s + \left\lceil \frac{16.75}{BpS_a} \right\rceil \cdot 4\ \mu s + 6\ \mu s \tag{2.2}$$

where l is the payload length, whereas BpS_d and BpS_a represent, respectively, the number of Bytes-per-Symbol for data frames (27 at the selected rate) and for acknowledgment frames (12 at the selected rate). In both the equations, the initial 20 μs term represents the time necessary for the transmission of both the physical preamble and headers; the terms $38.75 + l$ and 16.75 account for the transmission of the MPDUs of, respectively, data and acknowledgment frames, where the data frame contains also the LLC header. Finally, 4 μs is the symbol transmission time and 6 μs is a ramp-down guard period. The ceiling function is due to the OFDM modulation that needs an integer number of symbols (in practice, padding bits are used).

Considering a for example a payload of 64 bytes, from the above analysis we obtain: $t_{data} = 42$ μs and $t_{ack} = 34$ μs.

2.3 Wireless extensions of Ethernet Powerlink

The wireless extensions we analyzed refer to a specific network configuration which is one of the most deployed at the device level of industrial communication systems. We consider a one-level EPL network configuration comprising two segments, one wired, the other wireless. The EPL MN and a set of CNs are wired connected to the same Ethernet switch. Some additional controlled nodes, which constitute the EPL wireless extension, are located on the wireless segment and they are referred as Wireless Controlled Nodes (WCNs).

The interconnection between segments of a hybrid network is achieved by means of suitable devices, namely Intermediate Systems, that may operate, in principle, at almost all layers of the ISO/OSI reference model. However, according to [72] when dealing with industrial networks, the extensions take place almost exclusively either at the data link layer or at the application layer[2].

[2] As an interesting exception, in [50] the authors propose the interconnection between Profibus [16] and Radio Fieldbus [9] implemented by means of repeaters (and hence placed at the physical layer). This has been possible since both the networks use the same data link layer protocol which is, however, rather unusual when interconnecting such a different types of network.

The features of the EPL and IEEE 802.11 networks allow for the straightforward implementation of two different types of extension, characterized by the employment of an Intermediate System working either at the data link layer or at the application layer. At the data link layer, the Intermediate System is usually referred as *bridge*, whereas, at the application layer, it is known as *gateway*.

2.3.1 Extension at the data link layer

The wireless extension of EPL implemented at the data link layer is shown in Fig. 2.1. In this case the connection of a WLAN to a wired IEEE 802 LAN, as specified by the IEEE 802.11 standard, is achieved by means of a logical component known as *portal*. However, as already mentioned, in practical implementations an Intermediate System realizing the interconnection at the data link layer is commonly referred as bridge. The operation of bridges for lans (either wired or wireless) adopting the IEEE 802 family of MAC protocols is defined by the IEEE 802.1D MAC bridges standard [26].

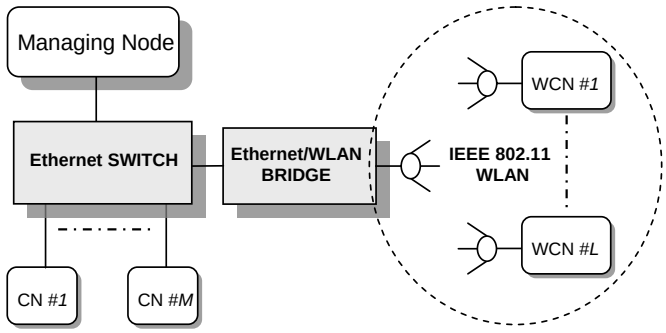


Figure 2.1: Extension at the data link layer.

The wireless extension of EPL implemented at the data link layer allows for the direct inclusion of the WCNs in the EPL cycle. Indeed, a *PReq* frame originated by the MN toward a WCN flows across the bridge transparently to the EPL protocol. The addressed WCN responds with a *PRes* frame which reaches the MN through the reverse path. However, as it will be detailed in the next section, the polling of a WCN requires a greater amount of time than that necessary for the same operation on a wired CN. Moreover, such a query is influenced by the randomness deriving from both the backoff procedure and the possible transmission errors that may occur on the wireless medium. This has a direct consequence on the timeout used by the MN for polling WCNs

which will have necessarily to be greater than that used for the wired CNs.

2.3.2 Extension at the application layer

The wireless extension of EPL implemented at the application layer is realized by means of a device known as gateway. The gateway, as shown in Fig. 2.2, is a rather complex device equipped with two interfaces necessary to communicate on the two different segments and it may be physically located either at the MN or at one of the wired CNs. Looking at the commercially available products, a set of components suitable to realize a gateway is, for example, the X20 CPU family by B&R [57]. Each CPU of the family is equipped with two Ethernet interfaces, one of these compliant with the EPL protocol (both the options, MN and CN, are available). Thus, a gateway could be practically implemented using an IEEE 802.11 access point connected to the second Ethernet port of the X20 CPU.

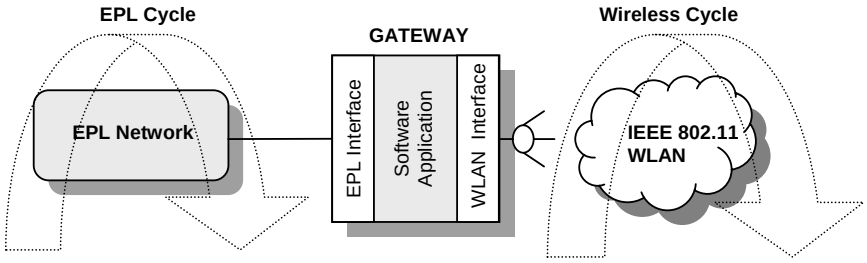


Figure 2.2: Extension at the application layer.

In this type of extension, WCNs are not directly included on the EPL cycle but, instead, they are periodically polled by a purposely defined application running on the gateway which communicates with the EPL network. Such an application may be either based on a polling procedure or driven by specific requests of data transmission to/from the WCNs. In the former solution, WCNs have to be periodically polled by the gateway, even if there are no data to transfer to/from the wireless segment. Alternatively, WCNs may be queried only when data have to be transmitted to/from one or more of them (event-driven procedure). The choice between the two alternatives is strictly application related and, as such, outside our scope. The analysis we propose, however, is based on the polling procedure. Thus, as described in Fig. 2.2, two different cycles are running on the hybrid network: the EPL cycle, handled by the MN that involves only wired CNs, and the wireless cycle, handled by

the gateway that polls the WCNs. Both the cycles run asynchronously. The way in which the data to/from WCNs are exchanged depends on the actual location of the gateway.

In detail, if the MN acts as gateway itself, then the data relevant to the WCNs are already present on the MN and they have just to be exchanged with the EPL protocol by means of a software application. Such a technique resembles that described in [51] where a protocol converter between a fieldbus and the IEEE 802.11 WLAN is described. Conversely, if the gateway is located on a wired CN, then the actual exchange of data with the MN takes place via the Ethernet switch during the EPL cycle. A gateway behaving in this way is often referred as *proxy*, since it practically represents all the WCNs as they were a single node.

2.4 Characterization of the wireless channel

Transmissions over a wireless channel are typically affected by high bit error rates due to the intrinsically error-prone nature of the transmission medium. According to [69], the behavior of a wireless channel may be negatively influenced, in particular, by two phenomena:

- *Interference*
  Wireless systems transmitting in the same band may interfere with each other causing severe drawbacks to the communication performance. Such a problem is particularly evident in the ISM band, where different wireless technologies (e.g. IEEE 802.11, IEEE 802.15.4 and Bluetooth) are employed without specific limitations. The interference effects as well as their mitigation, have been generally addressed in several papers, as reported in [69] and the references therein. More specific analysis on interference concerned with industrial wireless communication systems are provided in [38], [65] and [32]. In our analysis we considered the interference generated by another IEEE 802.11 system, which is a very common situation; we are conscious, however, that several other different interfering scenarios could have been considered.

- *Fading*
  Fading is basically the degradation of the received signal which may result in incorrect/missed reception of frames by the destination station(s). An interesting analysis of fading in industrial wireless communication sys-

tems is provided in [63]. Here, a basic distinction between large-scale and temporal fading is made and a set of meaningful measurements is given for both the cases. Roughly speaking, large-scale fading accounts for the fluctuation of the received power due to the fast time-varying mutual positions between transmitter and receiver, whereas temporal fading is defined as the variability of the received power at a fixed location which is mainly due to the movement of persons and/or machinery and to the addition/removal of obstacles to the signal propagation (this latter phenomenon is referred as *shadowing* in [69]).

From the characteristics of the wireless extensions analyzed in this work we may assume that the effect of large-scale fading is quite limited and hence it can be neglected. Indeed, the limited geographic area of the extensions as well as the knowledge of the site topography allow (at least in principle) to design a wireless system capable of ensuring that each station will receive an adequate power signal at every reachable location. Conversely, the unpredictable aspects of the temporal fading mentioned above (movement of persons, machinery, shadowing in general) are of relevant importance in our context, resulting in the predominance of such a kind of phenomenon.

The measurements carried out in [63] show that the temporal fading behavior is well approximated by a Rice distribution [58]. In practice, the received power comprises both a time-invariant component and an additional random component which accounts for the variability of the environment. The Ricean K-factor represents, basically, the ratio between these two components: stable environments exhibit high values of K-factors, whereas lower values are typical of environments characterized by a considerable level of movement and hence fluctuations of the received power. These results are in good accordance with a two-state (Gilbert-Elliot) channel [47], as actually already experimentally observed in [71] and [34], since the time-invariant component of the received power is able to maintain the channel in the good state, whereas the negative peaks of the random component force the channel in the bad state. Thus, in our analysis we referred to such a channel model.

2.5 Performance Evaluation

The performance analysis provided in this section refers to the one-level configuration discussed in section 2.3. Since EPL frames are encapsulated and transmitted into standard Ethernet payloads, the minimum size of the frames exchanged on an EPL network is 64 bytes. According to the Ethernet standard, such a frame size allows for the transmission of payloads in the range between 1 and 38 bytes. Hence, considering that the EPL protocol uses the first three bytes of the Ethernet payload, it follows that minimum size Ethernet frames may be employed by EPL (specifically by *PReq* /*PRes*) to transmit up to 35 bytes of data. In our analysis, we supposed that only minimum size Ethernet frames were exchanged between MN and controlled nodes during the isochronous period, being confident of covering a wide range of applications, since at the device level of industrial communication systems only limited amounts of data are typically exchanged between controllers and sensors/actuators. Moreover, we supposed the employed switches and bridges to be store-and-forward devices. The Orthogonal Frequency Division Multiplexing (OFDM) with 64-Quadrature Amplitude Modulation (64-QAM) at a transmission rate of 54 Mb/s was chosen for the physical layer of the IEEE 802.11 wlan.

The performance evaluation of both the extensions has been carried out by means of a theoretical analysis as well as numerical simulations (using Opnet). For the simulations, we employed the standard models provided by Opnet for switch and bridge devices, while for MN and CN we employed the purposely developed models described in section 1.3. Specific models have been purposely developed also for the WCNs and the gateway. In particular, the model of WCN has been realized implementing the EPL protocol on top of the MAC sublayer of a standard IEEE 802.11 station. Finally, the gateway has been realized as a CN equipped with two interfaces connected on the one side to the EPL network (and hence to the MN), and on the other side to the WCNs via a bridge. Further specific details about the gateway model will be provided later.

2.6 Bridge-based extension

For the bridge-based extension we refer, as a performance index, to the duration of the isochronous period T_I, since it represents a fundamental parameter,

that measures the precision of the communication system. Indeed, in an ideal scenario, T_I is constant and all the CNs are polled with a fixed polling frequency. However, since during the polling of a CN the issuing of both the *PReq* and *PRes* frames are not necessarily periodic (for example, the sending of *PRes* could be delayed or even timed out), the following CNs may be polled with a different lower frequency. Such a phenomenon, clearly, is much more likely when dealing with wireless CNs. The duration of the isochronous period is given by

$$T_I = t_{st} + T_{WD} + T_{WL} \tag{2.3}$$

where t_{st} is the time necessary to transmit the SoC frame, T_{WD} is the time necessary to poll the wired CNs and T_{WL} is the time necessary to poll the the WCNs. T_{WD} can be expressed as

$$T_{WD} = m \cdot (D_{CN} + D_{MN}) + m \cdot 4t_f \tag{2.4}$$

where m is the number of wired CNs, D_{CN} and D_{MN} are the delay introduced by the CNs and the MN respectively, and the term $m \cdot 4t_f$ accounts for the transmission time of the frames necessary to poll a CN. Indeed, for each CN, 4 frames have to be transmitted, and t_f is the time employed to send a single frame (as discussed above, it is the time necessary to transmit a minimum size Ethernet frame).

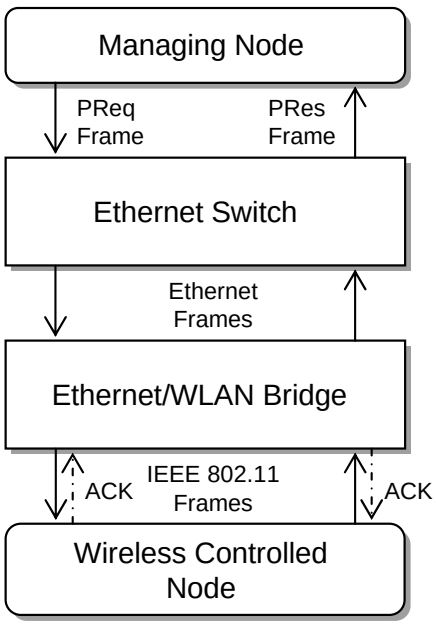


Figure 2.3: Polling sequence of a WCN.

In order to evaluate T_{WL}, Fig. 2.3 shows the sequence of operations involved in polling a single WCN. As can be seen, the *PReq* frame originated by the MN arrives at the bridge which extracts the payload and encapsulates it in an IEEE 802.11 frame which is then forwarded to the WCN. Here a *PRes* frame is generated and sent back to the MN. Thus, T_{WL} is given by

$$T_{WL} = t_b + T_{PWL} \tag{2.5}$$

where t_b accounts for the EPL frames transmitted (on the wired segment) between MN and bridge and can be obtained from Eq. (2.4) simply replacing m with the number of WCNs, l. The term T_{PWL} accounts for the IEEE 802.11 frames exchanged between the bridge and the WCNs to execute the polling procedure. Hence, T_{PWL} is given by the sum of the times requested by all the frames that are exchanged on the wireless segment. It can be derived from the service times of the IEEE 802.11 frames, as evaluated in [73]:

$$T_{PWL} = \sum_{i=1}^{l} \left[t_S(i) + \sum_{j=1}^{N_C(i)} T_C(i,j) + TO(i) + BO(i) + R(i) + D(i) \right] \tag{2.6}$$

where $t_S(i)$ denotes the time actually requested by the transmission of the frames necessary to poll the i-th WCN (two data frames, two acknowledgment frames, one DIFS and one SIFS). The term $\sum_{j=1}^{N_c(i)} T_C(i,j)$ accounts for the time taken by collisions: in particular, $N_C(i)$ is the number of collisions experienced in querying the i-th WCN and $T_C(i,j)$ is the time lost in the j-th collision. $TO(i)$ accounts for the timeouts due to failed transmissions caused by channel errors. It is worth noticing that both the terms $\sum_{j=1}^{N_c(i)} T_C(i,j)$ and $TO(i)$ have been introduced in Eq.(2.6) in order to better clarify the formal description of T_{PWL} but, in practice, they can not be distinguished. Indeed, according to [19], a transmission error caused either by collisions or whatever type of failure is always detected in the same way, i.e. by the missed (or incorrect) reception of the acknowledgment frame. In [73], only the contribution of collisions is considered since the authors refer to an error free channel. $BO(i)$ is the delay introduced by the backoff procedure. $R(i)$ is the time possibly spent waiting for the conclusion of transmissions initiated by wireless systems different from the EPL extension (e.g. interfering networks) since, according to the DCF operation, when either the bridge or the i-th WCN senses the channel as busy, it waits for the end of the transmission and then it starts decrementing the backoff counter. Moreover, it has to be observed that a transmission by a different network may occur while either the bridge or the i-th WCN is actually

decrementing the backoff counter as well. In this case the backoff procedure is stopped and resumed at the end of the transmission, with the consequent prolongation of the service time. Finally, $D(i)$ is the delay introduced by the bridge when it polls the i-th WCN. As pointed out in [53] and [51], such a delay is mainly due to two distinct causes, namely, latency and queuing[3]. The latency is related to the intrinsic structure of the bridge and, basically, it represents the delay between the instant in which a frame enters the bridge and the instant in which the converted frame starts exiting the bridge, supposing that no other frames are currently waiting to be transmitted. Although a general value for the latency can not be specified, some extensive measurements on commercially available products carried out in [37], showed that it typically falls into the $5-10$ μs range. Consequently, we selected 10 μs as the latency of the bridge model used in the simulations. As a second cause of delay, a frame which arrives at the bridge may be queued if other frames are waiting to be transmitted. It is worth mentioning that queueing represents a random delay directly related to general characteristics such as, for example, network configurations, traffic profiles, etc. As such, queueing can not occur in traditional (wired) EPL networks since the protocol is based on a TDMA technique which allows for the ordered delivery of frames. However, as it will be shown, for hybrid networks it may happen that interfering frames on the wireless segment have to be handled by the bridge, with the possible consequent introduction of such a type of delay.

