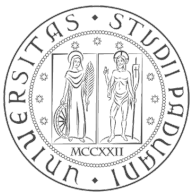


Università degli Studi di Padova

Facoltà di Ingegneria

Corso di Laurea in Ingegneria dell'Automazione

Analysis of Ethernet POWERLINK network and development of a wireless extension based on the IEEE 802.11n WLAN

Laureanda

Alessia Tagliapietra

Relatore

Prof. Stefano Vitturi

Correlatore

Dott. Federico Tramarin

Anno Accademico 2015-2016

A mio fratello Riccardo

Contents

List of Figures

List of Tables

Abstract

Nowadays, the industrial communication scenario is experiencing the introduction of wireless networks at all levels of industrial automation systems. The benefits deriving from such an innovation are manifold, even if wireless systems cannot be thought yet as a complete replacement of wired networks, but only in order to realize hybrid (wired/wireless) network. In this thesis we focus on Ethernet POWERLINK, one of the most popular Real-Time Ethernet networks thanks to its features and performance. Moreover, we propose an EPL wireless extension implemented by means of the IEEE 802.11n WLAN suitably tailored for the industrial communication through the dynamic rate adaptation algorithm RSIN. This solution can be achieved by customizing the open source protocol stack of POWERLINK and introducing a Linux ETH/WLAN bridge as interconnection between the wired and wireless segments. In order to carry out such activities we adopt methods mainly concerned with the theoretical analysis and experimental measurements on real systems. The outcomes of the experiments, on one hand revealed that the adoption of such a solution can actually provide some improvements to the performance of the EPL communication w.r.t. the current literature approaches. On the other hand, the assessments expose the potentially critical aspects of this implementation and highlight the direction for further interesting investigations in this framework.

Sommario

Al giorno d'oggi, lo scenario della comunicazione industriale sta sperimentando l'introduzione delle reti wireless a tutti i livelli dei sistemi di automazione industriale. I benefici derivanti da una tale innovazione sono molteplici, nonostante al momento non si potrebbero considerare i sistemi wireless come una completa sostituzione delle reti cablate ma solo come un mezzo per realizzare una rete ibrida (cablata/wireless). In questa tesi si analizza inizialmente Ethernet POWERLINK (EPL), una delle reti Ethernet Real-Time più popolari grazie alle sue caratteristiche e prestazioni. Viene poi proposta l'estensione wireless della rete POWERLINK basata sulla rete IEEE 802.11n (WLAN), con quest'ultima opportunamente ottimizzata per la comunicazione industriale attraverso l'algoritmo di dynamic rate adaptation RSIN. Questa soluzione può essere ottenuta da una parte, personalizzando lo stack open source di POWERLINK, dall'altra introducendo un bridge ETH/WLAN Linux, come interconnessione tra i segmenti cablato e wireless. Al fine di svolgere tali attività verranno principalmente adottati metodi come l'analisi teorica e le misure sperimentali eseguite su sistemi reali. I risultati degli esperimenti da un lato hanno rivelato che l'adozione di una tale soluzione può effettivamente fornire alcuni miglioramenti alle prestazioni della comunicazione rispetto agli approcci esaminati nello stato dell'arte. D'altra parte, tali risultati espongono anche gli aspetti potenzialmente critici di questa implementazione ed evidenziano la direzione per ulteriori indagini interessanti in questo campo.

Chapter 1

Introduction

1.1 Industrial communication networks

The Industrial communication networks (ICNs) have evolved considerably over the years according to the technology progress. The '80s have seen the development of several industrial communication protocols specifically for purposes of process and factory production control, commonly known as fieldbuses, able to provide deterministic performance. At that time, an European standard, i.e. EN50170, grouped the three most famous national fieldbuses: Profibus, P-NET and WorldFIP. Whereas, currently they belong to the IEC 61158 standard. Moreover, the well-known CAN (Controlled Area Network) belongs to ISO11898 standard [2].

However, in some applications the performance provided by fieldbuses may result not completely satisfactory due to their relatively low transmission rates (hundreds of Kb/s) as well as to the MAC protocols they adopt [3]. As a consequence, in the 90's, the performance provided by fieldbuses have begun to be considered too limited and the demand for a reliable communication system that would offer high performance rate, hight flexibility and across-the-board compatibility become more pressing.

IEEE 802.3 standard Ethernet [4] was the first to rise to this challenge: it was a well settled and tested technology that was free of patents and widely standardized to boot. Moreover, it had great potential to serve as a consistent, integrated communication solution that could support the interconnection of the control, process and field levels.
Although Ethernet was not originally designed to support real-time communications, a number of techniques have been proposed to adapt Ethernet so

that it can be used in the industrial field. The main issue of this research is to ensure that the tight delay constraints required by industrial applications are met, since frame delay is not deterministic in Ethernet.

Those networks, based on Ethernet technology, tailored for industrial communication are known as Real-Time Ethernet (RTE) networks. Nowadays, several commercial products by different vendors are available on the market, such as ProfiNet, Ethernet POWERLINK, EtherCAT, etc.. RTE networks are particularly suitable for employments at the lowest level of automation systems, where fast data exchange usually takes place between controllers and sensors/actuators.

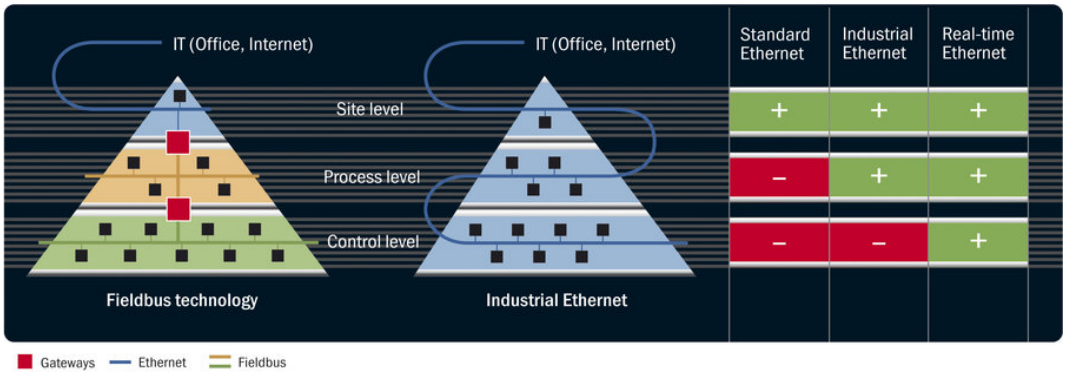


Figure 1.1: Fieldbus system evolution.

Figure 1.1 summarizes the technological evolution of the industrial communication systems. As can be seen, the main improvement achieved during this evolution is the possibility to adopt the same communication system for all the control, process and site level.

In this work of thesis we focus on Ethernet POWERLINK (EPL), [5], one of the most popular RTE network, which is an open technology defined worldwide by the IEC 61158 standard. EPL provides promising performance and real-time capabilities. The use of EPL in real-time industrial environments has gained increased interest, since it enables the reuse of existing hardware and avoids the costly development of ad hoc systems. A transmission speed of 100 Mbit/s and a synchronization accuracy of +/- 100 ns allow even the most demanding tasks in the areas of control engineering, robotics and motion control to be combined in a single network. The freedom provided by Ethernet standard topology also allows users to optimally adapt the wiring to the design of the machine and thus reduce wiring costs. Leading manufacturers of control, motion, sensor, pneumatic, vision and robotics technology already

rely on POWERLINK.

1.2 Wireless networks

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 automation. One of the most popular wireless technologies used to day is undoubtedly the IEEE 802.11 [6]. The adoption of such a family of wireless networks, also known as Wireless LANs (WLANs), is suggested by some of their appealing features.
The advantages of the employment of wireless technologies in industrial communication would be manifold: high transmission rates, support for mobility, reduced deployment and maintenance costs, reduced risk of cable/connectors failures, enhanced flexibility, etc. 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 fulfilment of tight requirements about reliable and real-time transmission typical of this field. Problems are caused, typically, by the CSMA/CA technique, which although has proven to be effective in general purpose communication systems, its adoption in the industrial context can be actually detrimental rather than beneficial. In detail, industrial traffic is often characterized by real-time requirements, such as low jitter on cyclic operations and bounded latency on alarm packets that, clearly, can be seriously compromised by latencies and randomness introduced by CSMA/CA based protocol.

Consequently, the majority of the currently defined WLANs cannot be thought of as an immediate and complete replacement of wired ICNs in factory environments, in particular, when real-time is one of the key issues. 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 connection between a wireless segment and the wired one is possible through some devices, for example like an Access Point, a Gateway, a Bridge etc. However, the conjunction and

interoperation of a wired ICN with a wireless network may be actually a critical issue. Indeed, wireless networks usually have transmission rates lower than the wired ones, 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 have to be solved in order to correctly integrated wireless protocols in a real-time application.

In this work of thesis we focus on the recent IEEE 802.11n High Throughput (HT) amendment, since it introduced several enhancements, at both the physical and MAC layers, that can be exploited to improve some significant performance figures for real-time for the Industrial Wireless Networks (IWNs), principally in terms of reliability and timeliness [7]. To this regard, we will exploit a new Rate Adaptation algorithm, named RSIN, introduced in [8], particularly targeted at real-time industrial traffic profiles. Since the RSIN algorithm is able to outperform all the other RA algorithm in terms of both reliability and timeliness, a deeper investigation in this direction looks as an appropriate choice.

The use of the new 802.11n amendment and of the new RSIN algorithm, open doors to explore a new possibility to improve the wireless extension of the POWERLINK protocol. This work, hence, addresses such a topic.

1.3 State of the art and contribution

The EPL protocol has been considerably studied in the literature lately. For example, in [9], [3] and [10] the authors present simulative assessments of EPL for distributed control and automation systems based on some of the most important performance metrics. Other studies consider different aspects of the protocol, such as the possibility to considerably reduce the times for data exchange between stations allowing the cross-communication [11]. Moreover, in [12], the authors present the event-triggered improvement of the PollResponse Chaining (PRC) mechanism, a new EPL standard feature aimed at increasing the network performance when nodes exchange small amount of data, especially if they are connected in line topology.

In conclusion, Ethernet POWERLINK has a certain degree of promotion and application in the field of industrial control. However, the introduction of wireless segments in the EPL communication scenario is still at a very initial stage. However, some meaningful contributions are worth to be mentioned.

In [13], [14] and [15], the authors present two different types of extension: a first one, implemented at the data link layer and a second based on a gateway. In both the cases, the wireless extension has strongly impact on the behaviour of the EPL network, particularly, this is mainly due to the retransmission procedure of IEEE 802.11.

To cope with latencies and randomness typical of the wireless scenario, some available industrial communication protocols (e.g. WirelessHART) adopt high layers services to resolve contentions and transmission errors, for example exploiting a master-slave relationship in a polling or TDMA-based scheme: hence, distributed and stochastic channel access schemes may result unnecessary, if not dangerous, since they might downgrade the overall performance. The first integration trial of the WirelessHART network to POWERLINK is described in [16]. Unfortunately, a similar approach is usually difficult to apply to commercial IEEE 802.11-based devices, since the use of such components, although justified by their affordability as derived from the high production volumes, imposes an implementation at Application level using a gateway.

This thesis, at the beginning provides a detailed description of the EPL features, according to [17], and investigates its behaviour through experimental sessions. Among the wide analysis of the EPL protocol behaviour done in literature, our contribution is the investigation about how the performances of a legacy EPL network are influenced by the introduction of an openPOWERLINK based device in the network. Usually, an EPL network is composed by specific hardware tailored to industrial communication, instead an openPOWERLINK based device is a desktop PC configured using the open source protocol stack of POWERLINK.

Furthermore, this thesis investigates a solution to scale up the current real-time Ethernet POWERLINK protocol, allowing the possibility to include wireless stations in the communication cycle whist reducing the problems, due to wireless randomness and jitter, founded in the literature approaches. In other words, this work of thesis analyzes and tests an original solution to integrate the 802.11n segment in a wired EPL network based network to POWERLINK. The proposed EPL wireless extension is especially based on an IEEE 802.11n, optimally tailored to the industrial communication through the dynamic rate adaptation algorithm RSIN, introduced in [8]. Moreover, the wireless extension integration into EPL can be achieved by customizing the protocol stack of POWERLINK and properly configuring a Linux ETH/WLAN bridge and the network parameters. In order to carry out such activities we adopt methods mainly concerned with the theoretical analysis and experimental measurements

on real systems. The obtained results can also be used as a starting point for the analysis of a wireless extension on other Industrial Ethernet approaches.

In detail, the thesis is organized as follows.

In **Chapter 2**, we focus on Ethernet POWERLINK. We provide a description of the EPL protocol and, particularly, of the EPL communication behaviour, as derived from the Ethernet POWERLINK specifications [17].

In **Chapter 3** an overview of the IEEE 802.11 standard is provided, that describes the specifications for Medium Access Control (MAC) layer and different types of physical layer, according to the IEEE 802.11 specifications [6]. Moreover, this chapter provides an accurate analysis of the aspects of interest for the industrial application of the the IEEE 802.11n amendment, and outlines the main motivation that suggested to consider this. Furthermore, the chapter describes the proposed EPL wireless extension based on the IEEE 802.11n, optimally tailored to the industrial communication through the dynamic rate adaptation algorithm RSIN, and on a Linux bridge, which realizes the interconnection between the wired and the wireless segment.

Chapter 4 describes the components of the experimental system that will be studied in this work. The application analyzed in this thesis consists of a hardware-based system which adopts a POWERLINK communication to perform real-time operations. This is achieved through B&R devices, relevant software and desktop PCs, configured using the open source protocol stack of POWERLINK.

In **Chapter 5** the behaviour of the both the EPL and the 802.11n protocols are characterized through a deep theoretical analysis of their communication performance figures. In particular we focus on the maximum time required to complete the polling operation in both the protocols. Moreover, the different sources of randomness of the IEEE 802.11n protocol are taken into account and a statistical analysis is performed in order to obtain a range of variability for this metric.

Chapter 6 shows the outcomes of a series of experimental campaigns about several configurations of EPL networks, both wired and hybrid (wired/wireless), which are set up using the hardware devices presented in Chapter 4. The aim of these experiments is to assess the protocols performance and compare them with the results of the theoretical analysis carried out in Chapter 5. In

particular, the polling time, whose expected behaviour has been analysed in detail, has been tracked in different scenarios.

Chapter 7 provides a summary of the work, listing the main activities conducted and the principal tools employed. A review of the most important results obtained through the thesis is presented. Finally, some considerations on possible future investigations are made, proposing other ways to test the wireless extension and possible actions to improve its performance.

Chapter 2

Ethernet POWERLINK

Ethernet POWERLINK (EPL) was introduced in the market as a proprietary standard of B&R Automation in 2001 [18], the following years it became a public technology. EPL belongs to Communication Profile (CP) 1 of Communication Profile Family (CPF) 13 of IEC 61158 International standard [19]. Ethernet POWERLINK Standarization Group (EPSG), founded in 2003, published the EPL specifications as an open standard, and currently manages the EPL network development. Its goals were to standardize and develop the POWERLINK protocol, furthermore EPSG's members share their know-how and actively contribute to improve the technology ever since. In addition, EPSG cooperated with the standardization bodies and associations, like CAN in Automation (CiA) [20]. This cooperation was a safe choice for the future, indeed CANopen is a robust protocol and it was often used in the industrial field. It follows that, from this cooperation the *CANopen over Ethernet* was born, officially approved in 2003 as Ethernet POWERLINK Version 2.0 [17]. This is the current used version which will used in this thesis.

Ethernet POWERLINK (EPL) is an industrial ethernet networking solution commonly used for Real-Time Ethernet (RTE) transmission of data, with more than 1.1 million POWERLINK systems installed [5]. In several application fields such as industrial control, transportation and national defence it has a certain degree of promotion and application .

In brief, absolute openness, maximum performance and unmatched features

are the main reasons for EPL's success.

To begin, EPSG standardizes POWERLINK technology fully compliant with the IEEE 802.3 Ethernet standard [4]. In such a way EPL benefits from the long-term evolution of Ethernet technology without requiring further investment. Especially, POWERLINK ensures all the benefits and flexibility of Ethernet technology as well as low costs and easier availability of the hardware.

Next, EPL provides maximum performance with cycle time down to 100 μs and system synchronization below 100 ns. Above all, to being able to connect up to 240 nodes in a single network, POWERLINK networks can be connected and fully synchronized to infinitely expand networking capabilities. EPL demonstrates its true strength when it comes to the growing number of electronically controlled drives, which in turn are required to exchange an ever-increasing amount of data.

Moreover, EPL integrates unmatched features and advantages from three different worlds: Ethernet, CANopen, and hard real-time capabilities. Consequently, EPL has rapidly become a worldwide leader for the real-time Ethernet solution. As a result of the integration between POWERLINK and CANopen, the process of migrating to EPL was smoother, indeed, users have a preference for a new protocol which is based on the older. Furthermore, in order to achieve its hard real-time capabilities, EPL relies on a mixed polling and time-slot procedure that allows only one node at a time to transmit data. In contrast to standard Ethernet, this procedure ensures no collision.

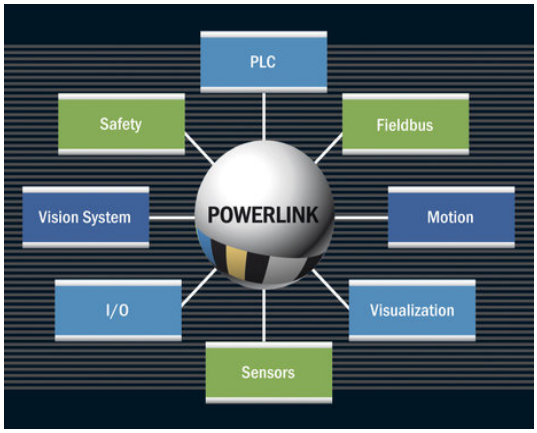


Figure 2.1: Ethernet POWERLINK can work in all systems.

POWERLINK is "*one network for all systems*", thus it is an industrial Ethernet solution designed to give users a single, consistent and integrated means for handling all communication tasks in modern automation (see Figure 2.1). It is generally suitable for all conceivable applications in machine and plant en-

gineering as well as for process industry applications. An EPL network is able to integrate all components in industrial automation, such as PLCs, sensors, I/O modules, motion controllers, safety controls, safety sensors and actuators, and HMI systems. Likewise, devices that do not belong to the immediate automation level can be included in the network environment as well, e.g. in the case of video cameras for site surveillance and access control. Moreover, a proper gateway also allows for the transparent communication with other non-POWERLINK communication systems within the asynchronous part of a cycle, i.e. the protocol enables the integration of various types of networks [5].

2.1 Reference model

The EPL standard specification [17] includes the description of the reference model, based on the ISO/OSI model, which is shown in Figure 2.2. As can be observed comparing this reference model with the Ethernet well-known one, POWERLINK is characterized by the definition of a Data Link layer (EPL DLL) protocol placed on top of the Ethernet medium access control (MAC) layer. For this reason, EPL frames are encapsulated and transmitted by means of Ethernet protocol data units.

It is possible to briefly describe the EPL reference model as follow, conversely the following chapters describe in detail them. The Physical and the MAC layers are exactly the same of Ethernet ones; on the contrary the EPL DLL, the core of EPL protocol, realizes the real-time communication of the system.

The use of IEEE 802.3 standard Ethernet at both the physical and MAC layers 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. Furthermore, this feature ensure that POWERLINK is completely compatible with legacy Ethernet [4].

The Network layer protocol IP and the transport layer protocol UDP and TCP are typical non real-time communication. But it is worth pointing out that the choice of those protocols allows the user to use the same network infrastructure both for the standard Ethernet traffic transmission and the EPL transmission of critical data.

Moreover, EPL DLL, at the higher layers of the stack EPL adopts an Application layer based on the CANopen profiles. The integration between CANopen and EPL guarantees the compatibility of the POWERLINK network with a large number of already deployed industrial communication systems (and the correspondents advantages).

Finally, the Network Management (NMT) is a transversal entity to OSIs

hierarchy, indeed it has to manage the network by initializing and monitoring network's nodes.

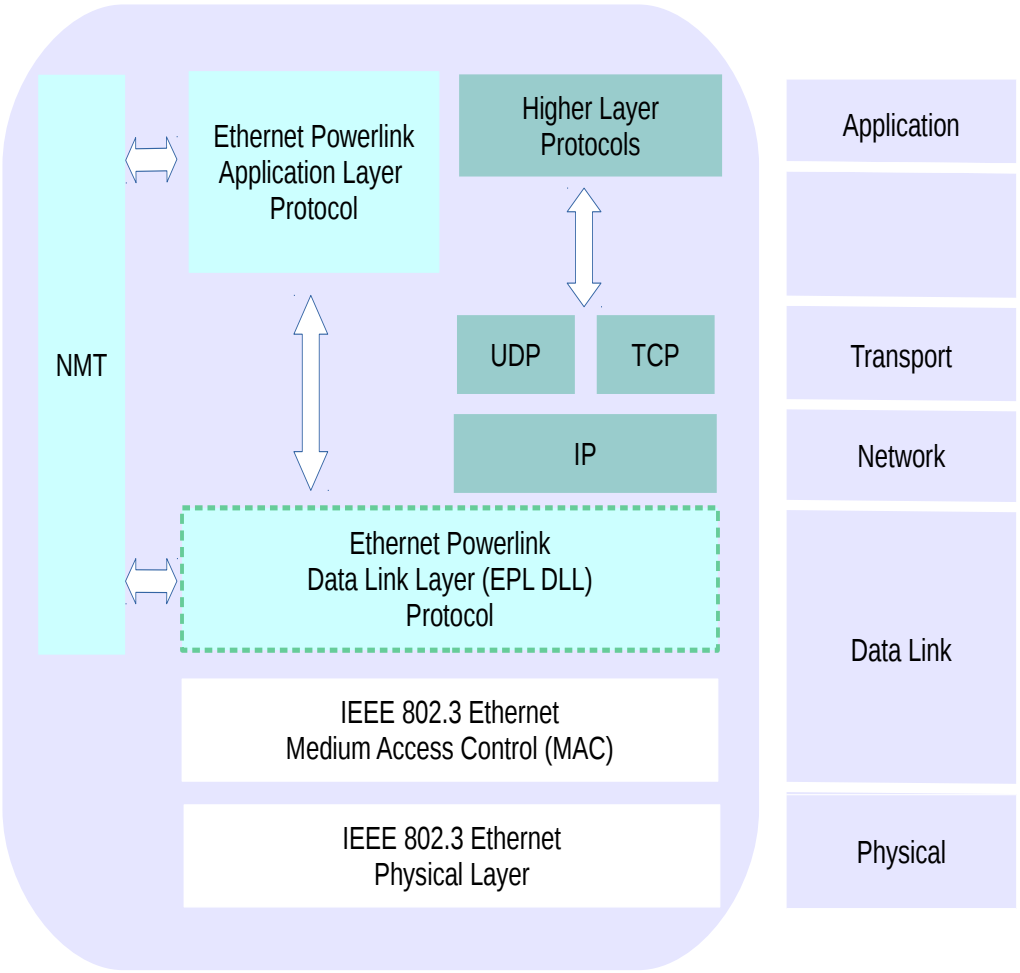


Figure 2.2: Ethernet POWERLINK reference model.

2.2 Physical Layer

The EPL Physical layer is defined as IEEE 802.3 standard Ethernet one. In particular, the POWERLINK specifications indicate an half-duplex transmission mode and standard patch cables (twisted pair).

2.2.1 Topology

An EPL network is made up of several nodes which can be connected through either traditional Ethernet hubs or switches. POWERLINK specifications encourage the use of hubs since they guarantee reduced path delay value and small frame jitter. On the other hand, they discourage the use of switches also since they may introduce additional non deterministic delays.