Clearly, T_{PWL} is a random variable which is responsible for the jitter of the isochronous period, T_I. Nonetheless, it is interesting to observe that, if within the EPL cycle the MN polls all the wired CNs at first and subsequently all the WCNs, then since the EPL cycle is started with exact periodicity, the jitter does not influence the wired CNs, provided that the duration of the isochronous period does not exceed the prefixed EPL cycle.Moreover, as further critical drawback affecting the correct execution of the EPL cycle, the MN could experience timeouts in querying the WCNs due to the randomness of the polling time. Such a situation reveals dangerous since several consecutive timeouts lead to the exclusion of the WCN from the isochronous period. This problem, however, may be solved (at least in principle) setting suitable timeout values after a careful evaluation of the time requested to poll the WCNs.

[3]Actually, in [51] the authors refer to switch devices. The analysis, however, is valid for bridges as well.

It can be concluded that the extension at the data link layer may be successfully implemented at the expense of longer EPL cycles and accepting the unavoidable presence of jitter on the wireless segment. Clearly, this may represent a problem in some contexts like for example motion control applications, where strict determinism may be required. Nevertheless in several other scenarios, typically those characterized by soft and weak hard real-time requirements [31], the above limitations are well tolerated and hence wireless extensions of EPL at the data link layer may be profitably employed. The remainder of this subsection provides some examples which allow to evaluate the effects of interference and fading on the behavior of the isochronous period.

2.6.1 Effects of interference

Assuming that only interference is present on the wireless channel, we may neglect the term $TO(i)$ in Eq. (2.6).

However, the analytical evaluation of T_I is still rather complex since it relies on the calculation of the polling times of the WCNs which, clearly, depend on the statistical description of the interference. An example of calculation is actually provided in [73] for saturated networks which allows to determine, for a single station, the probabilities of both collision and transmission from the analysis carried out in [33]. Unfortunately, this is an unrealistic assumption for industrial device level communication systems where, typically, the network load is very low. Thus, in order to get some useful insights on the effects of the interference, we resorted to analyze some specific configurations.

As a first case we consider an interfering network in which there is only one transmitting node. Assuming that the load of the interfering network is low, we may neglect the delays introduced by collisions in Eq. (2.6), as experimentally verified in [38]. Thus, intuitively, if an EPL station (either the bridge or a WCN) tries to transmit a frame and the network is busy, it waits for the end of the current transmission and elaborates the backoff time. When this latter elapses, the station transmits. The overall result is that the issuing of the EPL frames is occasionally delayed by the transmission of the interfering ones. Thus, in principle, the statistical description of T_{PWL}, may be derived from the traffic profile of the interfering network. For example, if the interfering station generates packets of fixed duration d according to a Poisson counting process with intensity λ packets/s, then the moment generating function of T_{PWL}, $G_{T_{PWL}}(z) = E[z^{T_{PW}}]$ can be computed. However, to show the analytical

expression of $G_{T_{PWL}}(z)$ is beyond the scope of this work. We limit to report the mean value of T_{PWL} since it allows to make some interesting considerations. It results:

$$E[T_{PWL}] = l \cdot \left[T_S + t_{slot}CW_{min} + D_m + \lambda t_{slot}d + (1 - e^{-\lambda})d\right] \qquad (2.7)$$

Eq. (2.7) has been derived under the assumption that CW_{min} is the only contention window used since, all transmissions are supposed to be successful.

Considering that the same number of input/output bytes is exchanged between MN and each WCN, we have $T_S(i) = T_S$. The term $t_{slot}CW_{min}$ is the mean time required by the backoff procedures. D_m is the mean delay introduced by the bridge. The last two addends in Eq. (2.7) account for the delays introduced by the transmission of interfering frames. In particular, the term $\lambda t_{slot}d$ is relevant to transmissions which occur while a station belonging to the wireless segment is decrementing its backoff counter, whereas the term $(1 - e^{-\lambda})d$ is the mean delay experienced by a station when it accesses the wireless medium for the first time and it finds it busy.

We simulated the behavior of a network configuration with 8 wired CNs and a variable number of WCNs ranging from 1 to 3 in presence of an interfering network generating a traffic profile as described above. IEEE 802.11 stations transmitting on the same band have been employed for simulating the interfering network. In detail, the interfering network comprised two stations, one acting as sender, the other as receiver. General purpose TCP/IP PDUs resulting in 1000 bytes payloads IEEE 802.11 frames were cyclically transmitted (the transmission time of one of these packets results 226 μs) with period 2.2 ms, generating a 20% average network load.

The most relevant EPL parameters used in the simulation are summarized in Table 2.2. We decided to set $\tau_{EPL} = 5\ ms$ (ensuring an adequate safety margin); moreover, the EPL timeout for the WCNs has been set to 1.5 ms, which is the sum of the maximum theoretical time necessary to poll a single WCN and the duration of the interfering frame plus a safety margin.

Fig. 2.4 shows the behavior of the isochronous period in terms of 90% confidence interval obtained from simulation along with the results deriving from Eq. (2.7). In particular, Fig. 2.4 (a) refers to an ideal channel (no interference), whereas Fig. 2.4 (b) is relevant to the behavior of T_I for a channel with an average interference load of 20%. Both the wireless extension and the interfering network were using non-prioritized frames.

As can be seen, the mean calculated values are in good accordance with

Table 2.2: Simulation parameters

Parameter	Description	Value
m	Number of CNs	8
l	Number of WNs	1 to 3
b	payloads bytes exchanged	16
t_{st}	time to transmit the SoC frame	45 μs
D_{CN}	CN response delay	8 μs
D_{MN}	MN delay	1 μs
T_{WD}	Polling time of wired CNs	235.84 μs
τ_{EPL}	EPL cycle time	5 ms
τ_{to}	EPL timeout	1.5 ms

those obtained by the simulations. There is, however, a certain discrepancy which may be noticed in presence of interference. This is due to a queuing delay introduced by the bridge. In fact, since the interfering network operates in *infrastructure mode*, each frame originating from that network is firstly received by the bridge and then relayed to the destination station. Thus, an interfering frame which has to be transmitted by the bridge may delay the transmission of the frames exchanged on the wireless extension. Although actually introduced by the queuing on the bridge, such a delay is specifically dependent by the traffic profile of the interfering network. Moreover, it is even more remarkable when the number of WCNs increases due to the consequent increment of the overall network load.

The performance of the hybrid network may be improved using prioritized frames in the wireless segment, since in this case the contention window is reduced, resulting in lower values of the isochronous period. We simulated such a scenario using frames belonging to AC[3] (Voice) for the wireless segment (in this case the contention window was $(CW_{min}+1)/4-1$), and frames belonging to AC[0] (Background) for the interfering traffic.

The results, reported in Fig. 2.5, show the considerable reduction of the mean values of the isochronous period. However, a difference with the mean value calculated from Eq. (2.7) as well as a residual, yet remarkable, jitter may still be observed. Besides the backoff procedures (which are responsible for the jitter, even if with more limited effects), the discrepancy is due, again, to the queuing delay introduced by the transmission of interfering frames by the

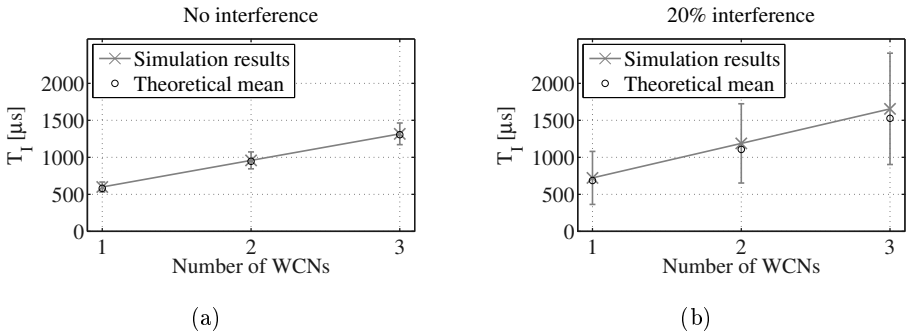
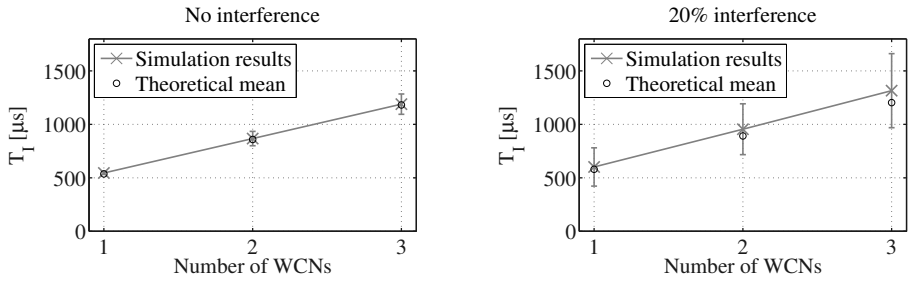


Figure 2.4: Isochronous period for non-prioritized frames: ideal channel (a); 20% average interference load (b).

bridge; in particular, we have verified that the problem is caused by interfering frames that are transmitted immediately before the MN begins to poll the WCNs. Indeed, these frames have low priority, but a considerable transmission time (226 μs). Thus, if the polling of the WCNs had to start when one of the frames is being transmitted, it would have necessarily to wait for the conclusion of the current transmission, although the priority of the frames employed by the polling is greater.

The queueing delay has not been explicitly introduced in Eq. (2.7) since it actually appears only when the network operates in infrastructure mode, whereas such a delay is not present in ad-hoc mode. The simulations carried out allowed to provide a precise characterization of the queuing delay showing, in particular, that it may be reduced using higher priority frames for the wireless segment. Consequently, the results obtained reveal helpful in order to correctly choose the design parameters of the hybrid network. However, configuring the wireless extension of EPL as an ad-hoc network, with a unique Service Set Identifier (SSID), prevents from the occurrence of the queuing delay since the frames generated by interfering stations are not transmitted to the bridge. Unfortunately, the simulation tool we employed did not allow to test such a latter type of operation since it was not able to configure the wireless extension of EPL with a SSID different from that of the interfering network. Nevertheless, such a configuration can be practically implemented using commercially available components, as shown in [38].

The statistics of the isochronous period are summarized in both Table 2.3 and Table 2.4 for the two cases (non-prioritized and prioritized frames) respectively considered.



(a) (b)

Figure 2.5: Isochronous period for prioritized frames: ideal channel (a); 20% average interference load (b).

Table 2.3: Statistics of the isochronous period (non-prioritized frames)

N. of WCNs	Mean (Ideal)	St. Dev. (Ideal)	Mean (20% Intf.)	St. Dev. (20% Intf.)
1	599 μs	41.6 μs	722 μs	263.7 μs
2	959 μs	69.1 μs	1189 μs	431.1 μs
3	1318 μs	88.5 μs	1654 μs	557.4 μs

Table 2.4: Statistics of the isochronous Period (Prioritized Frames)

N. of WCNs	Mean (Ideal)	St. Dev. (Ideal)	Mean (20% Intf.)	St. Dev. (20% Intf.)
1	545 μs	10 μs	602 μs	133.6 μs
2	867 μs	40.5 μs	954 μs	197.3 μs
3	1189 μs	56.8 μs	1316 μs	253.8 μs

In a further set of tests we considered the presence of more interfering stations. From a general point of view, if non-prioritized frames are used, as long as the network load remains low, it can be expected that the isochronous period occasionally assumes greater values due to the increased probability of collisions as well as contentions on the network. This is actually the result we obtained simulating a scenario with three interfering stations which generated a network load (20%) equal to that of the previous analysis (such an operational condition was achieved maintaining the same size of the interfering frames and correspondingly increasing the period of their cyclic transmission to 6.6 ms for each interfering station).

A more interesting result, however, has been observed when prioritized frames were used: in this case, the mean of the isochronous period assumed practically the same values shown in Table 2.4, whereas the standard deviation surprisingly showed a slight decrement when the number of WCNs was increased. This behavior may be explained considering that, as observed in the previous experiment, when prioritized frames are employed, then the interference only affects the time at which the polling of the WCNs starts. Thus, it may happen that more interfering stations which contemporaneously try to transmit immediately before the polling begins, could experience collisions and/or contentions, which results in higher access times, paradoxically leaving the channel free for the beginning of the polling.

2.6.2 Effects of Fading

Considering only the presence of fading, the access to the transmission medium of the stations belonging to the wireless segment is ordered, and a station (either the bridge or a WCN) that has a frame to transmit always finds the channel free. This means that, in Eq. (2.6), both the terms $R(i)$ and $\sum_{j=1}^{N_c(i)} T_C(i,j)$ may be neglected. However, packets may be lost and/or corrupted and hence they may need to be retransmitted.

Let us assume that the transmission from the bridge to the i-th WCN succeeds after $N_q(i)$ attempts, whereas the transmission on the reverse path requires $N_s(i)$ attempts (both $N_q(i)$ and $N_s(i)$ are independent random variables), then in Eq. (2.6) we have[4]:

$$BO(i) = \left\{ \sum_{j=1}^{N_q(i)} U\left[0, 2^{(j-1)} CW_{min}\right] + \sum_{j=1}^{N_s(i)} U\left[0, 2^{(j-1)} CW_{min}\right] \right\} t_{slot} \quad (2.8)$$

[4]the notation $U[x,y]$ represents a discrete uniform random variable between x and y

and:

$$TO(i) = [N_q(i) + N_s(i) - 2]\, t_{out} \tag{2.9}$$

where t_{out} is the ACK timeout. Both $N_q(i)$ and $N_s(i)$ are upper bounded by the retry limit of the IEEE 802.11 frames. These values have to be harmonized with the EPL timeout. In fact, if during the query of a WCN, the MN does not receive the *PReq* frame within the EPL timeout, then it moves on the next station. However, if the retry limit on the node that issued the relevant IEEE 802.11 frame (either the bridge or a WCN) is not exceeded, then that node will continue to transmit, possibly delaying the frames relevant to the query of the next station. Thus, the retry limit has to be set in such a way that, when it is reached, the EPL timeout has not expired yet. For example, the maximum time requested by three unsuccessful polling attempts of a WCN, obtained from Eqs. (2.6), (2.8) and (2.9) results 1242.48 μs (using non-prioritized frames and selecting, for each attempt, the maximum backoff time) hence, if the retry limit is set to 3, then the EPL timeout of each WCN has to be greater than such a value.

In order to verify these considerations, we simulated the behavior of the isochronous period for a network with 3 WCNs assuming a Gilbert-Elliot channel model for the wireless segment. In particular, according to [55], we set the mean packet error rate (PER) to 10^{-1}, whereas the average length of the error bursts on the channel was set to 1.7 ms, a value comparable with the duration of the isochronous period which allows to evaluate the effects of fading within an EPL cycle (clearly, depending on the operational scenario, different error burst lengths could be employed). Moreover, prioritized frames were used and the retry limit was set to 3, whereas the EPL time–out was unchanged from the previous experiments (1.5 ms). The outcomes of the simulation are shown in Fig. 2.6. The mean value of the isochronous period results 1369 μs and the standard deviation 720 μs. An immediate comparison with the ideal values shown in Table 2.4 (1189 μs and 56.8 μs respectively) reveals that the fading introduces a remarkable increase of the jitter. The most evident peaks in Fig. 2.6 confirm the expiration of the EPL timeouts which are clearly due to the occurrences of the channel bad states during the polling of the WCNs. In these situations the reduction of the retry limit revealed to be effective since no transmission attempts by a station were observed after the expiration of the EPL timeout. Moreover, given the short duration of the error bursts, the maximum number of consecutive EPL timeouts (set to 5 per station) was never

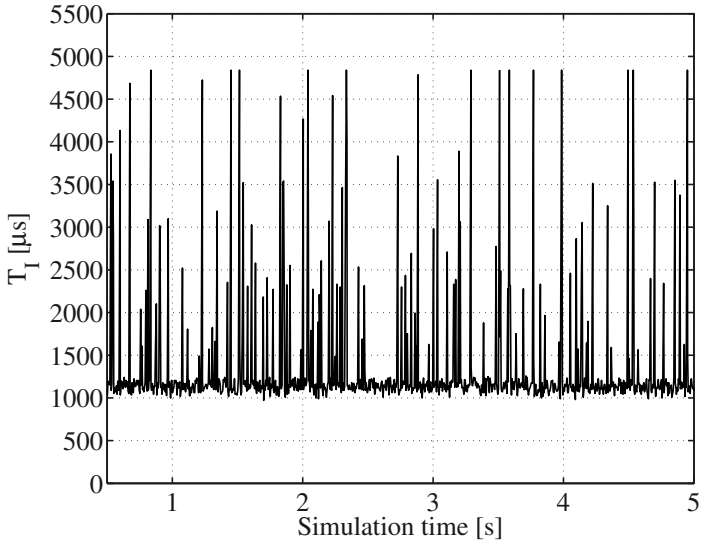


Figure 2.6: Isochronous period with prioritized frames in presence of fading.

exceeded.