The IEEE 802.3 standard Ethernet constraints of 5120 ns as maximum round trip time has no more to be satisfied by EPL networks since POWERLINK does not cause collisions. Due to this leniency in the topology, users are completely free to choose any type of network topology whatsoever. In particular, the line structures with a large number of nodes, that are widespread in applications in the industrial field, are made possible. Mixed tree and line structures, like star, tree or hybrid tree-line configurations, are also available. An example of EPL network topology can be seen in Figure 2.3.

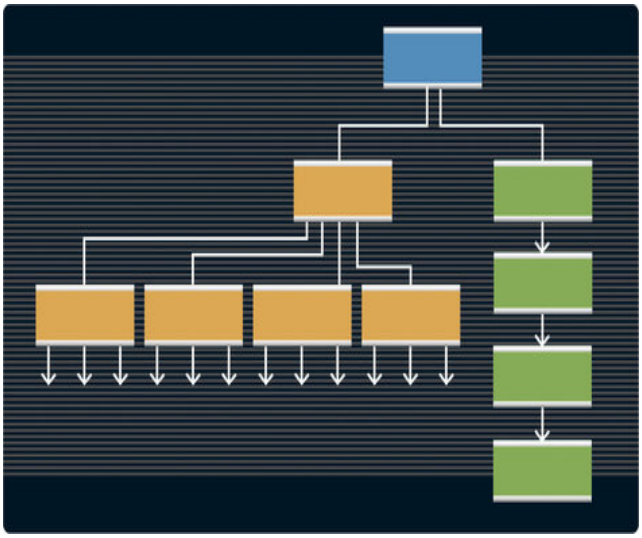


Figure 2.3: An example of EPL network topology.

2.3 Data Link Layer

The Data Link Layer can perform a variety of functions including coding/decoding data frames, controlling the order of sending and receiving data frames, flow control in transmission and error detection [11].

EPL makes use of the same MAC layer of the IEEE 802.3 Ethernet standard. As a result, the technique to access the transmission medium is also the

same. This implies that the technique is the *Carrier Sense Multiple Access/-Collision Detection* (CSMA/CD) technique. The core of the POWERLINK protocol is the Data Link EPL layer, which characterizes the POWERLINK behaviour.

There are two operating modes for a network: *POWERLINK mode* and standard Ethernet mode. The only difference between the two is the network behaviour: in the first mode it is defined by the EPL protocol and it is representative of the EPL real-time feature, instead in the second mode it is defined by IEEE 802.3 standard without applying any change.

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

2.3.1 EPL frame structure

As can be observed in Figure 2.4, an EPL frame is encapsulated and transmitted in the Data field of IEEE 802.3 standard Ethernet frames. According to the IEEE 802.3 standard, an ETH frame is composed by 3 fields: header (18 bytes), playload (up to 1500 bytes) and tail (4 bytes). In the ETH header field there are the MAC addressing fields (Destination MAc address and Source MAC address) and the type of Ethernet; in the ETH playload field there are the POWERLINK frame and finally in the ETH tail field there is the CRC32, which is a control bit used to report transmission errors.

Byte offset	**Field**	**Protocol**
0...5	Destination MAC address	ETH
6...11	Source MAC address	ETH
12-13	Ethernet Type	ETH
14	RES (1 bit) | Message Type (6 bit)	EPL
15	Destination Node ID	EPL
16	Source Node ID	EPL
17...n	Data	EPL
n+1...n+4	CRC 32	ETH

Table 2.1: EPL frame structure related to Figure 2.4.

An EPL frame consists of a header (3 bytes) and a payload (up to 1496 bytes). The header identifies the message type (SoC, PReq, PRes, etc.) and

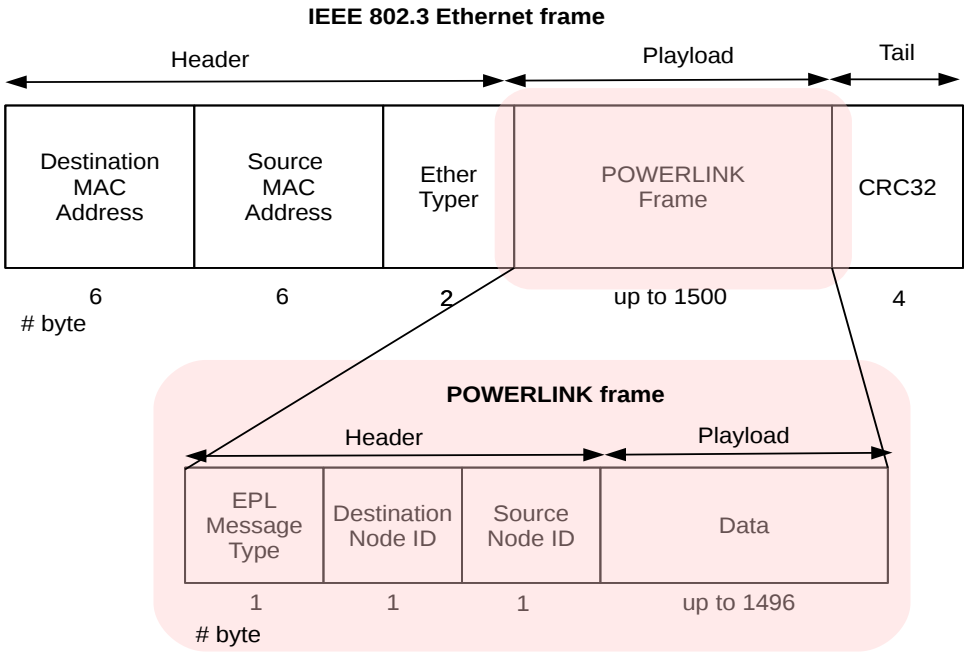


Figure 2.4: EPL frame.

contains addressing fields (destination node ID, source node ID), while the payload is dependent on the actual message type of the according POWERLINK frame.

Table 2.1 resumes the structure of an EPL frame above described.

2.3.2 POWERLINK Mode

In a POWERLINK network two different types of stations are specified : the Managing Node (MN) and the Controlled Nodes (CNs). As a rule each EPL network contains exactly one MN and several CNs. Moreover, each POWERLINK station (either MN or CNs) has a unique EPL address, called Node ID. This is a fixed node number, assigned before system start-up. Generally, 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 2.2 illustrates the complete POWERLINK Node ID assignment.

	Node Number
MN	240
CN	1 ... 239
Reserved	0,241 ... 255

Table 2.2: POWERLINK Node ID assignment

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. Therefore, network access is managed by the Managing Node and that can give to an other station the right to transmit. This centralized access rule precludes collisions, consequently the EPL protocol ensures the real-time feature in that the network behaviour is deterministic. In detail, the transmission medium access is based on a time division multiple access (TDMA) technique, which is referred as Slot Communication Network Management (SCNM) and is handled by the Managing Node. In POWERLINK Mode most communication packets are POWERLINK-specific frames, nonetheless an asynchronous slot is available for non-POWERLINK frames. UDP/IP is the preferred higher layer protocol in the asynchronous slot; however, it is possible to use any other protocol.

2.3.3 Ethernet POWERLINK cycle

The communication between the stations of an EPL network occurs on the basis of a cycle, the EPL cycle, managed by the MN and periodically repeated. The duration of the EPL cycle, the EPL cycle time t_{cycle}, is defined by the user during an offline network configuration phase and it is maintained constant during the network operation phase. The POWERLINK cycle time is determined depending on the industrial application: cycle times of up to several hundred of milliseconds may be good enough for soft real-time applications, e.g., temperature monitoring, while some motion control applications require cycle times well below one millisecond [21].

An example of the basic EPL cycle can be seen in Figure 2.5, and as can be observed it consists of four different periods:

1. **Start period**
   The start period is the beginning of the EPL cycle. As can be seen in Figure 2.6 at the beginning of a POWERLINK cycle, the MN sends a SoC (Start of Cycle) frame. The SoC is broadcast, so that it is received and processed by all other POWERLINK stations in the network. No

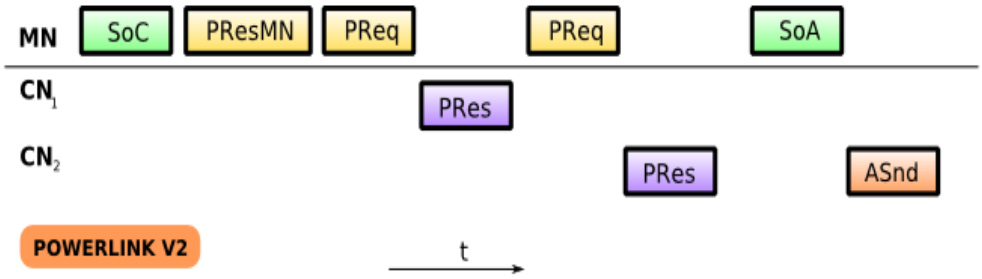


Figure 2.5: A complete POWERLINK V2 cycle.

application data is transported by the SoC, it is only used for synchronization. The SoC frame is the only frame independently generated and transmitted over the network every T_{cycle}, 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.).

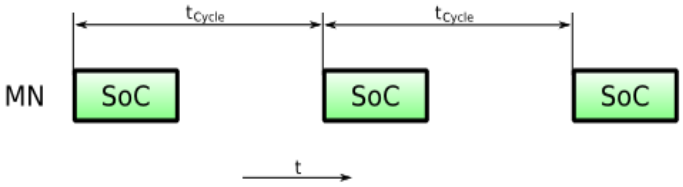


Figure 2.6: POWERLINK cyclic communication.

2. **Isochronous period**
   The isochronous period is a time slot dedicated to the real-time data exchange between the EPL stations. Immediately after transmitting the SoC, the cyclic real-time data exchange is realized by means of a sequential polling. In detail, the MN polls each CN in the network by sending a PReq (poll request) frame and waiting for a PRes (poll response) frame from the CN before moving to the next CN. The procedure can be seen in Figure 2.7. All stations are polled in order by the MN with a PReq. The PReq is addressed directly to CNi (i=1..239) as a unicast transmission, only that station receives the PReq in this frame. As a consequence PReq can only carry input data from the MN to the target CN. Conversely, the Pres frame is sent as a multicast and can therefore be received by the MN as well as by all other CNs in the network. Therefore, the PRes can not only send input data from the CN to the MN, but also allow

cross-communication among the CNs, this interesting possibility will be discussed in Section 2.3.4.

In order to avoid undesired (and potentially dangerous) delays, the POWERLINK protocol states that the polling of each CN has to be concluded within a fixed time interval, the Poll response timeout. If it expires before the reception of the PRes frame from the polled CN, then the MN moves on to the next CN. The data contained in these frames and the timeout time are specified by the user for each CN during the off-line network configuration phase.

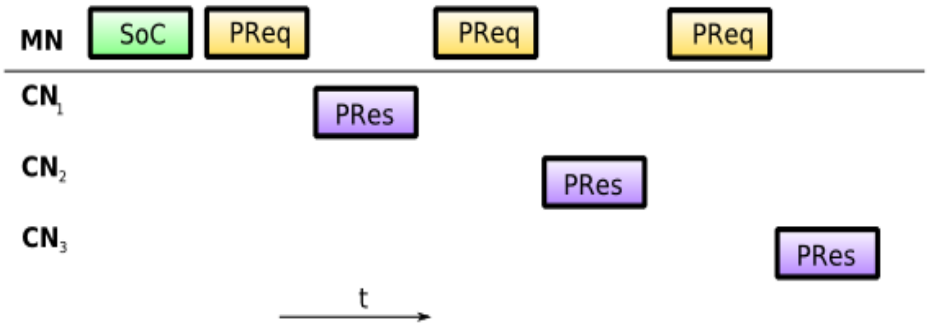


Figure 2.7: POWERLINK isochronous communication.

3. **Asynchronous period**
   The asynchronous period is a fixed time slot dedicated to the non real-time data exchange between the EPL stations. Asynchronous data differs from cyclic ones in that they must not be configured in advance but data are generated spontaneously by a POWERLINK station. In general, only one asynchronous frame can be sent per POWERLINK cycle. The CNs can tell the MN in their PRes frame that they have asynchronous data to send. Then the MN determines which station can start an asynchronous transmission, and includes this information in the SoA (Start of Asynchronous) frame. 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. The asynchronous period is dedicated to generic requests, typically deriving from the TCP/IP traffic (ARP, IP, visualization data, diagnostic data etc.) however, a maximum length (MTU = Maximum Transfer Unit) must not be exceeded. As discussed in [9], it might be possible to handle alarms during this period.

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.

Table 2.3 reports all the messages which can be used in a POWERLINK mode communication.

Symbol	**Name**	**Source & Destination**	**Meaning**
SoC	Start of Cycle	MN , Multicast	The MN sends this frame at the beginning of the POWERLINK cycle.
PReq	Poll request	MN, Unicast	The MN sends this frame along with the payload to each CN.
PRes	Poll response	CN/MN, Multicast	A CN sends a poll response with its data as a response to a poll request.
SoA	Start of Asynchronous	MN, Multicast	Marks the end of cyclic data communication and the beginning of asynchronous communication. Assigns send rights for asynchronous communication.
ASnd	Asynchronous Send	CN/MN, Multicast	This frame transports asynchronous data.

Table 2.3: Messages possibly exchanged during a POWERLINK Mode communication.

Normally, POWERLINK stations are "continuous", which means that they are addressed during every POWERLINK cycle with a poll request and a poll response. Using the *Multiplexed station setup* the communication with all multiplexed stations is distributed over a specified number of POWERLINK

cycles. The Figure 2.8 shows an example of this setup where the CNs 1, 2, 3 are continuous stations and CNs 4 to 8 are multiplexed stations. The continuous stations send important data in every cycle, instead the multiplexed ones send data only in some cycles.

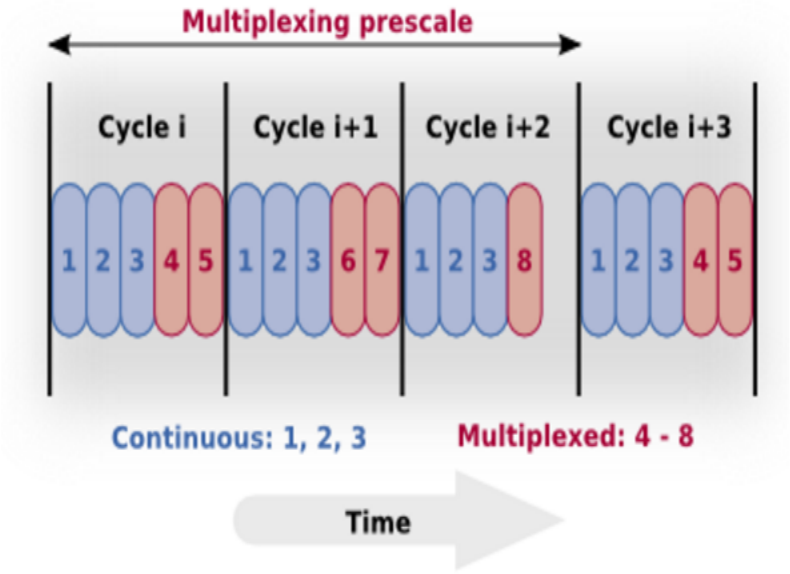


Figure 2.8: An example of multiplexed statin setup.

2.3.4 Cross-communication

Ethernet POWERLINK network can support cross-communication, which has great advantages. Indeed, cross-communication allows the CNs to direct swap information each other. In particular, it allows the times for data exchange between stations to be reduced considerably, since the data must not be copied in the MN. In this case, Poll request and poll response frames are used to exchange cyclic data in the network, with the constraints that a CN only transmits when it receives a directly addressed request (PReq) from the MN. As shown in Figure 2.9, the data (incorporated in the PRes frame) of cross-communication is transmitted from CNx to CNy. In Figure 2.9, as soon as CNx receives the PReq frame from MN, it broadcasts the Pres frame to the network. Then every node can detect this PRes frame and the attached data without being retransmitted by MN.

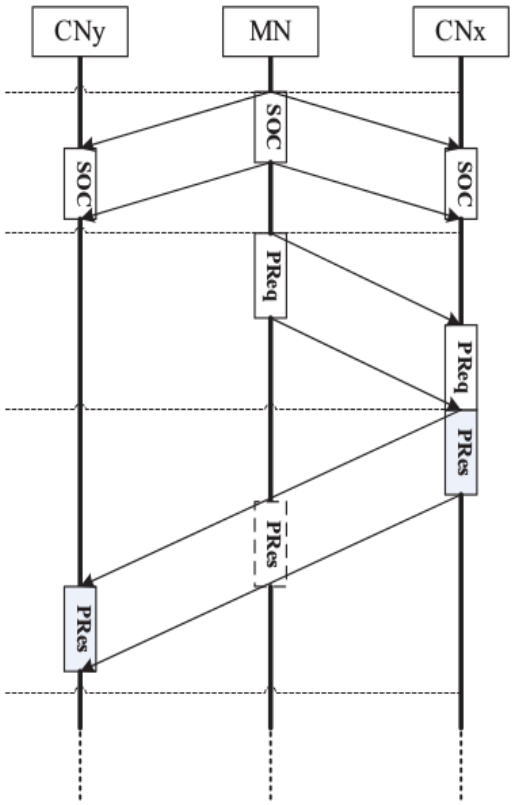


Figure 2.9: Packets sequence diagram of Cross-communication.

2.3.5 EPL Addressing

The EPL protocol defines an addressing system based on MAC address and Node ID: the first is used to ensure transparency with the IEEE 802.3 protocol, instead the second is a characteristic of EPL protocol. Moreover, each IP-capable POWERLINK node possesses an IPv4 address, a subnet mask and default gateway. The private class C Net ID 192.168.100.0 shall be used for a POWERLINK network. Indeed this class provides 254 (1-254) IP addresses, which matches the number of valid POWERLINK Node ID's.

Generally, an IP address has two components, namely the network address and the host address. If we take the example 192.168.100.1 and divide it into these two parts we get the following:

$$\begin{aligned} \text{Network:} &\quad 192.168.100. \\ \text{Host:} &\quad .1 \end{aligned} \tag{2.1}$$

This means that we should be able to give the IP addresses to 254 hosts. Since the EPL protocol allows to define up to 239 CNs we could give an IP addresses to 239 hosts (instead of 254), from 192.168.100.1 to 192.168.100.239. Two addresses that cannot be used are 192.168.100.0 and 192.168.100.255 because binary addresses with a host portion of all ones and all zeros are invalid. The zero address is invalid because it is used to specify a network without specifying a host. The 255 address (in binary notation, a host address of all ones) is used to broadcast a message to every host on a network. In other words, the first and last address in any network or subnet cannot be assigned to any individual host.

In particular, the Host ID of the private class shall be identical to the POWERLINK Node ID, in other words the last byte of the IP address (Host ID) must have the same value as the POWERLINK Node ID. Figure 2.10 illustrates the construction of the IP address. In such a way, the knowledge of the Node ID of a POWERLINK node and its IP address and vice versa can be determined easily without any communication overhead. The default subnet mask of a POWERLINK node is 255.255.255.0 and the Default Gateway preset shall use the IP address 192.168.100.254.

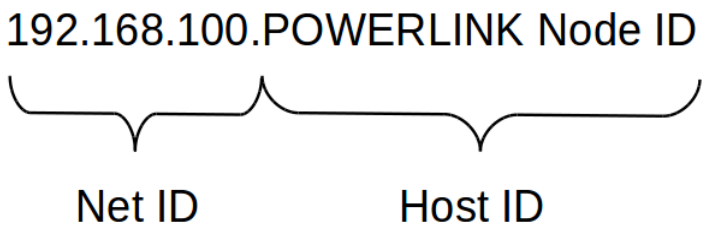


Figure 2.10: Construction of the IPv4 EPL address

2.3.6 Last improvements of the POWERLINK protocol

In the last release of POWERLINK protocol, there are some interesting improvements. Firstly, the *Multiple Asynchronous Send* feature allows to send multiple asynchronous frames per cycle, maximizing the amount of transfer data. Secondly, the *PollResponse chaining* allows a CN station to send its PollResponse immediately after the PollResponse of the previous station. In such a way it is possible to handle a larger number of stations in the same cycle time.

2.4 Application Layer

The EPL specifications define an application layer protocol based on the CANopen standard which, basically, introduces a set of communication objects to be exchanged over the network as well as a set of communication 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.

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.

2.5 Network Manager NMT

A communication profile has to handle the network holding it in the correct operational state. The NMT State Machine, baked up by CANopen, has to manage different station states: initialization, pre-operational, operational and not active.

- **Initialisation**: the node automatically shall enter this state. In this state the network functionally shall be initialised.
- **Pre-Operational**: the station is connected to network and they synchronize each other.
- **Operational**: the node is ready to transmit.
- **Not active**: a node enter in this state only due to a critic event. The node seem power off because only the NMT services can access to it.

At the beginning of network operations, the NMT initializes both MN and CNs in the same way, as shown by Figure 2.11, then there are two different NMT State Machines to hold both the station types in the correct operational state: the MN NMT State Machine is described by Figure2.12, whereas the CN NMT one is shown by Figure 2.13.

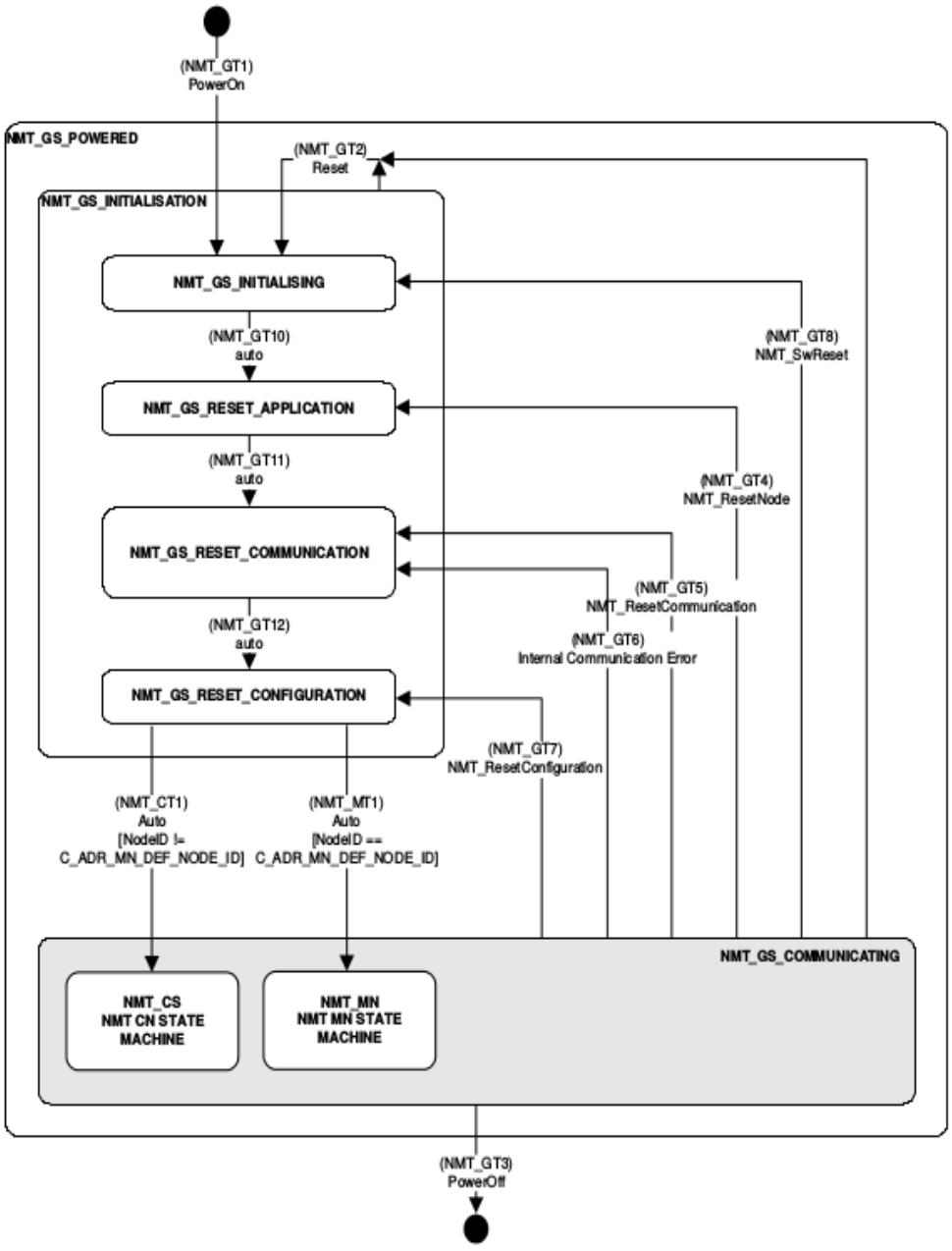


Figure 2.11: The NMT State Machine initialization procedure.