2.6.3 Combined Effects of both Interference and Fading

As a last experiment for the extension at the data link layer, we considered the case in which both interference and fading are contemporaneously present on the wireless segment. We referred to a network configuration comprising 3 WCNs exchanging prioritized frames with the bridge. The wireless channel was characterized by 20% of interference load generated by a single station and the aforementioned Gilbert-Elliot channel was used to modeling the presence of fading. The statistics of the isochronous period obtained from simulation are shown in the last raw of Table 2.5. In order to make an effective comparison with the previous experiments, the table also reports: the values of mean and standard deviation for an ideal channel, the values relevant to the separated effects of interference and fading and the coefficient of variation (CoV), defined as the ratio between standard deviation and mean.

It may be observed that the jitter on the isochronous period is mainly due to the fading, as reflected by the high value of the standard deviation and by its limited increase when both interference and fading are present. On the other hand, the mean value of the isochronous period is influenced practically in an independent way by the concurrent effect of these two causes of error (indeed, the increase of the mean value due to the sole effect of the interference is 127 μs, that due to the fading is 180 μs, whereas the increase

due to the combined effect of interference and fading is 327 μs). In practice, the effects of both interference and fading are superimposed. This may be explained considering that, as previously discussed, the interference causes a queueing delay on the bridge. Hence, if an interfering frame issued by the bridge is corrupted by fading, it will go through some further transmission attempts, until the channel reaches the good state; in the meanwhile, an EPL frame, possibly queued on the bridge has to wait for the correct delivery of the interfering frame before being transmitted.

Table 2.5: Complete statistics of the isochronous period

Channel	Mean	Std. Dev.	CoV
Ideal	1189 μs	56.8 μs	0.047 μs
20% Intf.	1316 μs	253.8 μs	0.19 μs
Fading	1369 μs	720 μs	0.52 μs
Intf.+Fading	1516 μs	756.4 μs	0.49 μs

2.7 Gateway-based extension

For this type of extension, with reference to the two alternatives mentioned in subsection 2.3.2, we decided of implementing the gateway on a wired CN. In particular, as described in Fig. 2.7, the gateway model used in the simulations has been realized as a composite device comprising a CN equipped with a two-port integrated switch (the model of such a device has been obtained modifying that of a CN) and a bridge (whose model, in this case, has been taken directly from the Opnet library). The first Ethernet port of the CN was used to communicate with the MN, whereas, the second one was connected to the bridge in order to communicate with the WCNs.

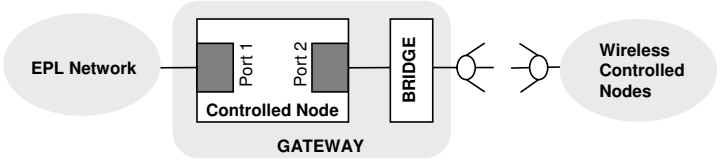


Figure 2.7: Gateway model.

The actual communication between gateway and MN takes place during the EPL cycle since the gateway is handled as a CN. As a consequence, differ-

ently from the extension at the data link layer, the duration of the isochronous period does not represent an effective performance index in this case, since it is always constant. Thus, in order to evaluate the performance of the extension at the application layer we resorted to refer to the set of performance indicators defined by the IEC 61784 International Standard and specifically to the *"Delivery Time"* (DT), which is defined as: *"the time needed to convey an APDU (Application Protocol Data Unit) containing data (message payload) that has to be delivered in real-time from one node (source) to another node (destination)"*.

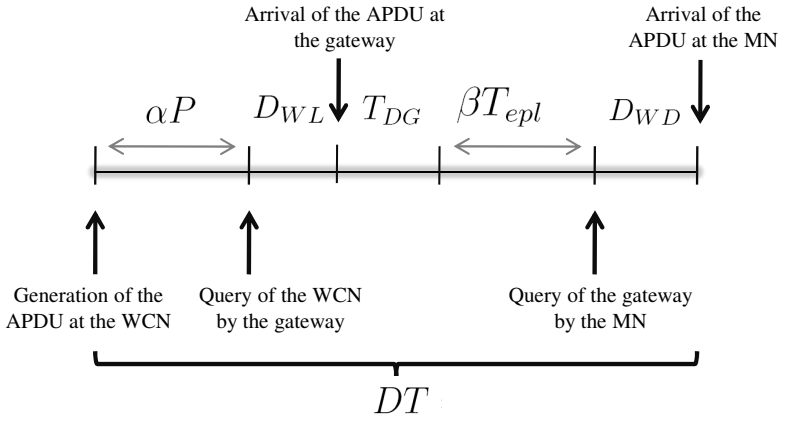


Figure 2.8: Calculation of the Delivery Time.

In particular, we evaluated the delivery time relevant to the sending of an APDU from a WCN to the MN. This transmission requires, basically, that the APDU is first sent by the WCN to the gateway and then from the gateway to the MN, as described by the time diagram shown in Fig. 2.8. Consequently, DT is expressed as:

$$DT = \alpha P + D_{WL} + T_{DG} + \beta T_{epl} + D_{WD} \tag{2.10}$$

where α and β are random variables uniformly distributed between 0 and 1, whereas P is the repetition period of the wireless polling cycle which implements the data exchange between gateway and WCNs. Thus, in Eq. (2.10), the term αP accounts for the latency of the APDU on the WCN before being acquired by the gateway; similarly, βT_{epl} is the time waited by the APDU on the gateway before being sent to the MN. D_{WL} is the time necessary to transmit the APDU from the WCN to the gateway: this implies the transmission of two IEEE 802.11 frames (a request frame from the gateway to the WCN and a response frame carrying the APDU from WCN to gateway) plus two Ethernet

frames (from the integrated switch of the CN in the gateway to the bridge and vice-versa). Due to the backoff procedures of the IEEE 802.11 frames, D_{WL} is a random variable which, if prioritized frames are used, has a mean value of 239 μs. T_{DG} accounts for the internal delay introduced by the gateway. Such a delay is mainly due to latencies, queueing and software execution times. In particular, the latencies and queuing may be caused by both the integrated switch and the bridge, whereas the software delays are strictly related to the specific implementation and may be variable. For example, in the simulation model, we have experienced that a *PReq* frame which arrives at the gateway when this latter is waiting for the response from a WCN causes the interruption of the wireless polling cycle, with its consequent temporary prolongation. Finally, D_{WD} is the time necessary to transmit the APDU from the gateway to the MN (this is achieved via the *PReq* /*PRes* frames and takes 10.24 μs).

As a final remark, it is worth observing that the actual duration of the wireless polling cycle is given by:

$$T_{PL} = T_{PW} + T_{CB} \tag{2.11}$$

where T_{PW} has the same meaning as in Eq. (2.6), whereas T_{CB} accounts for the time requested to transmit the Ethernet frames between the CN and the bridge, inside the gateway, necessary to implement the polling of the WCNs. Clearly, T_{PL} is a random variable and it represents the lower bound of the repetition period, P.

The EPL parameters used in the performance analysis carried out are the same of Table 2.2 with some minor differences: the number of WCNs, in this case was maintained constant to 3, the EPL cycle was set to $\tau_{EPL} = 1\ ms$ and the EPL timeout to $\tau_{to} = 100$ μs.

2.7.1 Effects of Interference

We simulated the behavior of the Delivery Time for different values of P (ranging from 10 ms down to T_{PL}) and considering an interference free channel as well as the presence of a 20% interfering load generated by a single station. It is worth mentioning that since T_{PL} is a random variable, when it is used as repetition period, differently from the other values of P, the WCNs are polled at a non-constant frequency.

Considering the number of frames necessary to poll the WCNs, the mean value of T_{PL} for an ideal channel is 747.7 μs . Prioritized frames were used in

the simulations, assigning Voice access category to the packets exchanged on the wireless extension, and Background to the interfering frames. The statistics of the delivery time obtained from the simulation is given in Table 2.6. The table actually includes also the data relevant to the presence of fading which will be discussed later on. Fig. 2.9 shows the behavior of the delivery time in term of 90% confidence interval along with the theoretical mean values which have been obtained from Eq. (2.10) neglecting T_{DG}.

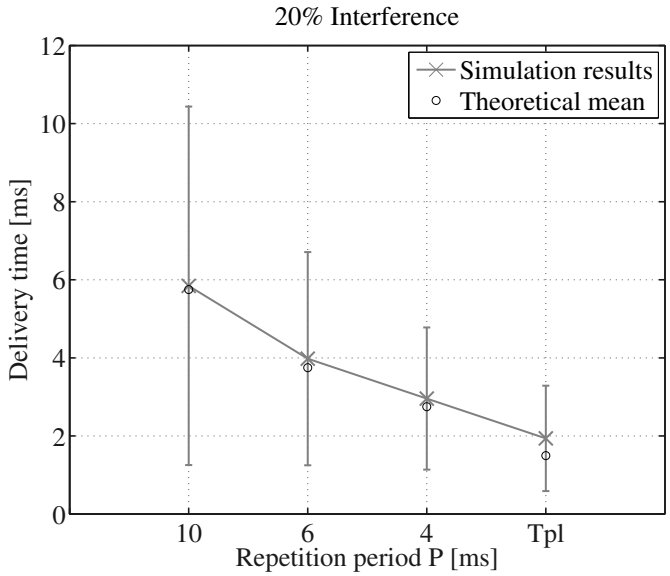


Figure 2.9: Delivery Time with 20% interference load.

Table 2.6: Statistics of the Delivery Time (prioritized frames)

P [ms]	Mean (Ideal)	St. Dev. (Ideal)	Mean (20% Intf.)	St. Dev. (20% Intf.)	Mean (Fading)	St. Dev. (Fading)
10	5.87 ms	2.91 ms	5.84 ms	2.95 ms	5.95 ms	2.76 ms
6	3.94 ms	1.75 ms	3.97 ms	1.77 ms	4.09 ms	1.72 ms
4	3.04 ms	1.14 ms	2.95 ms	1.18 ms	3.05 ms	1.16 ms
T_{PL}	1.48 ms	0.60 ms	1.94 ms	1.12 ms	1.68 ms	1.06 ms

The statistics presented in Table 2.6 allow to provide a characterization of the delay introduced by the gateway, T_{DG}. Considering, for example, the case $P = 10$ ms for an ideal channel, it results that the mean value of DT is 5.87 ms. Since the sum of the mean values of the terms in Eq. (2.10) except T_{DG} is 5.80 ms, the mean value of T_{DG} results 70 μs. As it may be noticed,

such a delay increases for lower values of P. This is understandable because the higher execution frequency of the polling cycles on the WCNs increases the probability that such cycles are interrupted (and hence prolonged) by the arrival of a *PReq* frame at the gateway.

Furthermore, the effects of the interference reveal negligible for high values of the repetition period P. This is well explained considering that, in such situations, interfering frames are mainly delivered in the interval between the end of a polling cycle and the beginning of the next one. Hence, the interference may only influence the beginning instant of a polling cycle, similarly to what happened for the extension at the data link layer. This behavior has, paradoxically, a slight beneficial effect on the Delivery Time. In fact, it may happen that at the beginning instant of the wireless polling cycle, the channel is occupied by an interfering transmission. In this situation, the gateway has necessarily to wait for the end of such a transmission before start polling the WCNs. Thus, an APDU which arrives in this period of time, will be transmitted in the current polling cycle instead that in the next one. This is the reason for which some mean values of the delivery time in the presence of interference are slightly lower than the correspondent ones for an ideal channel. On the other hand, for values of P comparable or equal to its lower bound (T_{PL}), interfering frames are transmitted within the polling of the WCNs, causing an increase of both mean and standard deviation values.

This analysis has been corroborated by a further simulation, carried out using non-prioritized frames, that as expected, showed values of the delivery time very similar to those of Table 2.6 for high repetition periods (since the effects of both backoff and interference were negligible) and worst values for repetition periods close to T_{PL}.

Finally, it is worth observing that the value of the repetition period P is often determined by the requirements of the specific applications since it is intrinsically related to the refresh time with which APDUs have to be received by the MN. Thus, we decided to execute a further simulation in order to assess the percentage of APDUs that have a delivery time that exceeds a specific threshold expressed as a function of P. The analysis has been carried out for the case of a channel with 20% interference and $P = 10$ ms. Results are shown in Fig. 2.10 where the x-axis reports the threshold, whereas the y-axis shows the percentage of values exceeding the threshold. It may be noticed that about 10% of the values are greater than P.

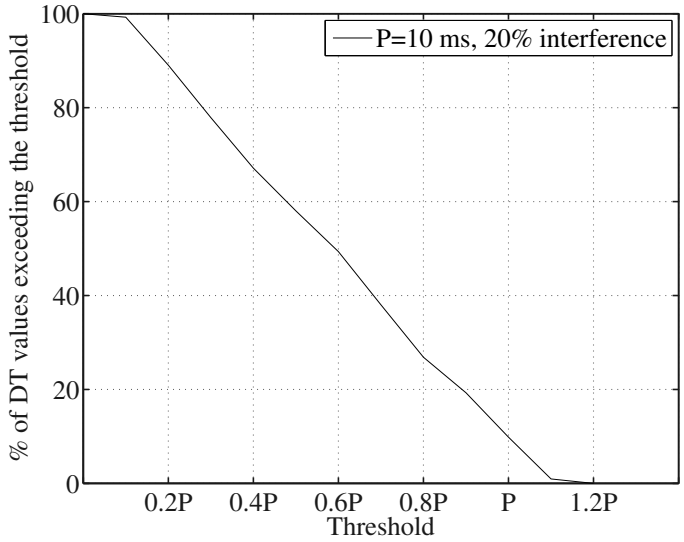


Figure 2.10: Percentage of DT values exceeding threshold.

2.7.2 Effects of Fading

The presence of fading compels temporarily the wireless channel in the bad state reflecting in possible failures (and, consequently, retransmissions) during the polling of the WCNs. We simulated such a scenario with the same channel model described in section 2.6.2 and using prioritized IEEE 802.11 frames in the wireless segment. The obtained behavior of the delivery time, in terms of 90% confidence interval, versus the repetition period is shown in Fig. 2.11, whereas its statistics are given in Table 2.6. As can be noticed, the behavior is very similar to that obtained in the case of presence of interference. Specifically, for high values of P, there are only slight differences with the results of the ideal channel model whereas, when $P = T_{PL}$, the negative effect of fading is well evident.

2.7.3 Combined Effects of both Interference and Fading

The combined effect of both interference and fading has been analyzed for the extension at the application layer as well. Among the possible values of the repetition period, we selected $P = T_{PL}$ which ensures the best performance but, on the other hand, it is the more sensitive to interference and fading.