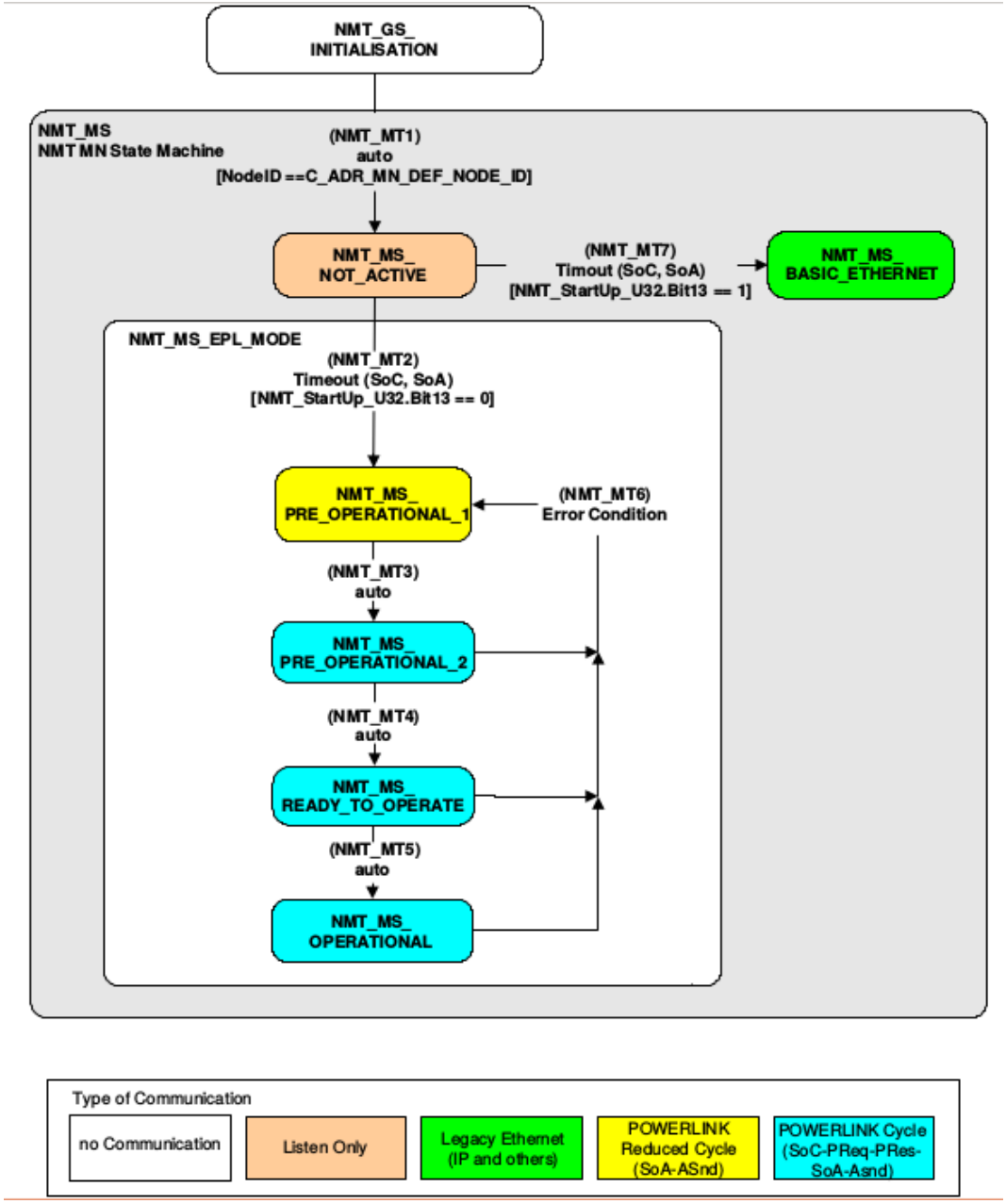


Figure 2.12: The NMT State Machine behaviour to keep the MN station in the correct operation state.

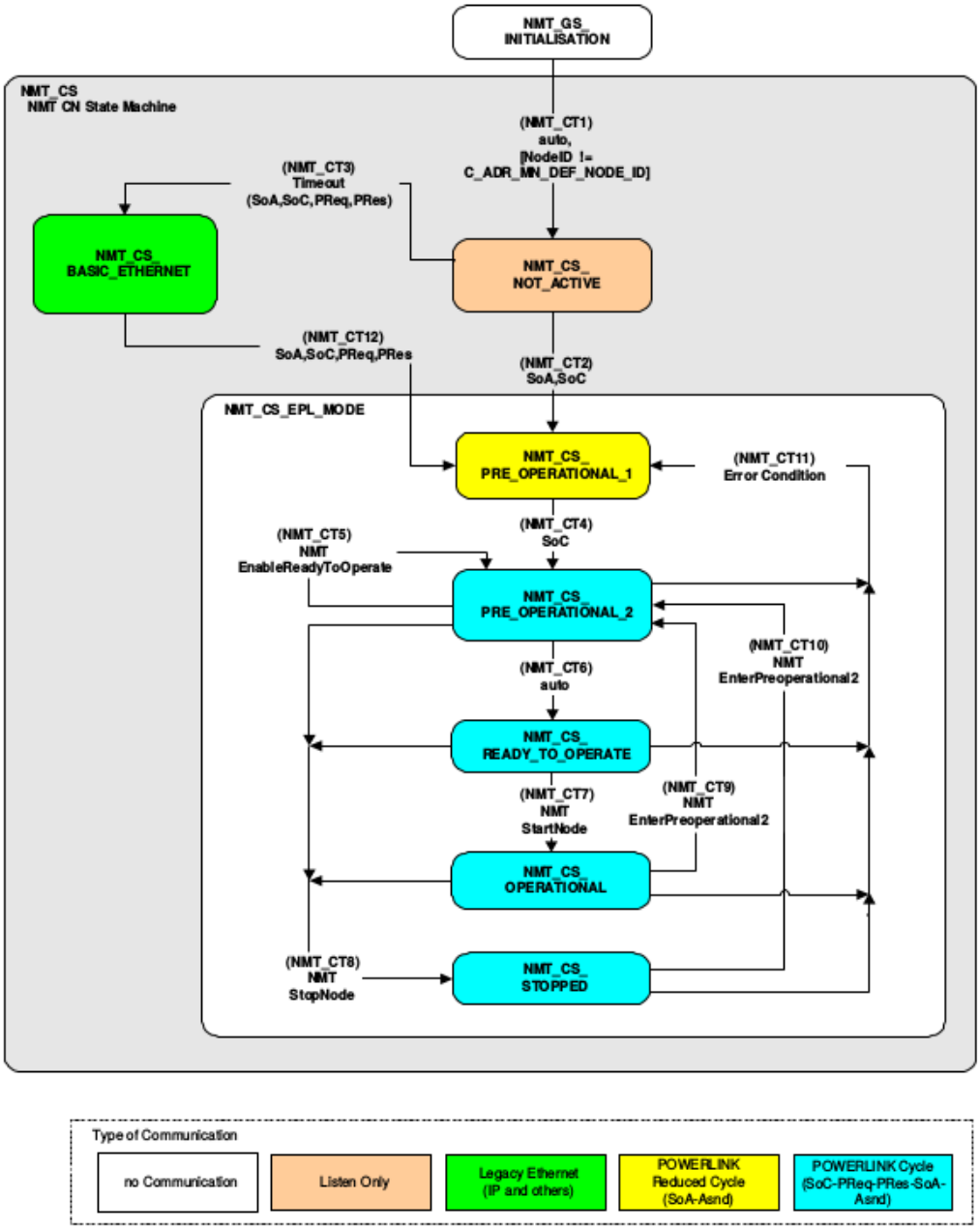


Figure 2.13: NMT State Machine behaviour to keep the CN stations in the correct operation state.

Chapter 3

The IEEE 802.11 Standard

The 802.11 wireless LAN (WLAN) standard of Institute of Electrical & Electronics Engineers (IEEE) actually specifies a family of standards for wireless local area communication. The first version of IEEE 802.11 has been released in 1999, then several updates have been published over the last years. Between different versions of the standard, the IEEE 802.11 task group releases several amendments, such as IEEE 802.11b, IEEE 802.11g and IEEE 802.11n. The IEEE 802.11 WLAN represents an interesting opportunity for real-time industrial communication since, besides the known advantages of wireless networks, it can provide satisfactory performance for a wide range of applications. Especially, the IEEE 802.11n standard has proved to be an effective solution to the communication problems typical of Industrial Wireless Networks (IWNs), where tight constraints in terms of both timeliness and reliability are often encountered [22]. The 2012 release of the Standard [6] is the currently used, and it is also will use in this thesis.

The scope of the IEEE 802.11 Standard Part 11 [6], as well as the IEEE 802.3 Std (Ethernet), is to define one medium access control (MAC) and several physical layer (PHY) specifications, so as to supply wireless connectivity for fixed, portable, and moving stations within a local area.

3.1 Architecture of a WLAN

In the design of wired LANs an address is always equivalent to a physical location, on the contrary, in WLANs this is not always the case. In IEEE 802.11 Std the basic element of a wireless network is an unit, named Station (STA), which is related to an address. Hence, a STA is a message destination, but not (in principle) a fixed location. There exist the fixed STA, the portable STA and the mobile STA; moreover, a STA might take on multiple distinct characteristics, each of which shapes its function: e.g. a quality-of-service (QoS) STA, a dependent STA or a hidden STA.

Turning to the network topologies, there are two possible configurations: the *Infrastructure Network* and the *Ad-Hoc Network*. The *Infrastructure Network* includes both the Basic Service Set (BSS), i.e. several STAs connected to an Access Point (AP), and the Extended Service Set (ESS), where two or more BSSs are linked together with a Distribution System (DS). It is worth pointing out that, in the Infrastructure Networks STAs can not directly communicate each other, therefore, a device must allow the communication. In detail, the AP allows the connection between BSSs or between STAs, instead, the portal allows the connection between a WLAN and another IEEE 802.x network.

The alternative is the *Ad-Hoc Network*, more precisely called Independent BSS (IBSS), which is composed by two or more STAs connected together without any AP. Figure 3.1, taken from [23], gives an idea of three possible Service Sets.

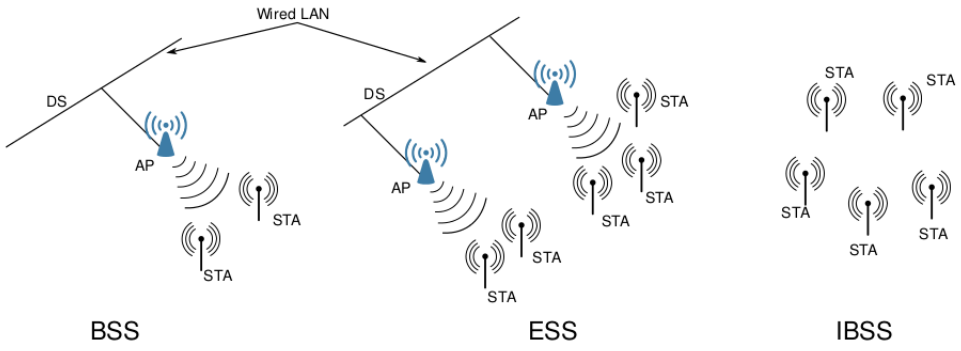


Figure 3.1: The three possible Service Sets defined in the 802.11 Std.

3.2 Medium Access Control Layer

The Medium Access Control (MAC) defines the provided services, the devices behaviour and strongly determines the performance of the network. The Std claims that the transmission medium can operate in the contention mode exclusively, requiring all stations to contend for access to the channel for each packet to be transmitted.

3.2.1 Access to the medium

The 802.11 Std defines a medium access algorithm alternative from the 802.3 one. This diversity is mainly due to the lack of two features in the wireless networks, with respect to Ethernet. Indeed, in a wired network firstly all nodes see each other, hence they know exactly how many stations are involved in the communication; secondly a STA is unable to listen to the channel for collisions while transmitting.

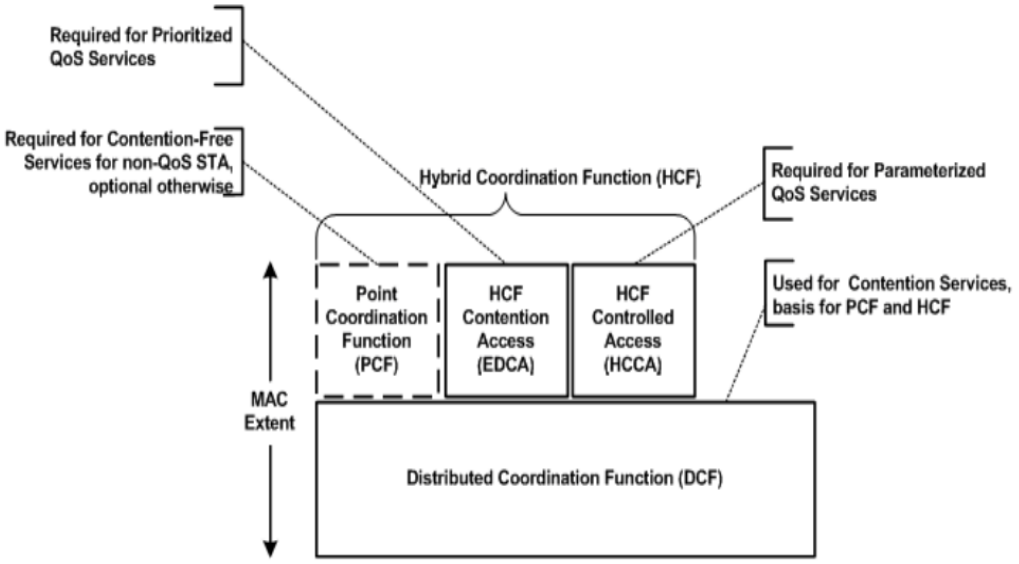


Figure 3.2: MAC architeture.

The devices compliant with the 802.11 Std are controlled by a Coordination Function (CF), an algorithm which determines when a STA is allowed to transmit over the medium. Any station that belongs to an IEEE 802.11 network implements the Distributed Coordination Function (DCF), while there are some other CFs that may or may not be implemented, namely the Point

Coordination Function (PCF) and the Hybrid Coordination Function (HCF). Figure 3.2 depicts the general architecture of the IEEE 802.11 MAC, distinguishing between the different types of CFs. As Figure 3.2 shows, the DCF sits directly on top on the physical layer and serves as the basis for both the others, indeed it offers the common access method known as CSMA/CA.

DCF

The DCF is the standard channel access mode of the IEEE 802.11 protocol and its implementation is mandatory for all devices which are compliant to the standard. The standard allows DCF to work according a basic procedure or a more complex one. In the present work, only the basic Distributed Coordination Function (DCF) has been considered, since it is the most widespread coordination function available. The basic DCF, essentially, provides the channel access method: the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

For IEEE 802.11, time is slotted in time periods that correspond to a Slot-Time, t_{slot}. Unlike slotted Aloha [24], where the slot time is equal to the transmission time of one packet, the Slot-Time used in IEEE 802.11 is much smaller than the duration of a MPDU[1] and it depends on the physical layer implementation.

The DCF key features are: the channel sense of the transmission medium, the inter frame spaces (IFS), the backoff time and the acknowledgment.

- The *channel sense* can be performed both through "virtual" mechanisms and physical ones, however, in the basic DCF only the latter is used and it is simply provided by the PHY layer. Physical carrier sensing detects the presence of other IEEE 802.11 WLAN users by analyzing all detected packets, and it also detects activity in the channel via received signal strength from other sources.

- In the CSMA/CA procedure, predefined intervals of time, called *inter frame spaces* (IFS), are adopted. An IFS interval is a mandatory period of idle time on the transmission medium. The standard defines six different IFS (RIFS, SIFS, PIFS, DIFS, AIFS, EIFS) to allow different priority levels to access the medium. Generally, the smaller it is the duration of an IFS, the higher the priority of the operation relying on that time is. Figure 3.3 shows the relationships between the different IFSs and their length hierarchy.

[1] MAC Protocol Data Unit, the minimal packet that can be sent.

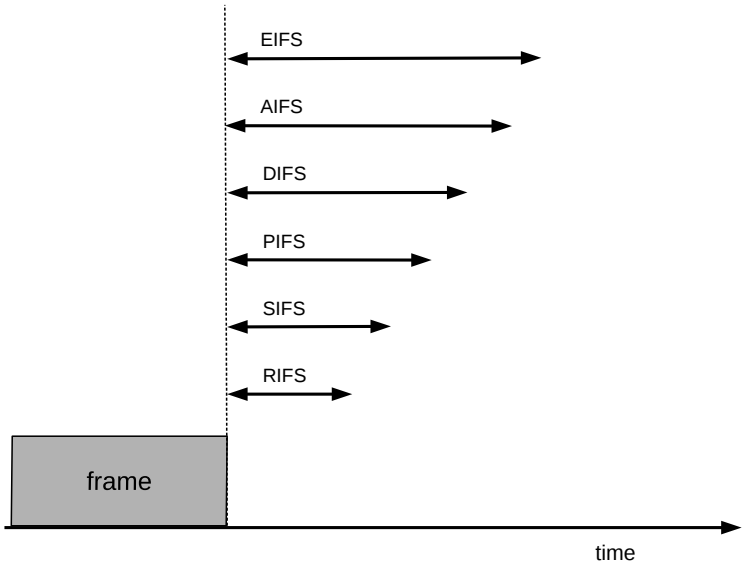


Figure 3.3: Relationships between different IFSs.

- When a collision is detected or when a STA senses the channel as idle after the end of a busy period, a random period is waited in order to avoid that different STAs transmit data simultaneously. This period, called backoff time, is computed as:

$$T_{BackOff} = random \cdot t_{slot} \tag{3.1}$$

where: *random* is a pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$. CW is a parameter, called Contention Window, bounded by the two values CW_{min} and CW_{max}, both related to the PHY layer used. To begin, the initial value of CW is set to CW_{min}, then, each time a transmission failure is detected, it is updated to $2CW + 1$. In case of several consecutive failures, the update process continues until the CW_{max} value is reached, then the CW value remains constant. Figure 3.4 describes the exponential increase of CW with the number of consecutive transmission attempts, assuming $CW_{min} = 7$ and $CW_{max} = 255$. At the first successful transmission, CW is restored to CW_{min} value. The update process relies on an internal counter kept by each STA, namely the STA Short Retry Count (SSRC), which keeps track of consecutive transmission failures.

- Another important feature of DCF is that each transmission of a data frame must be acknowledged, which means that a specific frame, called

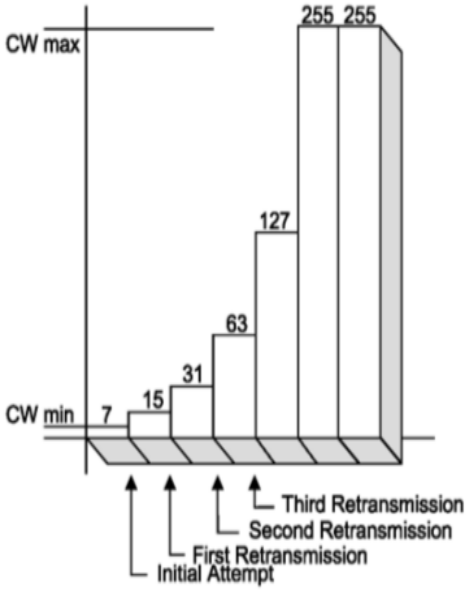


Figure 3.4: Exponential increase of the CW counter.

Acknowledgment (ACK) frame, must be sent by the receiver to the transmitter upon the successful reception of the data frame. The acknowledgement mechanism is used by STA to verify the success of transmissions and detect the presence of collisions. The period between the completion of packet transmission and the sending of the ACK frame is called SIFS.

For the basic access DCF, based on the CSMA/CA technique, a brief description follows. A station wishing to transmit listens to the channel for a DIFS interval to determine if another station is transmitting. If the medium remains free for all the duration of a DIFS, then the station is allowed to access the channel and starts its transmission. The receiving station calculates the checksum and determines whether the packet was received correctly. Upon receipt of a correct packet, the receiving station waits a SIFS interval and transmits a positive acknowledgment frame (ACK) back to the source station, indicating that the transmission was successful. Figure 3.5 reports a timing diagram illustrating the successful transmission of a data frame.

Otherwise, if the channel is sensed as busy, the CSMA/CA procedure is summarized as follows and it is shown by Figure 3.6, taken from [25]. The STA defers its transmission and it waits until the channel becomes idle for a DIFS period, then STA takes part in contest computing a random backoff time. In other words, the STA waits until the current transmission stops, and after that, it listens again the channel and checks if it remains idle for

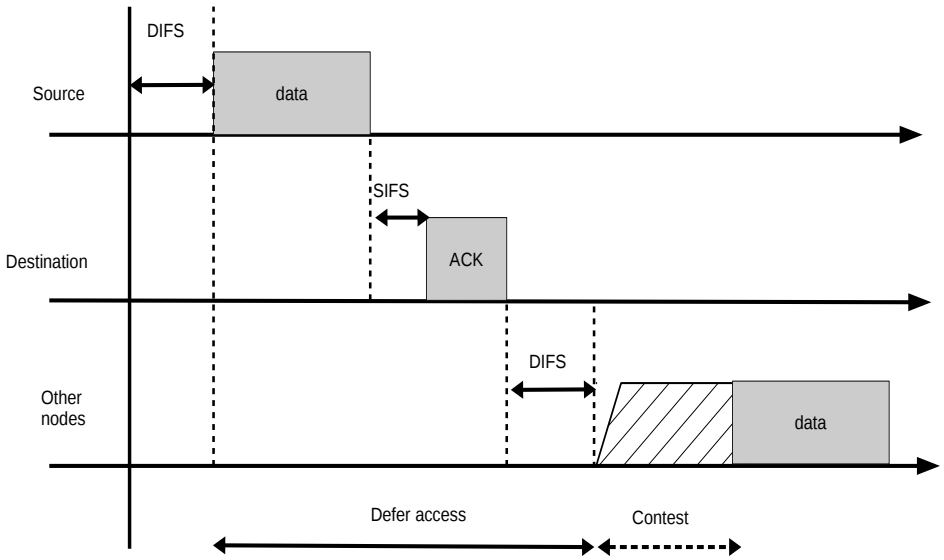


Figure 3.5: Time diagram of the transmission procedure using DCF.

the duration of a DIFS. Finally, the STA calculates a random backoff timer, during which it listens the channel, and when the timer expires, if the channel is still idle the STA is allowed to access the channel and starts its transmission. This procedure prevents multiple stations from gaining access to the medium immediately after the completion of the preceding transmission, ensuring no collisions.

The paper [26] shows that if a station has an exact knowledge of the network status and load configurations (i.e., number of active stations and length of the message transmitted on the channel), it is possible to tune its backoff algorithm to achieve a protocol capacity very close to its theoretical bound.

PCF

The PCF represents an optional medium access technique, and it is based on a polling procedure executed by a station referred as Point Coordinator, it is usually the AP. 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 [3].