The complete statistics of the delivery time is shown in Table 2.7 where we reported the values for an ideal channel, the values relevant to the separated effects of interference and fading and the coefficient of variation. It may be observed that, as for the results obtained for the extension at the data link

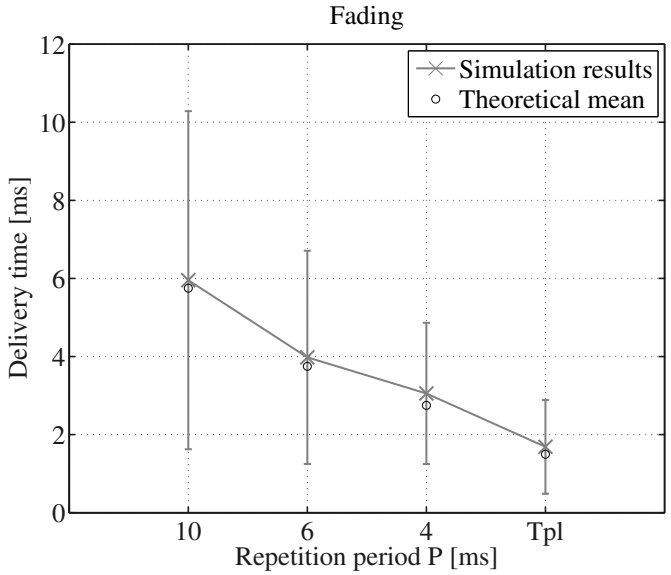


Figure 2.11: Delivery Time in presence of fading

Table 2.7: Complete Statistics of the Delivery Time

Channel	Mean	Std. Dev.	CoV
Ideal	1.48 ms	0.60 ms	0.40 ms
20% Intf.	1.94 ms	1.12 ms	0.57 ms
Fading	1.68 ms	1.06 ms	0.63 ms
Intf.+Fading	2.17 ms	1.78 ms	0.8 ms

layer, the contributions of both interference and fading to the increment of the mean value of DT are practically independent. Conversely, as a major difference, in this case the effects of the interference on the jitter are similar to those caused by the fading. This is explained considering that, since $P = T_{PL}$, the transmission of an interfering frame always prolongs the wireless polling cycle whereas, in the extension at the data link layer, interfering frames cause jitter only if they occur at the beginning of an EPL cycle.

2.8 Conclusions

In this chapter we addressed the extension of EPL networks based on the IEEE 802.11 wlan. Two distinct configurations have been considered based on, respectively, bridge and gateway devices. As a major difference between the extensions, the adoption of an Ethernet/wlan bridge allows for the direct

inclusion of WCNs in the EPL cycle, whereas this is not possible when a gateway is employed.

A performance evaluation, based on a theoretical analysis as well as numerical simulations, has been carried out taking into consideration the possible presence of interference and fading on the wireless segment. As far as the bridge based extension is concerned, the obtained results have shown how the performance of a traditional EPL network are influenced by the introduction of the wireless extension. In particular, the cycle times of the hybrid networks revealed greater than those achievable with the wired ones and a residual, at times non negligible, jitter affects the isochronous period even if prioritized frames are used. This is due to the intrinsic randomness of the IEEE 802.11 network as well as to the effects of interference and fading. These potentially critical aspects have been clearly pointed out and, at the same time, we clarified how they can be mitigated. Consequently, the performance analysis carried out allows to identify the range of applications for which such a type of wireless extension represents an interesting (and viable) opportunity.

Concerning gateway based extensions, the delivery time between a WCN and the MN has been considered as performance indicator. In this case, the wireless extension has no impact on the behavior of the EPL network, since the data exchange with the WCNs is handled independently by the gateway. Besides providing some performance figures under different operational conditions, the analysis focused on the characteristics of the gateway as well as on the impact of such a device on the overall performance of the hybrid network.

The very different nature of the two extensions makes impossible an effective comparison of the obtained results and, actually, this was not within our scope. Rather, the analysis was aimed at obtaining some useful insights concerning the implementation of both the types of extension as well as to evaluate their performance in operational scenarios typical of industrial environments.

In this work, we did not consider any of the PCF and HCCA techniques specified by IEEE 802.11 which allow for controlled channel access. This was motivated by the practical unavailability of commercial products providing such techniques. However, they would reveal helpful especially for the extension at the application layer since the implementation of the gateway is greatly simplified. Indeed, in this case, the wireless polling cycle executed by the gateway does not need to be realized, because it is implicitly made available by the controlled channel access technique which assigns specific slots to the WCNs.

Conversely, PCF and HCCA would not provide significant benefits if employed for the extension at the data link layer since the ordered access to the channel they introduce, is already implemented by the managing node.

Future developments of this work are obviously represented by the actual implementation of the proposed extensions followed by the execution of exhaustive experimental measurements, under environmental conditions reflecting those of typical industrial applications. In such a direction we actually implemented the extension at the data link layer by using commercially available access point devices and carried out several experiments whose results will be presented in chapter 3. On the other hand, the implementation of a gateway based extension likely implies a more substantial effort, since it requires the development of the wireless cycle (necessary to poll all the WCNs) as well as the realization of the software application which handles the data transfer between the two gateway interfaces.

3 Performance indicators for wireless ICNs

3.1 Motivation

As discussed in chapter 2, wireless technologies are particularly appealing for a large number of industrial communication applications. As an example, wireless sensors networks may be employed for factory environments/plants monitoring, as well as general purpose wireless networks (IEEE 802.11 wlan, IEEE 802.15.4 wpan [25], etc.) may be employed for wireless extensions of already deployed wired industrial communication systems (hybrid wired/wireless networks).

The availability of several different wireless technologies, along with the wide range of their possible application in the industrial communication context, suggest the need for the definition of a set of performance indicators capable of characterizing the behavior of a wireless technology as well as of its specific application. To this regard, an interesting initiative, limited however to wired networks, is represented by the performance indicators defined by the IEC 61784 International Standard for RTE networks that have been defined in order to *"specify capabilities of an RTE device and an RTE communication network as well as to specify requirements of an application"*. Clearly not all industrial communication systems employing wireless technologies can be effectively evaluated by the same set of performance indicators. Indeed, for example a typical device level communication application (e.g. motion control) is different from a wireless sensor network application like environmental/plant monitoring and consequently the performance indicators adopted in one case would not fit very well in the other.

In this chapter we refer to the wireless extension of Ethernet Powerlink based on the IEEE 802.11 wlan introduced in chapter 2 and we describe a

possible practical implementation of the extension at the data link layer which makes use of EPL standard devices connected to the IEEE 802.11 wlan by means of commercially-available, general purpose access points. In order to experimentally evaluate the performance provided by such a communication system, we define a set of performance indicators suitable for polling-based wireless industrial communication networks and, specifically for our case study. The set of performance indicators we use is particularly effective for evaluating the performance of industrial communication systems employing wireless technologies. Indeed, the operation of several industrial communication systems (wired as well as wireless) is based on a polling operation carried out by a master on a number of slaves, that is periodically executed. It is clearly the case of EPL but also of many other ones (wired as well as wireless) (e.g. ControlNet [40], Profibus DP [18], the Point Coordination Function access specified by IEEE 802.11 wlan, etc.).

In order to compare theoretical and practical results, we provide examples of both analytical and experimental computation of the defined performance indicators for our case study. Moreover we give some useful insights on the enhancements of performance for the considered communication system.

In detail, we organize the chapter as follows. Section 3.2 defines the set of performance indicators we will use through the chapter. Section 3.3 discuss our case study, that is the wireless extension of EPL based on the IEEE 802.11, and, in particular, its practical implementation. Finally, section 3.4 provides an example of theoretical computation of one of the performance indicators, while section 3.5 presents the their experimental computation. Section 3.6 concludes the chapter.

3.2 Performance indicators

We specifically took into account network configurations like the one depicted in Fig. 3.1, where a master station is connected, through a wireless network, to some passive stations. We supposed the data exchange over the network to be polling-based, that is we supposed the master to sequentially poll the passive stations, in particular sending each of them a request frame and receiving from it a response frame. As previously mentioned, this network configuration is typically employed in factory environments (e.g. in a device level industrial communication system, the master may represent the controller device while the wireless passive stations may represent sensors/actuators devices of a fac-

tory plant/machine) and master-slave protocols based on cyclic polling are frequently adopted in this context. Moreover, our case study, the extension of EPL based on the IEEE 802.11 wlan discussed in chapter 2, belongs to the set of considered network configurations.

We defined three performance indicators suitable for the network configurations we considered and, in particular, for our case study:

- the *polling time* of a wireless node;
- the *cycle time* of the system and, specifically, its minimum value;
- the *real-time throughput*.

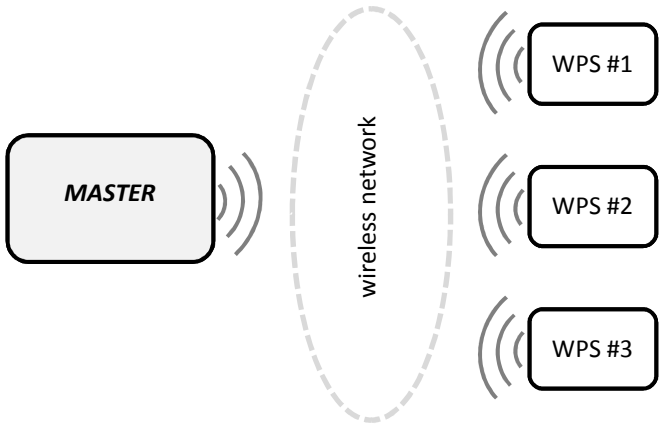


Figure 3.1: Example of considered network configuration.

3.2.1 Polling time of a wireless node

The polling time of a wireless node is the time elapsing between the sending of the request frame to the wireless node and the arrival of the correspondent response frame from it.

Clearly, the expression of such a performance indicator depends on the specific wireless network employed and the value it assumes is strongly influenced by possible backoff procedures as well as by the radio channel characteristics (bit error rate, presence of error sources, etc.). In general, the polling time of a wireless node, T_P, may be expressed as[1]

$$T_P = t_d + T_R \tag{3.1}$$

[1] In this chapter, capital letters are used to indicate random variables.

where t_d accounts for the deterministic component, whereas T_R accounts for the random one.

As an example, in case we employ an IEEE 802.11 wlan, for the deterministic component we have

$$t_d = t_f + t_a + t_i \tag{3.2}$$

where t_f is the time necessary to transmit the data frames (i.e. the request and response frames), t_a is the time necessary to transmit the acknowledgment frames and t_i accounts for the inter-frame times. For the random component we have

$$T_R = T_B + T_T + T_D \tag{3.3}$$

where T_B represents the time required by the backoff procedures of both the nodes involved in the polling, T_T is the time necessary to retransmit the frames due to possible collisions (e.g. in presence of interfering traffic) as well as to corruptions caused by transmission errors. Finally, T_D accounts for possible additional random delays introduced by the employed devices. For example, as shown in section 2.6.1 for the extension of EPL based on the IEEE 802.11 wlan, the frames used in the polling procedure could be delayed by the employed interconnection devices (e.g. access points) in presence of interfering traffic. The same effect (i.e. introduction of additional, unpredictable delays) may be experienced in the master-slave protocol due, for example, to the variable code execution times in the master when it is loaded with several different tasks.

3.2.2 Minimum cycle time

In cyclically operating industrial communication networks, the cycle time t_c, is a key parameter which is usually set by the user in an off-line configuration phase (e.g. the cycle time τ_{EPL} of an EPL network) and that characterizes the overall behavior of the system.

Clearly, in the considered systems the cycle time is limited by a lower bound, the *minimum cycle time* MTC, that represents the time necessary to poll all the passive stations in sequence, without introducing any idle time. Since the polling times of the passive stations are not necessarily constant, the minimum cycle time may be subjected to variations. This is particularly evident when wireless nodes are employed, due to the uncertainty of their data transmission times. The minimum cycle time for a polling-based system can

be trivially expressed as

$$MCT = \sum_{i=1}^{n} T_P^i \tag{3.4}$$

where n is the number of passive stations and T_P^i is the polling time of the i-th passive station defined in Eq. (3.1). The precise expression of T_P^i and, consequently of MCT, depends on the specific wireless network employed.

The IEEE 802.11 wlan (and actually other wireless networks, e.g. IEEE 802.15.4 wpan), for example, as described in section 2.2, uses a CSMA/CA technique to access the physical medium. Assuming for simplicity that there is only one transmission attempt and neglecting term T_d in Eq. (3.3), if each wireless node exchanges the same number of bytes with the master (i.e. t_d is the same for each passive device), we have:

$$MCT = n \cdot t_d + \sum_{i=1}^{n} (T_{BM}^i + T_{BS}^i) \tag{3.5}$$

where T_{BM}^i and T_{BS}^i are the backoff times relevant to the transmissions from, respectively, the master to the wireless node i and vice-versa. Consequently, MCT is given by a deterministic component, $n \cdot t_d$, plus a sum of $2n$ iid uniform random variables.

As a consequence, MCT is comprised within two bounds. The lower bound is given by

$$MCT^l = n \cdot t_d \tag{3.6}$$

While, supposing that all the stations execute a single backoff procedure, the upper bound is given by:

$$MCT^h = n \cdot t_d + 2n \cdot CW_{min} \tag{3.7}$$

Actually this is a worst case scenario since, as described in section 2.2, when a station accesses the network, if the medium is free for at least a DIFS it transmits immediately, thus in case of no interference we do not expect the stations to execute any backoff procedure.

Computing the MCT, beside giving useful insights on performance provided by an industrial communication systems, is fundamental to correctly configure the cycle time of the system itself so that it is never exceeded during the network operation phase and the system waists the minimum amount of time in the idle state.

3.2.3 Real-time throughput

The real-time throughput was formerly introduced by the IEC 61784 International Standard as a performance indicator for RTE networks but can be employed also for polling-based wireless industrial communication networks. The Real Time Throughput, RTT, is defined by the IEC 61784 International Standard as: *"the number of octets per second transmitted on a specific link exclusively relevant to real-time traffic"*. Assuming that only real-time communications take place in the considered network, the expression of real-time throughput relevant to the link between the master and the i-th passive device is the following

$$RTT^i = \frac{b^i}{t_c} \tag{3.8}$$

where b^i is the amount of bytes exchanged per cycle with the i-th passive device and, as mentioned above, t_c is the cycle time of the network. Supposing that b^i is maintained constant, clearly, the real-time throughput is inversely proportional to the cycle time. However, t_c can not be set below MCT and, hence, the real-time throughput has a higher bound

$$RTT^{i,h} = b^i/MCT$$

Additionally, it is worth noticing that reducing the cycle time down to values close to MCT causes an increased percentage of failed pollings, since there might be no sufficient time to poll the passive devices, with the consequent decreasing of the amount of bytes exchanged. In this situation, the network enters a sort of saturation state in which the throughput does not increase further. Such a phenomenon will be better discussed in section 3.5 where an example of RTT practical computation is presented.

3.3 Case study

The defined performance indicators have been analytically and experimentally computed for the extension of EPL based on the IEEE 802.11 wlan discussed in chapter 2. We considered a network configuration with the MN connected via the IEEE 802.11 wlan to a variable number of WCNs. In order to make the performance analysis as clear as possible, we avoided to insert wired CNs in the network configuration since, in any case, they introduce only deterministic delays (clearly the performance indicators we defined may be computed also in case of a hybrid wired/wireless polling-based communication system). We

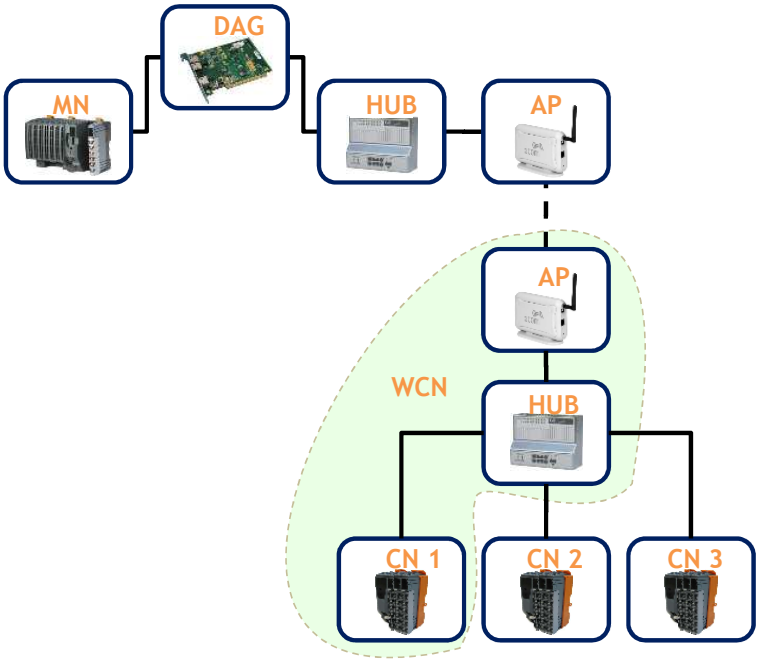


Figure 3.2: Implementation of extension of EPL based on the IEEE 802.11 wlan.

considered only the extension at the data link layer, with the MN connected to an access point device that allows for the communication with the IEEE 802.11 wlan where the WCNs of the wireless extension are located.

In detail, the wireless extension of EPL based on the IEEE 802.11 wlan was implemented as depicted in Fig. 3.2. The MN, a CPU X20CP1484 by B&R, was connected to a hub device and then to an IEEE 802.11 access point. An identical access point was connected to another hub device and then to some wired CNs in order to implement the WCNs. The use of hub devices instead of switches allows to considerably reduce the latencies: indeed, the adopted devices (AC808 by B&R), have a maximum latency of 0.5 μs, that is almost negligible for our purposes.

The wireless extension of the network was thus realized by employing two access point devices connected in bridge mode. Two different commercially available products have been tested, namely *Linksys WAP54G* and *3Com Office Connect*. The 3Com access points, in particular, implement also the Wireless MultiMedia extension, that is the IEEE 802.11e standard definig Quality-of-Service (QoS) functions, but this feature was not used during our experiments. Consequently, the access points were used in IEEE 802.11g mode and the radio channel was carefully selected as the one less disturbed by other

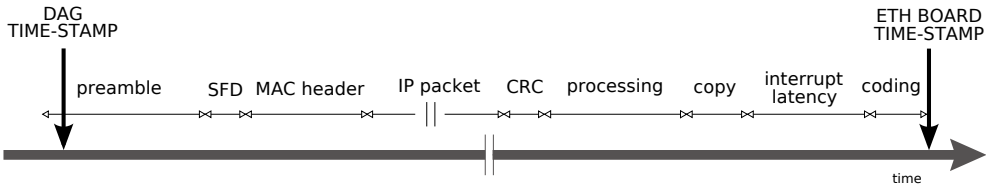


Figure 3.3: DAG vs Ethernet time stamping.

surrounding networks (channel 8), whereas the transmission rate was set to 54 Mb/s. All the measurements were performed at the MN side using a high precision time stamping Ethernet board, namely the DAG 3.6E by Endace Technology Ltd., inserted in series between the MN and the hub and equipped with two Ethernet interfaces. As depicted in Fig.3.3, the time stamping of a received packet is performed in a completely different way with respect to a conventional Ethernet board. The DAG board uses an internal generated high precision clock that has a resolution of about 60 ns. The time stamps are always taken during the preamble of the packet and are directly accessed by the driver of the DAG board. A conventional Ethernet board is necessarily less precise, since the time stamps are based on the PC clock and they include the latency to receive, copy and process the whole packet, plus a variable interrupt latency. Finally, the PC with the DAG board was running a traffic sniffing software (Wireshark) able to collect and store all the frames ensuring considerable precision and resolution.

All the experiments were conducted in a laboratory environment characterized by a very low level of interference. Such a condition was experimentally verified by means of a spectrum analyzer. Table 3.1 summarizes the most relevant parameters used in the experiments.

The polling time of a WCN, T_P, was evaluated by means of histograms representing the empirical pdfs of the performance indicator. Indeed the empirical pdf may highlight many more features if compared to the classical approach of analyzing only the mean and the standard deviation. All the measurement sessions have been performed over about 5000 EPL cycles, resulting in a maximum duration of 100 s (the maximum value of the EPL cycle time was 20 ms). The sessions have been purposely kept short in order to limit the possible occurrence of external interferences. The relatively limited number of points on which we have performed the pdf estimation, however, leads to quite noisy histograms. For such a reason, for all the pdfs that will be shown, a simple 5th order moving average filter was adopted in order to smooth the shape and

Parameter	Description	Value
n	Number of WCNs	1 to 3
l	Payload bytes exchanged	30
t_c	EPL cycle time	5 to 20ms
t_{to}	EPL timeout	5 to 10ms
r	Transmission rate	54 Mb/s
t_{slot}	Slot time	9 μs
$DIFS$	Distributed Inter-Frame Space	28 μs
$SIFS$	Short Inter-Frame Space	10 μs
CW_{min}	Min contention window size	15
CW_{max}	Max contention window size	1023
$t_{ack,out}$	Acknowledgment timeout	45 μs

Table 3.1: Case study parameters

underline the pdf. Finally, the chosen bin step was 1 μs, according to the time stamps resolution.

It is worth noticing that the polling time has been evaluated also for network configurations comprising more than a single WCN in order to highlight possible differences between subsequent queries. Moreover, the exchange of 30 bytes payloads per query ensures that only minimum size Ethernet frames were used.

3.4 Analytical evaluation of polling time of a WCN

The polling time of a WCN can be expressed as in Eq. (3.1).

3.4.1 Deterministic transmissions time

We refer to subsection 2.2.1 for the analysis of transmitting times of data and acknowledgment IEEE 802.11 frames. Indeed, in our experiments, we chose the maximum available transmission rates and the payload size was fixed to 64 bytes (deriving from the minimum length Ethernet frames we assumed used by EPL). Thus the theoretical analysis of subsection 2.2.1 is valid and we obtain: $t_{data} = 42$ μs and $t_{ack} = 34$ μs. Such values have been experimentally checked measuring, by means of a real-time spectrum analyzer, the in-channel power during a transmission. Results are shown in Fig. 3.4. As clearly visible, the

lengths of the transmitted frames are very close to the theoretically calculated values.

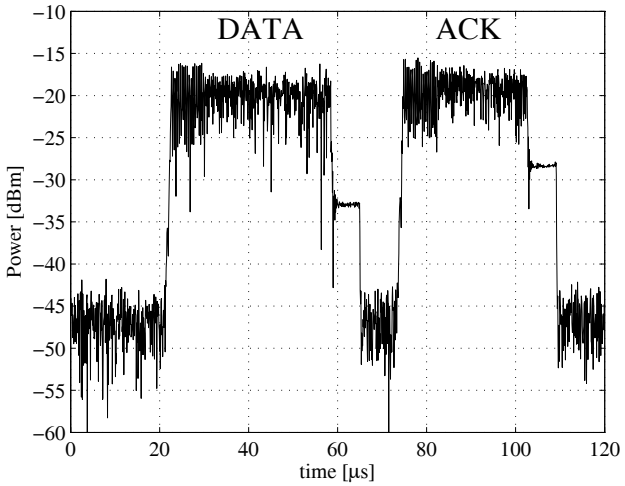


Figure 3.4: In-channel measured power.

In conclusion, given the above analysis, the deterministic part of the polling time of a WCN results

$$t_d = 2 \cdot (5.12 + t_{data} + t_{ack} + SIFS + DIFS) = 238.24\ \mu s \qquad (3.9)$$

where 5.12 μs is the transmission time over a 100 Mb/s link of a minimum size Ethernet frame and accounts for the transmission of *PReq* and *PRes* frames from the MN to the access point.

3.4.2 Backoff time

In the polling procedure initiated by the MN, the wireless medium is accessed twice. The first time by the MN that sends the *PReq* frame and the second one by the WCN that sends the *PRes* frame. Thus, in principle, both the accesses could require the execution of the backoff procedure. This is, however, related to the network load as well as to the components employed. For example, considering the TDMA mechanism defined by the EPL specifications and the IEEE 802.11 standard, when the MN accesses the network to poll the WCN, in absence of interference or competing traffic, the medium remains free for a $DIFS$, so the station transmits immediately, without executing the backoff procedure. On the WCN side, the backoff procedure that would be executed in case of the possible contemporary presence of the acknowledgment and of the *PRes* frame in the MAC sublayer transmission queue of the WCN, is likely

not performed, since the presence of elaboration delays on the WCNs causes the delayed transmission of the *PRes* frames from the application layer and, consequently, it allows the MAC sublayer to finish performing the acknowledgment procedure related to the successful reception of the *PRes* frame before starting to transmit the specific *PRes* .

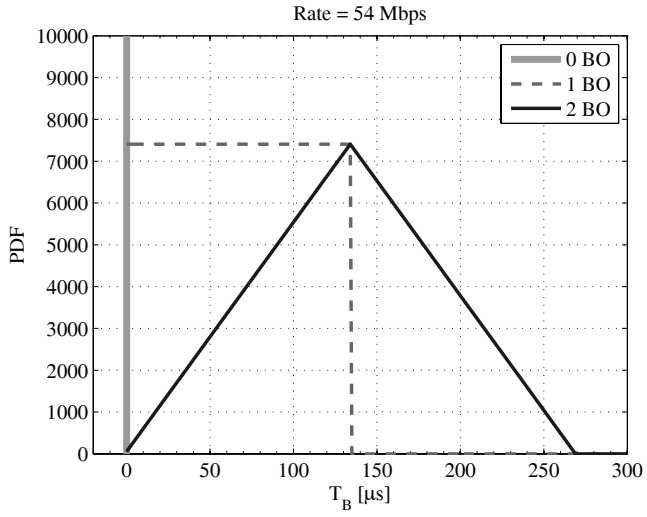


Figure 3.5: T_B pdfs.

In conclusion, three different behaviors for T_B are possible They are shown in Fig. 3.5 that reports the expected probability density function (pdf) of T_B. Specifically, it may happen that no backoff is carried out by any of the stations involved (0 BO), only one backoff procedure is carried out (1 BO) (either by the first access point or by the second one) or two backoff procedures take place (2 BO) (by both the access points).The experiments we will describe have been conducted in a laboratory environment characterized by a very low channel load and hence in absence of interference. Moreover, the distances between the devices have been kept very short in order to guarantee a very low bit error rate. Finally, we used commercially available components whose internal delays are likely longer than the acknowledgment time t_{ack}. Consequently, for all the above reasons, the contribution of T_B in Eq. (3.3) is expected to be null in the considered case study.

3.4.3 Retransmissions

Retransmissions of IEEE 802.11 frames are necessary in case of incorrect/miss reception that may be caused by transmission errors, as well as by collisions with frames issued by different wireless systems operating in the same band.

Thus, considering the test conditions described above, we may safely neglect the term T_T in Eq. (3.3). It is worth observing that, assuming the absence of interference, competing traffic and transmission errors allows to focus on the behaviors of the commercial components we used (and hence on their effects) which represents the basis for a correct analysis of the performance of the systems in real industrial applications.

3.4.4 Additional Delays

The use of access points in the practical implementation of the communication system allows to considerably reduce the forwarding delays. However, IEEE 802.11 access points have hardware/software architectures that may strongly influence the value of term T_D in Eq. (3.3). Indeed, Ethernet frames crossing an access point have to be decoded, acquired and forwarded to/from the radio hardware. Such operations may introduce unknown latencies due for example to the operating system behavior, that are clearly implementation dependent. The access points used are embedded systems running some minimal Linux distribution, but surely they can not be considered as real-time systems. As a consequence, such processes can introduce non negligible as well as random delays that can be only detected experimentally.

3.5 Experimental evaluation of performance indicators

We carried out extensive measurements on the set-up of Fig. 3.2 in order to evaluate the performance provided by the real communication system.

3.5.1 Linksys access points

The first test we carried out was the measurement of the polling time of a WCN when the network has only one WCN. The cycle time and timeout were set respectively to $t_c = 20$ ms and $\tau_{to} = 15$ ms. The obtained measurements and the relevant empirical pdf of T_P are shown, respectively, in Fig. 3.6 and Fig. 3.7. As can be seen, the pdf assumes non-zero values in the range $1350 \leq T_P \leq 1600$ μs. The shape is clearly multi-modal, showing a main mode similar to a triangle centered at 1414 μs, whose base measures 50 μs since it starts at 1390 μs and stops at 1440 μs plus several merging secondary modes. Such a 50 μ s duration can not be explained as deriving from the sum of

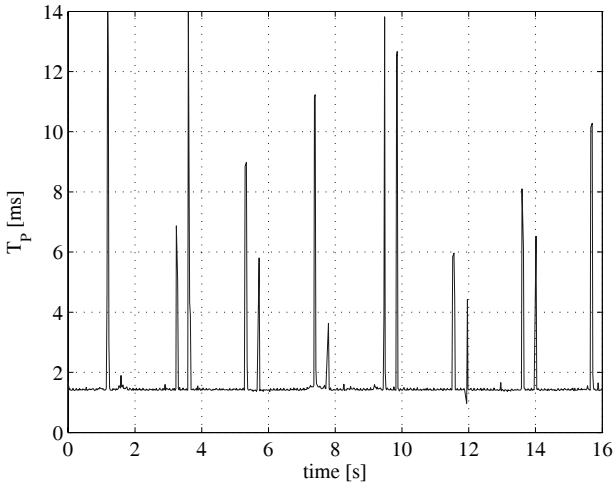


Figure 3.6: Measured polling, 1 WCN (using Linksys access points).

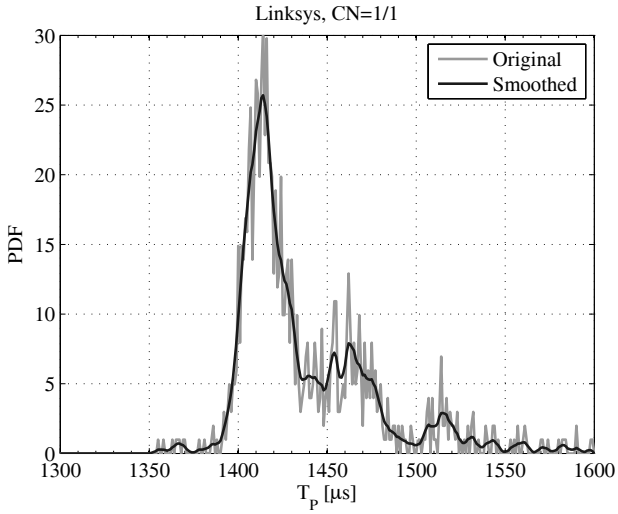


Figure 3.7: Polling time empirical pdf, 1 WCN (using Linksys access points).

two initial backoff, since as shown in Fig. 3.5 the triangle base would be in this case 270 μs-wide. Clearly, the experimental pdf shape could be larger than the theoretical triangle (due for example to some unpredictable random latencies) but it can not, in any case, be shorter. As we expected, the variability in the pdf is, hence, due to the term T_D in Eq. (3.3). Indeed, as already observed, the bridging process performed by a low-computational-capable processor using a non-real time operating system, may suffer from interrupts as well as processing latencies that are neither negligible nor a-priori estimable.

Finally, it is worth noticing that the pdf stops at 1600 μs but, conversely, the collected samples showed periodic couples of peaks every 2 seconds, up to 14 ms. This behavior is almost lost in the pdf since the occurrence of the peaks is statistically negligible. However, they may reveal really dangerous for industrial communication systems. This totally unexpected behavior is likely due to either some internal processes or sleep-awake cycles of the access point.

3.5.2 3Com access points

The 3Com access points have been initially used for measuring the polling of one WCN in a network with different cycle times. In all the experiments, unexpected peaks were never detected (to this regard, see Fig. 3.8) and, as expected, the backoff procedure was never carried out. Conversely, some unpredictable behaviors were observed as well. To this regard, Fig. 3.9 reports the measured polling time pdfs. As can be seen, the pdfs are very different

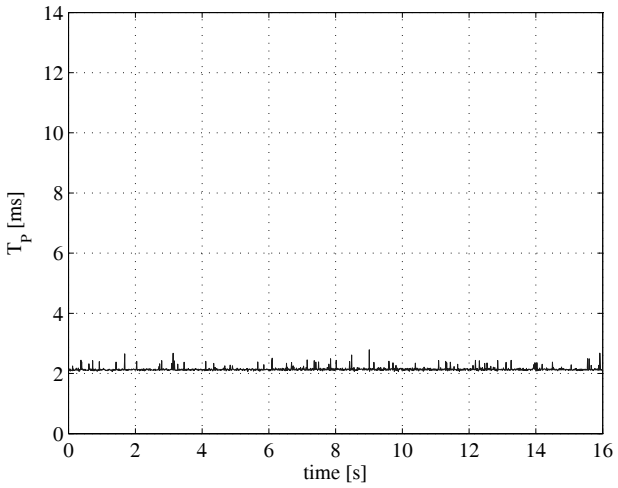


Figure 3.8: Measured polling, 1 WCN (using 3Com access points).

from those obtained with the Linksys access points, with regard to both mean

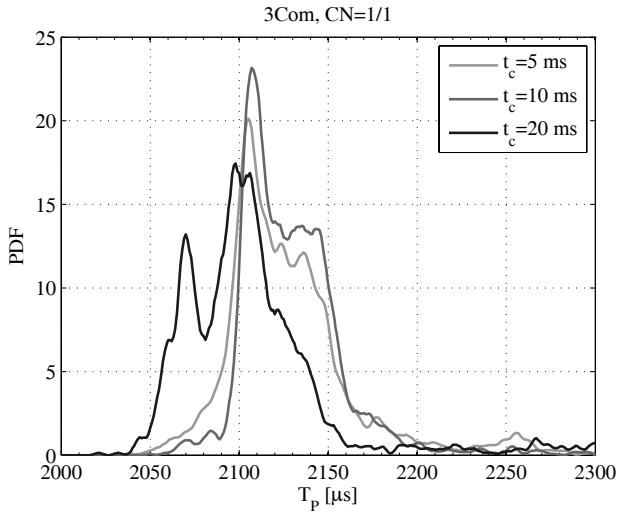


Figure 3.9: Polling time empirical pdfs, 1 WCN (using 3Com access points).

and shape. Indeed, the 3Com access points, when using 5 and 10 ms cycle time, show a peak centered at 2110 μs plus a rectangular shape almost bounded in $[2090, 2160]$ μs. Both at the left and at the right of this main mode, the pdfs show a heavy-tailed behavior. Using longer EPL cycles (for example setting $t_c = 20$ ms), the shape changes. In particular, it becomes more triangular-like, centered at 2100 μs and spanning from 2040 μs to 2160 μs. A second peak is noticeable at 2070 μs. This totally unpredictable behavior seems to follow this rationale: the use of a quite long cycle time allows to reduce the mean polling time at the expense of a greater variance. On the other hand, reducing the cycle time of the network leads to a more compact pdf with a greater mean and a smaller variance. It is worth noticing that the polling time of one WCN obtained using the 3Com access points shows a mean value that is approximatively 700 μs greater than the one shown by the Linksys access points. Once again, this enforces the deduction about the fundamental role played by implementation aspects reflected in the T_D term of Eq. (3.3).