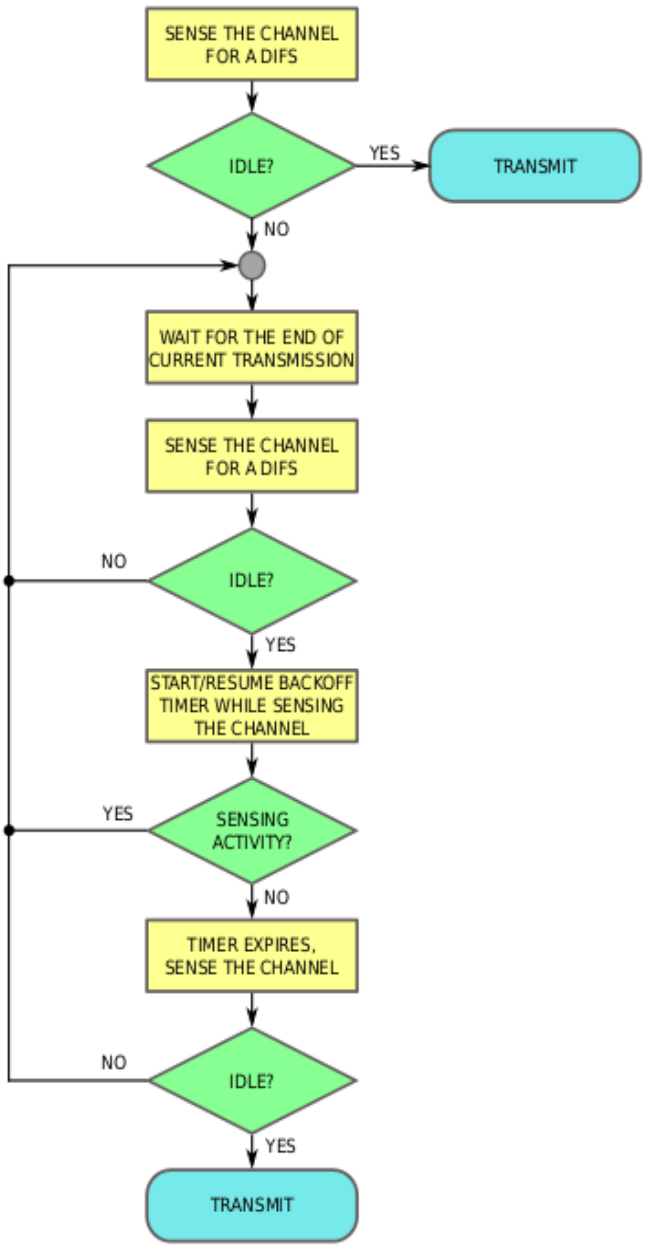


Figure 3.6: Flow diagram of the channel access procedure with DCF.

HCF

The IEEE 802.11 WLAN standard also provides the possibility of adopting a quality-of-service (QoS) mechanism in data transmission. 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). Although the analysis of the HCF may be interesting, it will not be considered further since we are mainly interested in the DCF.

3.2.2 Fragmentation of frames

The MAC layer of the IEEE 802.11 standard provides the possibility of fragmentation and defragmentation of information units. The initial information unit can be a MAC Service Data Unit (MSDU) or a MAC Management Protocol Data Unit (MMPDU), depending on its content (data or management information), while the frame actually sent over the channel is called MAC Protocol Data Unit (MPDU). The fragmentation process can be applied only to unicast frames. This feature will not be discussed in more detail because in the industrial communication field frame sizes are very limited, hence this features is not useful.

3.2.3 Frame structure

Each MAC frame consists of the following basic components: the *MAC header*, a variable-length *frame Body* and the *Frame Check Sequence* (FCS). As can be observed in Figure 3.7, the header has a more complex structure than the one of the Ethernet frame. Indeed in the header there is all the information needed to handle the access medium procedure above described.

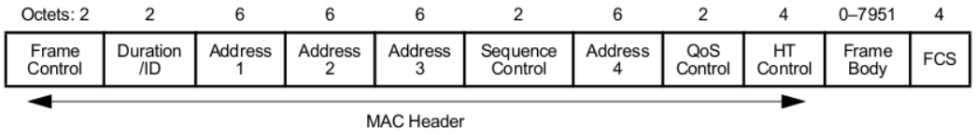


Figure 3.7: Structure of a 802.11 frame at MAC layer.

The first three fields (Frame Control, Duration/ID and Address 1) and the last field (FCS) together make up the minimal frame format, so they are present in all IEEE 802.11 MAC frames, while all the other fields are present only in some types and subtypes of frames. In general, frames could be separated into three topologies:

- *Management frames*, also called MMPDU: they are used by Wireless stations to join and Leave the BSS.

- *Control frames*: they assist with the delivery of the data frames, they must be heard by all the stations, therefore they must be transmitted at one of the basic rates. Moreover, they are also used to clear the channel, acquire the channel and provide the unicast frame acknowledgements (ACK). These frames are the shortest, indeed they contain only header information.

- *Data frames*, also called MSDU: they carry actual data that is passed down from higher layer protocols.

3.3 Multirate support

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. Multirate Support (MRS) is a feature offered by the IEEE 802.11 Wireless LAN (WLAN) standard to improve system performance. Basically, MRS allows a station to dynamically select the transmission rate for a forthcoming packet with the aim of increasing the chance of successful delivery, by better adapting to channel conditions. Indeed, MRS relies on the fact that lower rates adopt more robust modulations, and are hence able to ensure higher transmission success probabilities even under low signal-to-noise ratio (SNR) conditions, which is proved in [23]. The IEEE 802.11 standard does not define any rate selection algorithm, leaving the practical implementation to manufacturers of compliant devices, but in order to provide coexistence and interoperability on multirate-capable PHYs, the IEEE 802.11 standard defines a set of rules to be followed by all STAs.

This has led to the design of different Rate Adaptation (RA) strategies which revealed not effective in the industrial communication scenario, as assessed in [27], [28], since the design choices were mainly targeted at network throughput maximization, while performance indexes of prominent importance for real-time industrial communications, such as timeliness and reliability, were not addressed.

3.4 IEEE 802.11n

In 2009 the IEEE 802.11n amendment, referred to us as High Throughput, was released, providing several improvements to the previous versions at both the physical and data link layers. Figure 3.8, taken from [29], shows the relevant features introduced at the MAC layer by IEEE 802.11n amendment. However, only some of its features reveal actually useful in the context of industrial

wireless networks, principally in terms of reliability and timeliness [7]. This is due to the different types of traffic and requirements typical of this scenario. Indeed, industrial communication systems are usually characterized by critical timing and reliability constraints, with the exchange of small-size packets.

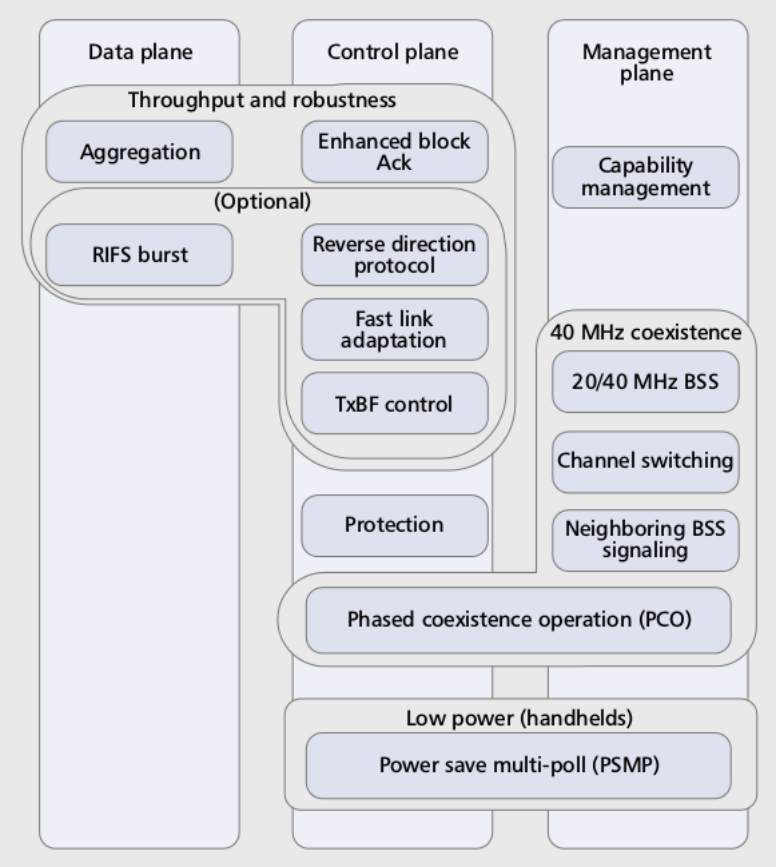


Figure 3.8: Summary of 802.11n MAC enhancements.

The key requirement that drove most of the development in 802.11n is the capability of at least 100 Mb/s MAC throughput [30]. Considering that the typical throughput of 802.11a/g is 25 Mb/s (with a 54 Mb/s PHY data rate), this requirement dictated at least a fourfold increase in throughput. Two basic concepts are employed in 802.11n to increase the PHY data rates: MIMO and 40 MHz bandwidth channels.

Increasing from a single spatial stream and one transmit antenna to four spatial streams and four antennas (both receiver and transmitter) increases the data rate by a factor of four. The term spatial stream is defined in the 802.11n standard as one of several bitstreams that are transmitted over multiple spatial dimensions created by the use of multiple antennas at both ends of

a communications link. In an industrial communication context, however, it is more convenient to use secondary antennas to improve communication reliability through the adoption of Space-Time Block Coding (STBC) techniques, actually implementing a sort of redundancy at the PHY layer [31].

A second substantial improvement is the introduction of 40 MHz bandwidth channels in place of the traditional 20 MHz ones, doubling the typical channel width of previous PHY technologies, thus providing twice the data rate.

Other expedients adopted at PHY layer to improve transmission rates include the use of a reduced guard interval between transmissions (only under certain channel conditions).

In addition to throughput enhancements, IEEE 802.11n introduces also strategies to improve communication robustness (which might also be inherently increased by the use of multiple antennas).

Finally, MAC layer expedients are also introduced, however, they do not bring any significant performance improvement for real-time communication [28].

3.4.1 Optimization of the 802.11n for industrial communication

Nowadays, IEEE 802.11n networks are widely deployed in general purpose communication systems. However, this is not the case for the industrial scenario, where these networks are still rarely deployed. In the literature the enhancements introduced by IEEE 802.11n are evaluated with the aim of defining a set of recommendations for the effective exploitation of IEEE 802.11n in the context of real-time industrial communications, subject to reliability and timeliness constraints.

To begin, the performance analyses of the 802.11n protocol pursued in [31], [32] reveal that significant benefits can be achieved by suitably exploiting the MIMO architecture defined in the standard. To this regard, an effective MIMO configuration is represented by the reduction of the number of independent transmission streams, thus sacrificing the maximization of the throughput, in favour of the adoption of the Space Time Block Coding (STBC) to enhance communication reliability.

Moreover, due to the intrinsic unreliability of the wireless medium, the current research efforts aim at improving both timeliness and reliability of such a protocol in view of its adoption for real-time applications. A significant issue in this context is represented by the reduction of the randomness that affects packet delivery times. An important benefit in this direction can be obtained

by the deactivation of the standard legacy carrier sensing and backoff procedures. The simulation outcomes discussed in [33] show that the deactivation of CS and backoff procedures allows to decrease the number of failed pollings, which reflects in an increased system reliability as well as in an improved timeliness.

Finally, the papers [34],[8] propose the Rate Selection for Industrial Networks (RSIN), an algorithm designed to optimally select the frame transmission rate in IEEE 802.11 industrial networks according to channel conditions. This innovative RA technique is based on the following main features:

- dynamic identification of the channel status, exploiting device-measured Signal to Noise Ratio (SNR) levels;
- rate selection based on a constrained minimization of the packet error rate;
- knowledge of the deadline on packet delivery time.

RSIN is based on two main assumptions:

1. Firstly, in any data exchange between two stations, each packet has to contain an additional field in which the transmitting node inserts the perceived SNR relevant to the last received packet from the other node.
2. Secondly, any transmitting node is aware of the relationship between the Packet Error Rate (PER) and SNR for any possible transmission rate.