The analysis has been completed considering a network with, respectively, 2 and 3 WCNs. The above described results are confirmed by the pdfs shown in Fig. 3.10, obtained from a network with 2 WCNs where, however, only the polling time of the first WCN is shown. Also in this case, the adoption of a more relaxed cycle time causes a spreading in the pdf and a shift down to smaller values. Thus, it may be concluded that the shapes shown in Fig. 3.10 are in good agreement with those in Fig. 3.9.

Conversely, this is not the case when considering the second WCN. The

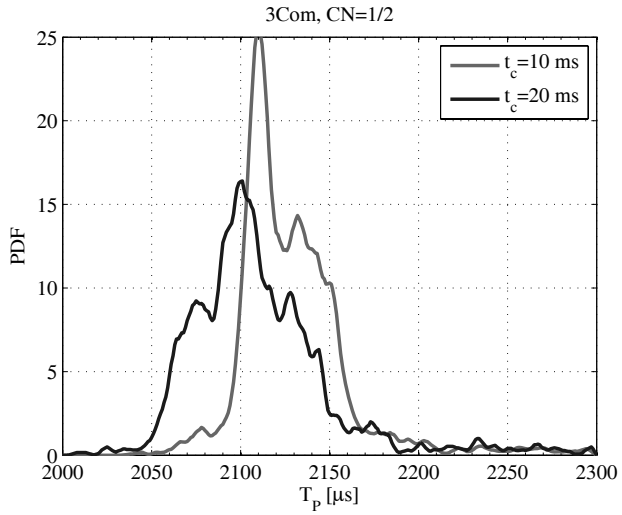


Figure 3.10: Polling time empirical pdfs, WCN 1 of 2 (using 3Com access points).

relevant pdfs of the polling time are shown in Fig. 3.11. As can be seen, in this case, a completely different behavior is observed. Indeed, the mean values of the pdfs are influenced by the selected cycle time. The shape for both the pdfs is triangular-like with a 50 μs-wide base, centered at 1710 μs for the case $t_c = 20$ ms and centered at 1726 μs for the case $t_c = 10$ ms. However, it is evident that no backoff is performed in both the cases, since the value of the variance is not compatible with the expected duration of the backoff periods. Consequently, it has to be accounted to some internal latencies of the access points. It is also interesting to observe that the mean polling time of the second WCN is 700 μs smaller than that of the first one, resembling a sort of sleep-awake cycle similar to that already noticed for the Linksys access points. Even if not reported, experimental pdfs measured for the case of 3 WCNs showed similar results. The pdf of the first WCN is similar to that of both Figs. 3.9 and 3.10, whereas both the second and the third WCNs showed identical behaviors in agreement with that reported in Fig. 3.11.

As a further final observation, it has to be remarked that all the experiments highlighted the huge difference in the expected theoretical mean for the polling time of a WCN. In fact, Eq. (3.9) provides a correct estimation for such a value (238.24 μs). On the other hand, the experimental results show that, for the Linksys access points, the mean is centered at 1420 μs, whereas for the 3Com access points the mean is variable (depending on the cycle time and on whether the WCN is the first in the cycle or not) and it ranges from 1710 μs to 2120 μs.

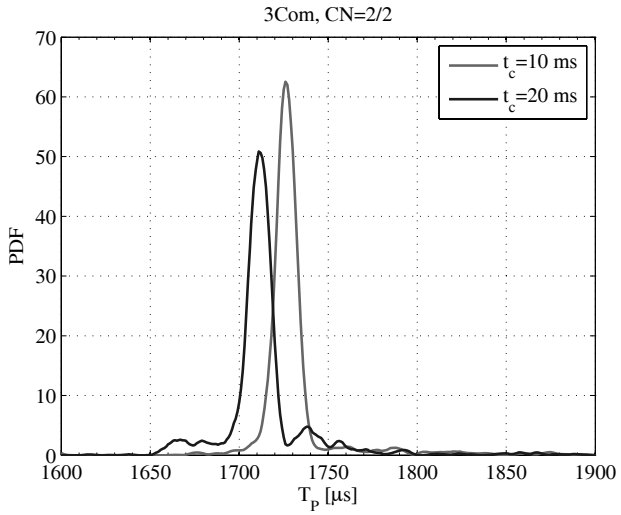


Figure 3.11: Polling time pdfs, WCN 2 of 2 (using 3Com access points).

3.5.3 Minimum cycle time

The minimum cycle time, has been evaluated for a network configuration with a variable number of WCNs ranging from 1 to 3. The relevant pdfs are shown in Fig. 3.12, whereas the corresponding statistics are reported in Table 3.2 (all values are in milliseconds).

As may be noticed, the behavior of the pdfs is strongly asymmetric, similarly to heavy-tailed distributions [67]. A further unexpected result is given by the mean MCT values. As can be derived from Table 3.2, on average, the polling of the first WCN requires 2.14 ms, whereas the polling of each of the subsequent ones is executed in 1.7 ms. The reason of such a non proportional behavior is not precisely explainable. However, it seems that the access point enters a sort of start-up phase at the beginning of each cycle, which eventually results in a longer time needed to poll the first WCN.

Table 3.2: Statistics of the MCT

N. of WCN	Mean	Std Dev.
1	2.14	0.066
2	3.88	0.083
3	5.62	0.095

As a last set of practical results, we provide the behavior of the MCT in terms of percentiles, as shown in Fig. 3.13, where the asymmetry is even more

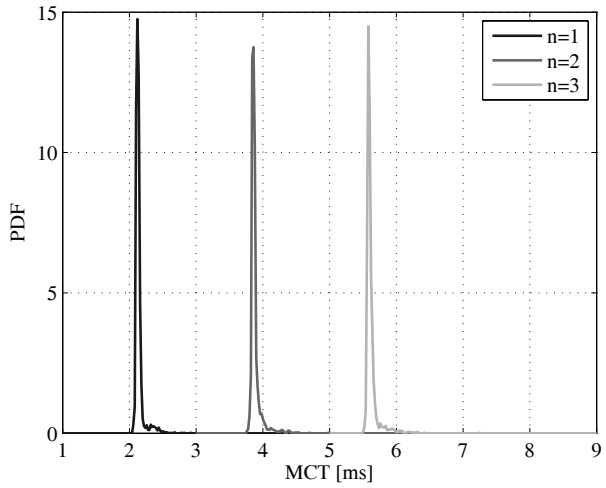


Figure 3.12: Measured MCT pdfs.

evident (in the figure, the expression $p = 1 : 99$ refers to the values comprised between the first and the 99th percentile). Clearly, some sporadic measured

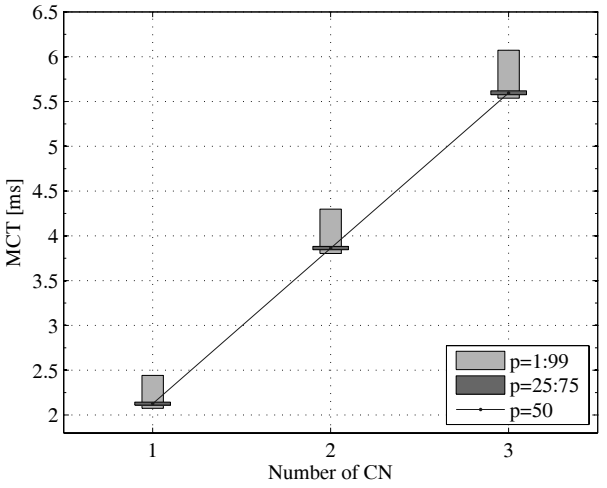


Figure 3.13: Measured MCT pdfs

values are greater than those shown in Fig. 3.13. In particular, it is worth mentioning that for the case with 3 WCNs, we measured a maximum value of 7.15 ms for the MCT.

3.5.4 Real-time throughput

The final measured performance indicator was the real-time throughput. In this case, a network with 3 WCNs has been considered and we measured the

RTT on the logical links between the MN and every WCNs. The results, shown in Fig. 3.14, report the mean values of the three RTTs.

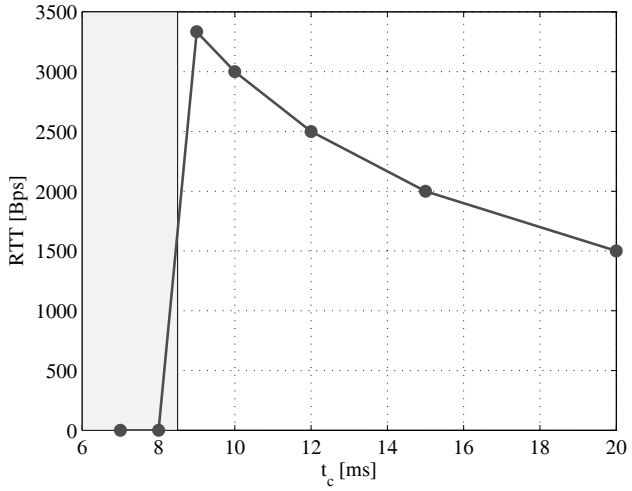


Figure 3.14: Measured RTT for a network with 3 WCNs.

As can be seen, contrarily to the description provided in section 3.2 (according to which the RTT was supposed to enter a saturation state for low values of the cycle time) the behavior of the RTT has a very sudden decrease for cycle times close to the lowest value of the MCT. This is due to a clearly identified phenomenon.In practice, reducing the cycle time of the network, implies necessarily a reduction of the EPL timeout. Unfortunately, the number of IEEE 802.11 frame retransmissions can not be accordingly reduced. Thus, the EPL timeout of a WCN being polled may expire while the access point is still trying to transmit, forcing the MN to move to the following WCN. In this situation, however, it may happen that the MN receives a *PRes* frame from a WCN different than that currently polled (i.e. from one of the WCNs previously queried that did not respond within the timeout). Consequently, according to the standard, the MN enters the non-operational state in which no data exchange with the WCNs may take place. This is the reason that causes the sudden decrease of the RTT. It is interesting to observe, in Fig. 3.14, that the phenomenon starts to take place for values of the cycle time greater than MCT^h (7.15 ms in this experiment), since it is triggered by a single occurrence of delayed *PRes* that does not imply necessarily a cycle time lower than the MCT.

3.6 Conclusions

The performance evaluation of industrial communication systems represents often a critical issue, which is even more accentuated when wireless networks, with the randomness they introduce, are employed. In this chapter we defined a set of performance indicators particularly suitable for polling-based wireless industrial communication systems and we provided analytical and experimental examples of their computation for a particular case study, which makes use of the IEEE 802.11 wlan.

The analysis revealed considerable differences between theoretical and experimental results in terms of unpredictable, unexpected delays. In particular, delays introduced by the employed access points, represented by the term T_D, represent a critical source of uncertainty. This led to the conclusion that the behavior of the adopted devices (in particular the two access points) is much more unpredictable than expected. The experience suggests that even the adoption of TDMA techniques would not lead to significant improvements, since a considerable degree of randomness is relevant to the components rather than to the protocols. The lesson learned encourages to plan further experimental analysis in order to better investigate the behavior of different types of components, for example including either professional access points or development boards with customizable (possibly real-time) operating systems.

Other future developments of the concepts outlined in this chapter are: the definition of additional performance indicators focusing on different applications (e.g. those based on producer-consumer protocols) and further examples of practical experiences (e.g. in more critical environments).

4 Real IEEE 802.11 wlan components behavior

4.1 Motivation

The results of the experiments presented in chapter 3, carried out on the practical implementation of the extension of Ethernet Powerlink based on the IEEE 802.11 wlan, have highlighted that the behavior of real network components strongly influences the performance provided by a communication system. In particular, the polling time of the WCNs was negatively affected by the relevant, variable, unpredictable delays introduced by the employed access points devices. Moreover, the measurements campaigns carried out using couples of access points from different vendors (Linksys and 3Com) have highlighted consistently different polling time behaviors (particularly interesting was the unexpected behavior presented by the Linksys devices that cyclically entered a sleep state causing polling times up to 14 ms).

These observations suggested us to focus our analysis on the behavior of network components that could be used in industrial applications where the tight timing requirements make the delay they introduce in communication even more relevant. As an example, RTE networks are usually requested to cope with cycle times in the order of few milliseconds, or hundreds of microseconds and with very limited jitters. These values are comparable with the elaboration delays of the controllers, with the latencies of the (real-time) operating systems they employ and, finally, with the delays introduced by some hardware operations like, for example, filtering which is an action typically performed by front-end devices. As a consequence, these delays have to be determined and taken into account when evaluating the performance of the networks in which the devices are employed. This is, clearly, not a simple task, since it requires a complete characterization of each specific device. The problem may

reveal even more complex when wireless networks are considered, due to the randomness they introduce.

Several analysis have been carried out aimed at investigating the behavior of industrial communication networks and protocols. The methods traditionally employed by the scientific literature, have led to the definition of theoretical models (derived from networks/protocols specifications) and often validated by numerical simulations. Anyway, although the results obtained with these methods are of undoubtable interest, they usually do not consider the real behavior of the network components employed (e.g. controllers, sensors/actuators, interconnection devices as switches, access points, etc.). It has to be mentioned, however, that in some of the aforementioned analysis, practical experiments have been actually carried out. Nonetheless, these experiments were not intended as a way of characterizing the behavior of the devices but, rather, they were focused on the overall performance of the considered communication systems [52].

In order to investigate the behavior of a common real wireless component that could be employed in industrial applications, for example in wireless extensions of already deployed wired communication systems [39, 60], we carried out extensive measurements on a IEEE 802.11 wlan access point. In particular, we considered a general purpose, commercially available access point. Clearly, we are conscious that different devices will show different behaviors, but the method we used to deduce the behavior of the specific access point we considered, as well as the analysis we present, are absolutely general and, as such, may be employed for evaluating the behavior of any other device (e.g. access points specifically developed for industrial applications, etc.). Indeed, our goal was, beside obtaining a precise characterization of a real access point behavior, to analyze the performance degradation that could occur if this behavior is not taken into consideration during the communication system development.

Specifically, we deduced the empirical pdf of the delay introduced by the access point and used such a pdf in a simulation model of the device itself. We then considered a hybrid network comprising two segments interconnected by the considered access point (a controller was connected to the wired segment, whereas some of the sensors/actuators devices were located on the wireless one). For this configuration (an infrastructure network, as defined by the IEEE 802.11 standard) we evaluated, by means of numerical simulations, the performance of two master-slave protocols typical of industrial applications.

In detail, the chapter is structured as follows. Section 4.2 presents the characterization of the behavior of the access point in term of delay it introduces in communication. Section 4.3 describes the network configuration employing such a device we consider for simulation along with the two master-slave protocols we evaluated. Section 4.4 provide and compare simulation results for considered network configuration running the two protocols, particularly focusing on how the negative effects of the access point behavior may be reduced by carefully setting some network parameters. Finally, section 4.4 concludes the chapter and provides directions for future activities.

4.2 Characterization of an access point behavior

We investigated the behavior of an access point using the experimental set-up of Fig. 4.1.

A simple communication system was built up between a controller connected to an access point and a wireless passive station where the controller periodically sent a frame of fixed length to the wireless passive station. Clearly, each transmitted frame was firstly received by the access point and then forwarded through the wireless medium to the passive station. In our experiments, we measured the time necessary to successfully carry out such an operation.

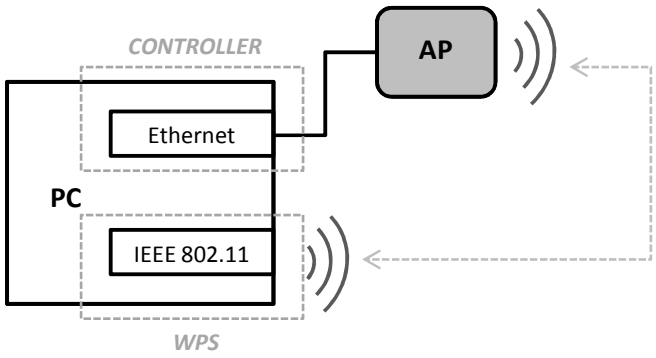


Figure 4.1: Experimental set-up.

In order to achieve precision in the time measurement, both the controller and the passive wireless station were implemented on the same Personal Computer (PC). To this regard, two different communication boards were used as stations: an Ethernet IEEE 802.3 LAN board as the controller and an IEEE 802.11 WLAN board as the wireless passive station. This implementation, analogous to the one described in [38], is effective since we were interested on

the latencies introduced in the communication by the access point only. Using a purposely developed software, we periodically (every sending period τ) generated a frame in the PC and sent it from the Ethernet board to the IEEE 802.11 board. We then measured the difference between the CPU time the frame had been delivered to the Ethernet board and the CPU time the frame had been successfully received from the IEEE 802.11 board.

The access point employed was a *3Com Office Connect* commercially available device. It was configured to specifically use the IEEE 802.11g protocol with a transmission rate of 54 Mb/s. The access point transmitting power was fixed to the maximum allowed value and the selected communication channel was the number 5, centered at 2.432 GHz, since, as verified with a spectrum analyzer, in the test environment it was not used by other wlans. The PC employed was a machine running the Linux operating system and equipped with two communication boards: 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 disabling its power management capabilities.

The experiments were conducted in an interference-free environment characterized by a very low bit error rate (both these conditions were experimentally verified). The access point was very close (less than 1 meter) to the passive station, in order to avoid path loss effects and minimize propagation delays. Given the almost ideal conditions of the measurement environment and the periodic basis onto which the query of the passive station was carried out, the backoff procedure typical of the IEEE 802.11 MAC Distributed Coordination Function was never executed by the access point, since the medium was always free for (at least) a DIFS when it tried to access it.

Under these assumptions, the expression of the time employed by a frame to go from the controller (master) to the passive wireless station (slave) results

$$T_{m \to s} = t_{tx,e} + T_d + t_{DIFS} + t_{tx,w} + T_l \tag{4.1}$$

where $t_{tx,e}$ is the time necessary to transmit (on the cable) the Ethernet frame from the controller to the access point, T_d is the delay introduced by the access point, t_{DIFS} is the DIFS time that has to be waited by the access point before transmitting, $t_{tx,w}$ is the time necessary to transmit (on the air) the IEEE 802.11 frame from the access point to the passive station and, finally, T_l accounts for the latencies introduced by both the controller and the passive station. It is worth observing that the times $t_{e,tx}$, t_{DIFS} and $t_{w,tx}$ are constant 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 LAN and the IEEE 802.11 WLAN specifications. On the contrary, the latencies introduced by the devices T_d and T_l are random variables since their values may depend on event-driven phenomena such as operating system latencies, buffer queues, conflicting tasks, etc.

In order to describe the behavior of the considered access point, we carried out extensive measurements, in presence of different traffic conditions, of the frame transmission time $T_{m \to s}$. From the measured value of $T_{m \to s}$ it is, in fact, simple to obtain the value of T_d, using Eq. (4.1) and assuming the time T_l known. Such a latter value, however, may be neglected in the considered scenario since, as discussed in [38], it is mainly related to the PC operating system latencies that are, typically, in the range of a few microseconds. Table 4.1 resumes the main parameters of the experimental set-up. The experiments were

Parameter	**Description**	**Value**
r	Transmission rate	54 Mb/s
c	IEEE 802.11g transmission channel	5
β	Transmitted data bytes (payload)	46
τ	Sending period	$5, 10, 20, 30$ ms
t_{DIFS}	Distributed Inter-Frame Space	28 μs
$t_{tx,e}$	Time to transmit the Ethernet frame	5.76 μs
$t_{tx,w}$	Time to transmit the IEEE 802.11 frame	34 μs

Table 4.1: Parameters of experiments

carried out varying both the sending period τ and the number of data bytes β transmitted. Fig. 4.2 reports the empirical pdf of T_d obtained for sending periods of, respectively, 5 ms, 10 ms, 20 ms and 30 ms, while keeping the frame length fixed to 46 bytes, which corresponds to a minimum size Ethernet frame payload. The basic statistics of the measured samples are shown in Fig. 4.3 and summarized in Table 4.2. The solid black line indicates the mean values of T_d, while the dark and light gray rectangles display, respectively, the 25th–75th and the 10th–90th percentile intervals.

A further set of measurements was carried out in order to evaluate the behavior of T_d 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. Results are provided in Fig. 4.4.

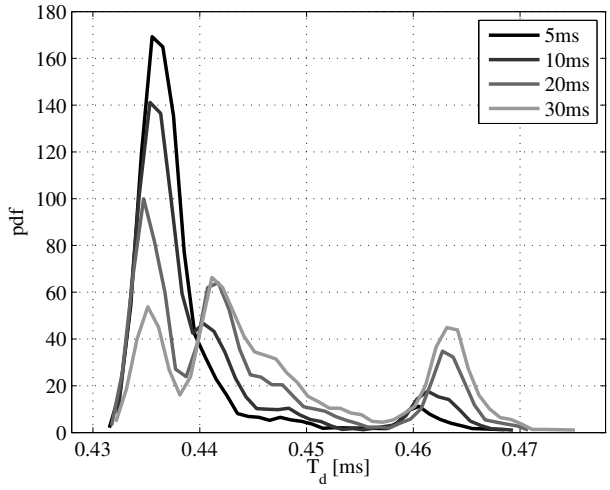


Figure 4.2: Empirical pdfs of the delay introduced by the access point.

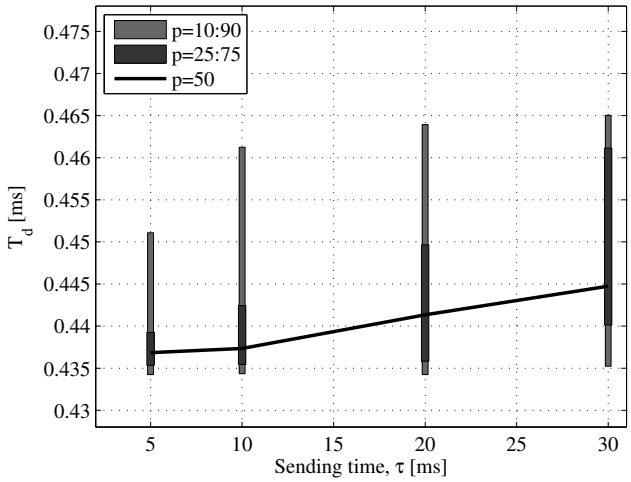


Figure 4.3: Access point delay vs. sending period ($\beta = 46$ bytes).

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

Table 4.2: Statistics of the access point delay

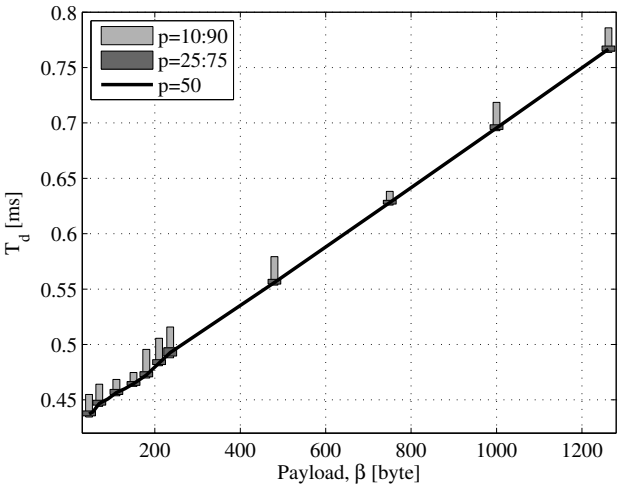


Figure 4.4: Access point delay vs. number of data bytes transmitted ($\tau = 3$ ms).

From the pdfs shown in Fig. 4.2 and from the results of all the experiments, the following deductions can be made:

1. *T_d is definitely a random variable*
   As expected, the delay introduced by the access point is not fixed and the values it assumes are not only related to the frame length.

2. *The randomness of T_d is tighter for lower values of τ.*
   As can be seen in Fig. 4.2, the lower the sending period, the better the shape of the pdfs, that tends to an almost deterministic delay. Such a behavior is particularly evident in Fig. 4.3 and is confirmed by the statistics of T_d provided in Table 4.2. In practice, it seems that the access point enters a sort of sleep state between two queries when the sending period is not sufficiently low. Thus, 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 in this work, looking at both Fig. 4.2 and Table 4.2, it may be estimated as $\tau^* \sim 10$ ms. It is worth underlining that pdfs obtained for $\tau < 5$ ms, even if not reported in Fig. 4.2, showed almost the same shape of the one obtained for $\tau = 5$ ms.

3. *T_d is lower bounded*

As can be seen in Fig. 4.2, that is actually a zoomed version of the figure showing the whole empirical pdfs, all the shapes indicate that the delay introduced by the access point is lower bounded. Moreover, the value of the lower bound of T_d is almost the same in all the cases (around 430μs) that likely represents the physical lowest limit of the access point delay.

4. *T_d 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 T_d does not change varying the packet length, but depends only on the sending period τ.

4.3 Master-slave protocols

The behavior of the access point described in section 4.2 has to be carefully taken into consideration when such a device is employed in systems requiring real-time performance, as it often happens for industrial applications. In particular, critical aspects are the non-negligible delay introduced by the access point in the communication and, more significantly in case the device is employed in cycle-based applications, the strong dependency of its behavior on the cycle duration.

In order to get some useful insights in this direction, we simulated the behavior of a hybrid (wired/wireless) network configuration, typical of industrial applications. As it is shown in Fig. 4.5, in the considered network configuration an access point is employed to connect a controller, located on the wired segment of the network, to three wireless passive stations (WPSs).

We assumed the controller and the wireless passive stations to introduce only negligible delays in the communication. This assumption is not necessarily true for real devices but it is reasonable in this context, since our goal is to point out the effects on the network configuration behavior of the delays specifically introduced by the access point. The access point behavior has thus been modeled according to the empirical pdfs presented in section 4.2. Furthermore, we assumed the statistical description of the access point delay to be the same for both the uplink and downlink directions. This means that,

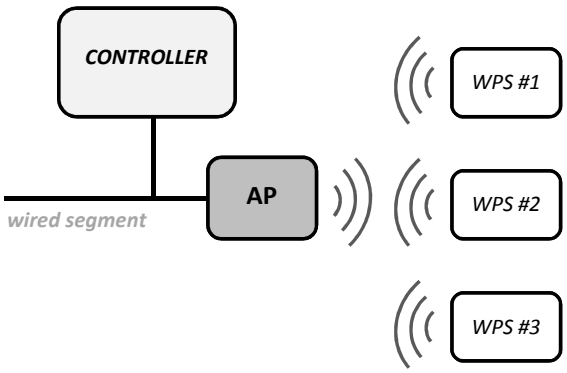


Figure 4.5: Simulated network configuration.

the time necessary to the controller to poll a passive station (that is to send a frame to a passive station and receive a frame back from it) results, with refer to Eq. (4.1), $2T_{m \to s}$.

We simulated the behavior of the network configuration of Fig. 4.5 for two different master-slave protocols:

- *Continuous Polling*

  For this protocol, the system operation is based on the continuous repetition of a cycle. The duration of the cycle is defined off-line by the user at the system configuration phase and is maintained constant during the system operation. At the beginning of each cycle the master station (the controller in the considered scenario) polls the slave stations (the wireless passive stations). Once the polling of all the slaves has been executed, the master simply waits for the beginning of a new cycle.

  In particular, the polling of slave station i is carried out through the exchange of two frames. A first frame, referred as *Req* (request frame, possibly carrying output data for the slave), is sent by the master to slave i in order to notify that its turn of communicating has started. Once received the *Req* frame, slave i sends back to the master a second frame, referred as *Res* (Response frame, possibly carrying input data for the master).

  A *timeout* time τ_{to} is defined by the user in order to avoid the master to indefinitely wait for a *Res* frame. The polling of slave $i+1$ takes place immediately after the successful execution of the polling of slave i, or, in case of failed polling operation, after the expiration of the timeout time (measured by the master starting from the *Req* frame transmission).

- *TDMA-based Polling*

  The system operation is the same as in the *Continuous Polling* protocol case, but the master station dedicates a time slot of fixed duration to the polling of each slave station. This technique resembles the polling strategy proposed in [66] for IEEE 802.15.4 devices. In this case, the master, once executed the polling of slave i, waits for the beginning of time slot $i+1$ before starting the polling of slave $i+1$.

  The duration of the time slot dedicated to each slave, τ_{slot}, is defined by the user in the off-line network configuration phase. Clearly it has to be carefully chosen in order to allow the slave to successfully answer to the *Req* frame and, at the same time, to avoid the degradation of the system performance caused by long waiting times between two consecutive polling.

In the *Continuous Polling* protocol case, the time necessary to the master to execute the polling of all the slaves results

$$T_{poll} = \sum_{i=1}^{n} T_{p,i} \quad \leq \quad \sum_{i=1}^{n} \tau_{to,i}$$

where n in our simulations is equal to 3. $T_{p,i}$ is the time necessary to poll the i-th slave and, since we assumed the up-link and down-link paths to be symmetric, it is given by $T_{p,i} = 2T_{m \to s}$. Moreover, with the term $\tau_{to,i}$ we want to stress out that the timeout time may be set differently for each slave.

In the *TDMA-based Polling* protocol case, the expression of the time necessary to the master to execute the polling operation is constant and results

$$T_{poll} = \sum_{i=1}^{n} \tau_{slot,i}$$

where with the term $\tau_{to,i}$ we want to stress out that the slot time may be set differently for each slave.

It is worth stressing out that in both the protocols the access point alternates active and inactive periods. For this reason, it is very important, during the configuration phase of the network, to take into account that if the duration of inactive periods is longer than τ^*, then the access point behavior (as deducted in section 4.2) will be more random, with negative effects on the overall system performance.

4.4 Simulation results

We firstly simulated the behavior of the system for the *Continuous Polling* protocol. All the simulations presented have been carried out using Matlab.

In order to configure the system, that is in order to choose the values of τ_{to} and, consequently, of the cycle duration τ_c, some considerations have to be made in advance. Firstly, with the *Continuous Polling* protocol, the access point is always active during the polling phase, thus its behavior is well described by the empirical pdf obtained in Fig. 4.2 for $\tau < \tau^*$ (we actually chose $\tau = 5$ ms). Therefore, we deduce $\tau_{to} = 2\max(T_{m\to s}) = 10$ ms (in order to successfully poll most of the slaves). The cycle time was consequently set to its maximum possible value $\tau_c = 30$ ms.

Anyway, it is worth noticing that, as can be observed from the shapes of all the empirical pdfs of T_d, T_{poll} will be likely much smaller than τ_c. As a consequence, the access point will be inactive for a long interval between the end of the polling of the third slave and the beginning of the polling of the first slave in the next cycle. Thus, it is reasonable to assume the delay introduced by the access point in the transmission of the first *Req* frame to be described by the empirical pdf obtained for $\tau > \tau^*$ (we actually chose $\tau = 20$ ms). Hence, as a conclusion, it is important to stress out that the access point behaves differently in the same cycle, depending on which node it is going to be polled.

As performance indicators, we chose to measure the jitter on the polling time, respectively of the first and third slaves of the system. We defined the jitter J_i on the polling time of a slave i as:

$$J_i = \frac{T_{p,i} - \tau_c}{\tau_c} \quad \%$$

in practice, J_i is the error on the query of a slave expressed as a percentage of the theoretical value of the polling time τ_c. The jitter represents a good indicator of the performance of the system since it describes the error affecting the sampling period of the variables exchanged between master and slaves.

The results obtained for the jitter in the *Continuous Polling* protocol case are shown in both Fig. 4.6 and Fig. 4.7. As can be seen, as a consequence of the bad behavior of the access point in the first polling, the jitter on the first slave is consistent. Indeed, neglecting failed pollings in some cases, it reaches values up to 20%. To this regard, it is worth pointing out that the theoretical maximum jitter that could be expected is $\tau_{to}/\tau_c \simeq 33\%$ which corresponds to a query of a slave which ends immediately before the timeout expires. Moreover, sometimes

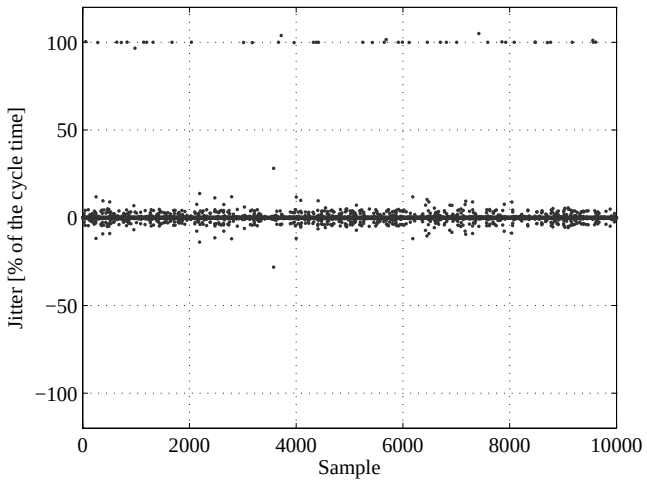


Figure 4.6: Jitter on the first slave. *Continuous Polling* case.

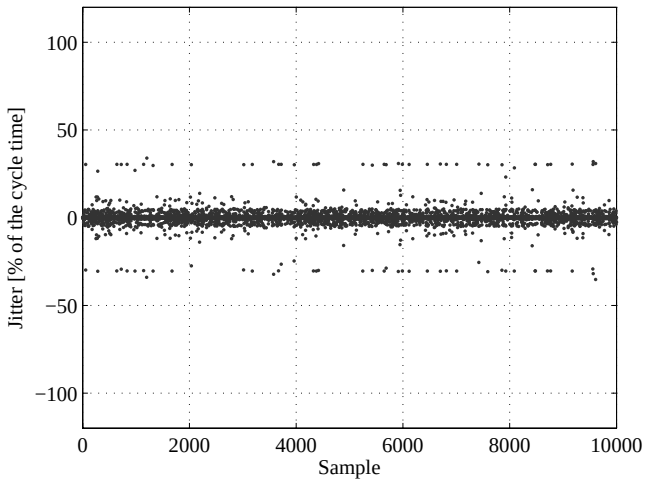


Figure 4.7: Jitter on the third slave. *Continuous Polling* case.

the first polling fails resulting in a jitter around 100%. This can be simply explained, resuming the fact that the first *Req* frame transmission delay follows the pdf obtained for $\tau = 20$ ms. As a consequence, this transmission could be delayed longer than the timeout τ_{to}, and the cycle would be consequently lost. Therefore, the time occurring between the subsequent (and successful) polling and the previous one is twice[1] the cycle time duration plus the randomness due to the access point delay. Finally, as can be observed in Fig. 4.7, the jitter on the third slave is more consistent than in the first one since it accounts also for the jitter on the first two slaves, even if the timeout mechanism has a positive influence on it (since it avoids complete cycle losses).

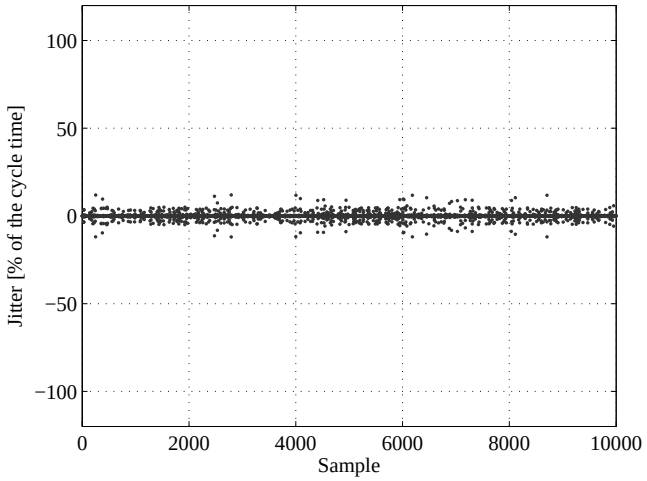


Figure 4.8: Jitter on the first slave. *TDMA-based Polling* case.

In order to avoid negative effects of the access point behavior on the jitter, we can adopt the *TDMA-based Polling* protocol. The TDMA mechanism, in fact, distributes the active periods of the access point along the whole cycle time so that, carefully choosing τ_{slot} it is possible to avoid such a device to enter the sleep state. We thus simulated the behavior of the system in presence of the *TDMA-based Polling* protocol. We chose to set the slot time as $\tau_{slot} = 10$ ms so that the cycle time still resulted $\tau_c = 30$ ms. Differently from the previous case, under these conditions, the access point inactive phase never exceeds 10 ms, which means that even in the worst case, the access point never entered the sleep state.

[1] It is worth noticing that it could also be three times or more τ_c, since, even if with a very low probability, two or more cycles could be lost. Nonetheless, in our simulation this never happened.

The results deriving from the simulations of the *TDMA-based Polling* are shown in Fig. 4.8. In this case the jitter affects the slaves in the same way, and it is much more limited than in the previous polling scheme (less than 10 %).

However, it is worth observing that the advantage introduced by the *TDMA-based Polling* protocol is strongly related to the duration of the τ_{slot} since, if this value is greater than τ^*, then the access point will enter the sleep state and, consequently, it will introduce a more relevant jitter. In any case, this polling scheme avoids the *propagation* of the jitter from one slave to another, that instead is particularly evident in Fig. 4.6 and 4.7 relevant the *Continuous Polling* protocol.

4.5 Conclusions

We characterized, by means of extensive measurement campaigns, the behavior of a commercially available, general purpose IEEE 802.11 access point, in terms of the delay it introduces in communication. We used the obtained results in order to evaluate, by means of numerical simulations, the influence of such a delay on the performance of a hybrid (wired/wireless) industrial communication network. Specifically, we focused on the jitter affecting cyclic polling operations, caused by the random behavior of the access point. The simulations carried out for two different master-slave protocols show, actually, significant results and provide some insights of relevant interest for the configuration of industrial communication systems employing such a device.

Future work in this context will concern the experimental analysis of different real wireless components. In particular, we are interested in the characterization of behavior of access points devices that are purposely designed for industrial applications such as, for example, those belonging to the Scalance family of products provided by Siemens [10].

Bibliography

[1] Rad-ism-900-xd-bus. URL http://www.phoenixcontact.com/. 14

[2] Actuator sensor interface - iec 62026/2, iec 94, en 5029. URL http://as-interface.net/. 11

[3] Bernecker & Rainer Industrie Elektronik GmbH. URL http://www.br-automation.com. 20

[4] Ethernet Powerlink Standardization Group: www.ethernet–powerlink.org. URL http://www.ethernet-powerlink.org. 20

[5] The IAONA Industrial Ethernet Planning and Installation Guide, IAONA Guide. [Online]: http://www.iaona-eu.com/home/downloads.php. 21

[6] LonWorks - ISO/IEC 14908. URL http://www.lonmark.org/. 11

[7] Multifunction vehicle bus (mvb) standard - iec 61375. 11

[8] P-Net standard - International Fieldbus Standard IEC 61158 Type 4. URL http://www.p-net.dk/. 11

[9] R-fieldbus. URL http://www.rfieldbus.de. 52

[10] Siemens Scalance Access Points family. URL http://www.automation.siemens.com. 112

[11] Smart distributed system (sds) standard. URL http://content.honeywell.com/sensing/prodinfo/sds/sdspec.stm. 11

[12] Wirelesshart international standard iec 62591. URL http://www.hartcomm.org/. 14

[13] RFC 768, User Datagram Protocol, August 1980. 23

[14] RFC 791, Internet Protocol, January 1981. 23

[15] RFC 793, Transmission Control Protocol, September 1981. 23

[16] Profibus Standard: Translation of the German National Standard DIN 19245 parts 1 and 2, 1991. 11, 52

[17] ISO/IEC 9545 standard – Information technology – Open Systems Interconnection – Application Layer Structure, 1994. 20

[18] Profibus DP Standard: Translation of the German National Standard DIN 19245 part 3, 1994. 31, 80

[19] IEEE Standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications, Aug 1999. 13, 59

[20] EN 50325-2: Industrial communication subsystem based on ISO 11898 (CAN) for controller-device interface -part 2: DeviceNet, June 2000. 11

[21] IEEE Standard 802.3: carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications, October 2000. 12, 20

[22] IEC 61158: Digital data communications for measurement and control - fieldbus for use in industrial control systems - parts 2 to 6, January 2000. 12

[23] IEC 61158-3,4: Digital data communications for measurement and control - fieldbus for use in industrial control systems - parts 3 and 4: Application layer service definition and protocol specification, communication model type 8, January 2000. 11

[24] Ethernet/ip specification, 2001. URL http://www.odva.org. 12

[25] IEEE Standard 802.15.4: Wireless medium access control (mac)and physical layer (phy) specifications for low–rate personal area networks (lr–wpans), September 2003. 79

[26] IEEE Standard for local and metropolitan area networks, IEEE 802.1D: media access control (MAC) bridges, June 2004. 53

[27] Profibus international: Profinet IO application layer service definition, application layer protocol specification" version 1.0, march 2004, 2004. 12

[28] Industrial Radio Network (InduraNET) p, 2007. URL http://www.pilz.com/. 14

[29] IEC 61784 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, December 2007. 12, 20

[30] 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, June 2007. 50

[31] G. Bernat, A. Burns, and A. Llamosi. Weakly Hard Real–Time Systems. *IEEE Transactions on Computers*, 50(4):308–321, April 2001. 61

[32] M. Bertocco, G. Gamba, and A. Sona. Is CSMA/CA really efficient against interference in a wireless control system? An experimental answer. In *IEEE Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany, September 2008. 55

[33] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18 (3):535–547, March 2000. URL http://hccc.ee.ccu.edu.tw/courses/wlan/performance_evaluation_bianchi.pdf. 61

[34] 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 *IEEE International Workshop on Factory Communication Systems*, pages 165–174, Torino, Italy, 2006. 42, 48, 56

[35] *CANopen Application Layer and Communication Profile, CiA/DS 301, Version 4.01.* CAN In Automation, International Users and Manufacturers Group e.V., June 2000. 11, 20

[36] G. Cena, L. Durante, and A. Valenzano. Evolution of standard fieldbus networks. In *Intelligent Systems and Robotics*. Gordon and Breach Science Publishers, Amsterdam, 2000. 30

[37] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino. Reasoning about communication latencies in real WLANs. In *IEEE Conference on Emerging Technologies and Factory Automation*, pages 187–194, Patras, Greece, September 2007. 60

[38] G. Cena, I. C. Bertolotti, A. Valenzano, and C. Zunino. Evaluation of response times in industrial WLANs. *IEEE Transactions on Industrial Informatics*, 1(3):202–214, May 2007. 13, 55, 61, 64, 101, 103

[39] G. Cena, A. Valenzano, and S. Vitturi. Hybrid wired/wireless networks for real-time industrial communications. *IEEE Industrial Electronic Magazine*, 2(1):8–20, March 2008. 47, 100

[40] *ControlNet specifications*. ControlNet International, March 1998. URL www.odva.org. 80

[41] J. D. Decotignie. Ethernet-based real-time and industrial communications. *Proceedings of the IEEE*, 93(6):1102–1117, June 2005. 42

[42] *EtherCAT: Ethernet for Control Automation Technology*. EtherCAT Technology Group, 2003. URL http://www.ethercat.org. 12

[43] *Ethernet Powerlink v. 2.0 Communication Profile Specification, Draft Standard v. 0.1.0*. Ethernet Powerlink Standardization Group, 2003. URL http://www.ethernet-powerlink.org. 12, 20, 48

[44] *Ethernet Powerlink v. 2.0 Communication Profile Specification, Draft Standard v. 1.1.0*. Ethernet Powerlink Standardization Group, 2008. URL http://www.ethernet-powerlink.org. 20

[45] European fieldbus standard EN50170. Worldfip protocol. URL http://cern-worldfip.web.cern.ch/cern-worldfip/pdffiles/WFPROTOC.PDF. 11

[46] D. Gao, J. Cai, and K. N. Ngan. Admission control in IEEE 802.11e wireless LANs. *IEEE Network*, 19(4):6–13, July/August 2005. 51

[47] E. N. Gilbert. Capacity of a burst-noise channel. *The Bell System Technical Journal*, 39:1253–1265, Sep. 1960. 42, 56

[48] B.R. Havenkort. *Performance of Computer Communication Systems–A Model based Approach*. Wiley, Chichester, UK, 1998. 27

[49] L. Kleinrock. *Queueing Systems*. John Wiley&Sons, New York, 1975. 39

[50] C. Koulamas, S. Koubias, and G. Papadopoulos. Using cut-through forwarding to retain the real-time properties of Profibus over hybrid wired/wireless architectures. *IEEE Transanctions on Industrial Electronics*, 51(6):1208–1217, December 2004. 52

[51] K. C. Lee, S. Lee, and M. H. Lee. Worst case communication delay of real-time industrial switched Ethernet with multiple levels. *IEEE Transactions on Industrial Electronics*, 53(5):1669–1676, October 2006. 35, 40, 55, 60

[52] S. Lee, K. C. Lee, M. H. Lee, and F. Harashima. Integration of mobile vehicles for automated material handling using Profibus and IEEE 802.11 networks. *IEEE Trans. on Ind. Electr.*, 49(3):693–701, June 2002. 100

[53] F. Li, M. Li, R. Lu, H. Wu, C. Mark, and K. Robert. Measuring queue capacities of IEEE 802.11 wireless access points. In *4th International Conference on Broadband Communications, Networks and Systems*, pages 846–853, Sept. 2007. 60

[54] *OPNET Modeler user manual.* OPNET Technologies. URL http://www.opnet.com. 31

[55] F. De Pellegrini, D. Miorandi, S. Vitturi, and A. Zanella. On the use of wireless networks at low level of factory automation systems. *IEEE Transactions on Industrial Informatics*, 2(2):129 – 143, May 2006. 67

[56] D. Qiao, S. Choi, and K.G. Shin. Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs. *IEEE Transactions on Mobile Computing*, 1(4):278 – 292, oct-dec 2002. 51

[57] Bernecker & Rainer Industrie Elektronik GmbH. X20 Series Input–Output modules. URL http://www.br--automation.com/. 54

[58] T.S. Rappaport. *Wireless Communications–Principles and Practice.* Prentice Hall, 2002. 56

[59] D. Kim Y. Doh M. Pham E. Choi S. Yoo, P. Chong and J. Huh. Guaranteeing real-time services for industrial wireless sensor net- works with ieee 802.15.4. *IEEE Trans. Ind. Electron.*, 57(11):3868–3876, February 2010. 14

[60] T. Sauter, J. Jasperneite, and L. Lo Bello. Towards new hybrid networks for industrial automation. In *14th International Conference on Emerging*

Technologies and Factory Automation, Palma de Mallorca, Spain, September 2009. 100

[61] *Serial Real-time Communication System (SERCOS) Specifications*. Sercos International, November 2001. URL www.sercos.org. 11

[62] R. Steigmann and J. Endresen. Introduction to WISA and WPS. URL http://www.eit.uni-kl.de/litz/WISA.pdf. 14

[63] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme. The industrial indoor channel: Large–scale and temporal fading at 900, 2400 and 5200 MHz. *IEEE Transactions on Wireless Communications*, 7(7):2740 – 2751, July 2008. 56

[64] J. P. Thomesse. Fieldbus technologies in industrial automation. *Proceedings of the IEEE*, 93(6):1073–1101, June 2005. 36, 47

[65] E. Toscano and L. Lo Bello. Cross–Channel Interference in IEEE 802.15.4 Networks. In *IEEE International Workshop on Factory Communication Systems*, Dresden, Germany, 2008. 55

[66] S. Vitturi, I. Carreras, D. Miorandi, L. Schenato, and A. Sona. Experimental Evaluation of an Industrial Application Layer Protocol over Wireless Systems. *IEEE Transactions on Industrial Informatics*, 3(7):275–288, November 2007. 31, 108

[67] W. Whitt. *Stochastic–Process Limits*. Springer–Verlag, Berlin, 2002. 95

[68] T. J. Williams. *A reference model for computer integrated manufacturing: a description from the viewpoint of industrial automation*. Instrument Society of America, 1989. 35

[69] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Transactions on Industrial Informatics*, 4(2): 102–124, May 2008. 55, 56

[70] A. Willig and A. Wolisz. Ring stability of the PROFIBUS token-passing protocol over error-prone links. *IEEE Transaction on Industrial Electronics*, 48(5):1025–1033, October 2001. 42

[71] 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 Transactions on Industrial Electronics*, 49(6):1265–1282, December 2002. 42, 48, 56

[72] A. Willig, K. Matheus, and A. Wolisz. Wireless technologies in industrial networks. *Proceedings of the IEEE*, 93(6):1130–1150, June 2005. 47, 52

[73] A. Zanella and F. De Pellegrini. Statistical Characterization of the Service Time in Saturated IEEE 802.11 Networks. *IEEE Communication Letters*, 9(3):225–2277, March 2005. 59, 61