The RSIN technique is defined as an optimization problem. Given a packet to be transmitted with a deadline D, and a specific transmitter-receiver pair, the problem can be formulated as to find the number of attempts and the relevant sequence of rates to be used for the transmission of that packet, with the twofold goal of minimizing the residual transmission error probability, while ensuring the packet is delivered within its deadline [8].

3.5 POWERLINK Wireless Extension

Wireless networks represent an interesting opportunity for real-time industrial communication thanks to the benefits they are able to bring. Among the set of available network standards, the IEEE 802.11 WLANs revealed effective since it is able to provide satisfactory performance, in particular in terms of timeliness and reliability. Indeed, the IEEE 802.11 has an high transmission speed which, in general may allow to maintain the real-time behaviour

of an industrial application. The use of IEEE 802.11 is mainly employed as a natural extension of Ethernet segments. Thus, IEEE 802.11 networks may be effectively employed to implement wireless extensions of already deployed wired communication systems, such as the POWERLINK protocol. The main advantage of a wireless extension of the EPL protocol is the possibility of connecting components that cannot be reached by cable (e.g. mobile components such as robots, cranes). Moreover, another important advantage is related to the cost, indeed a wireless connection, when it is made properly, decreases costs for cables installation and maintenance.

In a POWERLINK communication a centralized architecture is set-up, in which a controller cyclically polls its attached nodes under tight timing requirements, which in turn also imply that packet jitter is kept below a specific threshold. As it is well known, the most critical aspect of a wireless communication is represented by an high transmission failure probability, which might in some cases bring the reliability and robustness of the network to unsatisfactory levels. The transmission success probability is typically boosted exploiting retransmission schemes in which random times (backoff slots) are waited between two consecutive transmission attempts. In practice, the reliability of the channel is increased at the price of random delays in data packets delivery. Such delays may severely affect jitter and even compromise real-time operation of the factory automation control system. Therefore, a particular care is necessary in the design of protocols for real-time industrial wireless network, in order to adhere to the maximum threshold for packet jitter using a wireless protocol. Moreover, accurate analysis are also needed to prove the effectiveness of wireless solutions in POWERLINK applications, both through numerical simulations and experimental measurements on real devices.

3.5.1 POWERLINK and 802.11n

The analysis results obtained in [3] and [14] have shown how the performance of a traditional EPL network are influenced by the introduction of the wireless extension. Obviously, the cycle times of the hybrid networks revealed greater than those achievable with the wired ones. However, the main problem is a residual jitter which affects the isochronous period. In particular, it is due to the the intrinsic randomness of the IEEE 802.11 network as well as to the effects of interference and fading.

In the paper [14], the main problem is the retransmission procedure of IEE 802.11. In fact, if during the query of a wireless CN, the MN does not receive the PRes frame within the Poll response time-out, then it moves on the next

station. However, if the N_{max} (the maximum number of retransmission) on the node that issued the relevant IEEE 802.11 frame is not reached, then that node will continue to transmit, possibly exceeding the EPL polling timeout. The forthcoming and devastating consequence is the collapse of the EPL communication. Therefore, the retry limit has to be set in such a way that, when it is reached, the EPL timeout has not to be expired yet.

The reduction of the retry limit revealed, in the simulations done in [14], to be effective since no transmission attempts by a station were observed after the expiration of the EPL time-out. Indeed, in this situation the jitter is reduced and hence it may be tolerated by several application where the precision required is not very critical. Therefore, undertaking a thorough investigation in this direction in this thesis work looks an appropriate choice.

The use of the new 802.11n amendment [6] and in particular, its industrial optimizations done in literature, described in Section 3.4.1, open doors to explore new possibility to improve the wireless extension of POWERLINK. Especially, the "Soft-MAC" wireless card compliant with 802.11n allows to verify on the hardware the possibility to mitigate the effects of retransmission by reducing the retry limit.

Within this framework, this paper prosed an EPL wireless extension based on the IEEE 802.11n, optimally tailored to the industrial communication through the dynamic rate adaptation algorithm RSIN, and on a Linux bridge, which realizes the interconnection between the wired and the wireless segment.

3.5.2 POWERLINK and RSIN

The Multirate Support option led to the development of several rate selection algorithms which can improve the IEEE 802.11 performance. Particularly, among all the algorithms which can be found in literature, in this work of thesis we choose the RSIN, described in Section 3.4.1, because according to the analysis done in [8] it is able to outperform all the other RA algorithms in terms of both reliability and timeliness. Hence, undertaking a thorough investigation in the use of RSIN for the wireless segment of a POWERLINK communication looks as an appropriate choice.

In a basic EPL communication, where no complex configuration of communication such as PollResponse Chaining, Multiple Asynchronous Send or Multiplexed station setup are used, only five basic frames are exchanged, i.e. SoC, PReq, PRes, SoA and ASnd. It is reasonable to presume that all frames

have approximately the same size, moreover the same assumption is often used in the literature such as [12], [3], [15].

For the case in which all frames share the same length and are subjected to the same deadline, an alternative, more convenient, solution w.r.t the complete algorithm RSIN is devised in [8]. In particular it consists in the following operations: the WNIC may initially (off-line) execute the RSIN algorithm to build a look-up table where the final rate sequence is stored for each possible value of the SNR. Consequently, the selection of the suitable rate sequence to be used for the transmission of a packet simply reduces to a search procedure within the look-up table, with a considerable reduction of the computational burden.

3.5.3 Hybrid networks

The wireless extension of a communication system, such as EPL, is composed by the following elements:

1. the controller, i.e. a station which is responsible of the traffic schedule, it is usually connected to a wired segment.

2. one wireless segment with few stations connected to it, typically with limited geographic extension.

The interconnection between the wired and the wireless 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.

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. In this work, we will implement only bridge based solutions.

3.5.4 Bridging IEEE 802.3 and IEEE 802.11

A network bridge is a device commonly used to connect different networks segments together, so that there will appears as one extended LAN to the participants. The IEEE 802.1D MAC Bridges standard [35] claims that IEEE 802 LAN of all types can be connected together using MAC Bridges. Therefore, a bridge can also connect a wireless interface running in hostap mode to a wired network and act as an access point.

A bridge is supposed to be a transparent entity towards the network, thus it automatically initializes, configures itself and runs with no intervention from the stations. Since bridges operate below the MAC service boundary, they are transparent to protocols operating at, or above, the logical link control sublayer. Each of the network segments being connected, wired or wireless, corresponds to one physical interface in the bridge. These individual segments are bundled into one bigger ('logical') Ethernet, this bigger Ethernet corresponds to the bridge network interface.

Each bridge has a number of ports attached to it. Network traffic coming in on any of these ports will be forwarded to the other ports transparently, so that the bridge is invisible to the rest of the network. In particular, frames are forwarded or discarded based on a comparison of the frames' destination address to the information contained in the forwarding data base. Figure 3.9 shows the path of a packet from Host A to Host B using a network bridge.

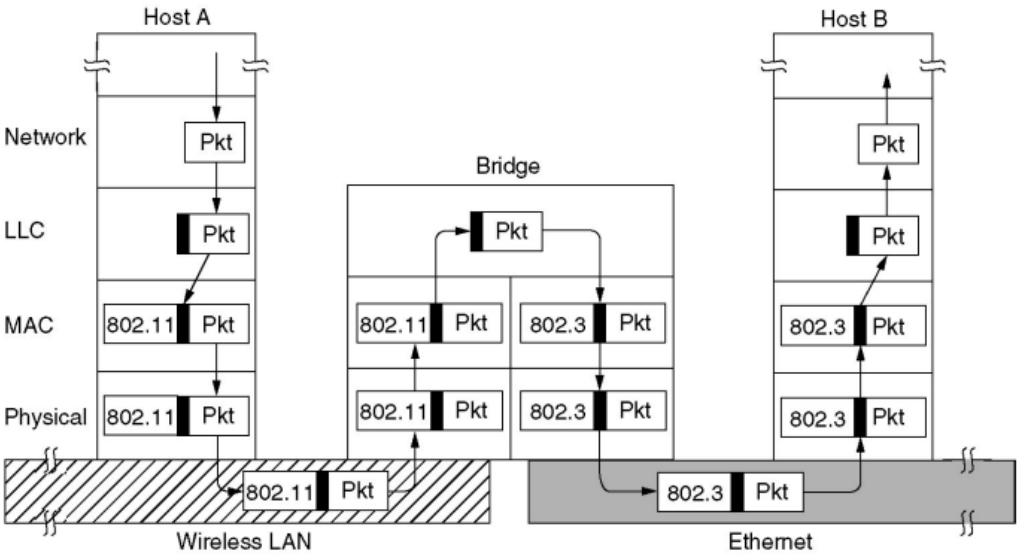


Figure 3.9: Bridging procedure between a WLAN station and an Ethernet one.

3.5.5 Bridge practical implementations

A real bridge is not a transparent ideal device, instead it introduces a delay, which is mainly due to two causes: latency and queueing.

The latency is the time among the instant in which a frame enters the bridge and the instant in which the converted frame starts exiting the bridge.

Moreover, a frame which arrives at the bridge may be queued if other frames are waiting to be transmitted. In particular, queueing represents a random delay directly related to general characteristics such as, for example, network configurations, traffic profiles, etc.

Clearly, 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.

Both [13] and [36] address the topic of hybrid (wired/wireless) industrial networks, in which the interconnection between wired and wireless network segments is done by an *Access Point* (AP). This latter device has a noticeable impact on the timeliness of the whole communication system. In the design of an hybrid industrial network one is much more concentrated in the timely delivery of a packet between the wired and the wireless segment, and the knowledge of the degree of determinism provided by this basic service is a key aspect for an effective implementation of a wireless extension of a POWERLINK network. Unfortunately, AP designers typically do not have such system requirement in mind, so that they do not guarantee nor specify the delays an AP may introduce, even for those devices declared to be specifically designed for industrial usage [36].

In this work of thesis the interconnection between wired and wireless network segments is done by a Linux bridge. Indeed, the Linux kernel contains a bridge module that can be used to create local area networks by combining network interface ports of a computer under a single bridge. While Linux bridges are not able to compete with specialized vendor hardware in performance, Linux bridging can be used in environments where we would like to observe the performance of an 802.11n protocol tailored for such a specific field of application. Moreover, the Linux bridge uses the same clock as the IEEE 802.11 board and, hence, can be easily synchronized with such a board.

The only restrictions of Linux bridge are the following:

- all devices will share the same maximum packet size (MTU). The bridge does not fragment packets.

- Devices must look like Ethernet, i.e. packet must have 6 bytes of source and destination address.

The Linux package `brctl`, available in [37], implements a subset of the IEEE 802.1D standard.

`brctl`, Ethernet bridge administration, is used to set up, maintain, and inspect the bridge configuration in the linux kernel.

`brctl addbr <name>` creates a new instance of the Ethernet bridge.

`brctl addif <brname> <ifname>` will make the interface <ifname> a port of the bridge <brname>. This means that all frames received on <ifname> will be processed as if destined for the bridge. Also, when sending frames on <brname>, <ifname> will be considered as a potential output interface.

Finally `ifconfig <name> up` builds up the bridge, so it can work.

The bridge keeps track of Ethernet addresses seen on each port. When it needs to forward a frame, and it happens to know on which port the destination Ethernet address (specified in the frame) is located, it can 'cheat' by forwarding the frame to that port only, thus saving a lot of redundant copies and transmits. However, the Ethernet address location data is not static data. Machines can move to other ports, network cards can be replaced (which changes the machine's Ethernet address), etc. The aging time is the number of seconds a MAC address will be kept in the forwarding database after having received a packet from this MAC address.

Multiple Ethernet bridges can work together to create even larger networks of Ethernets using the IEEE 802.1d spanning tree protocol. This protocol is used for finding the shortest path between two Ethernets, and for eliminating loops from the topology.

Chapter 4

Hardware and Software

The application analyzed in this thesis consists of a hardware-based system which adopts a POWERLINK communication to perform real-time operations. This is achieved through the B&R devices, relevant software and desktop PCs.
The considered devices of the system are described in detail using the datasheets taken from [18],[1], providing a list of all its physical components and their role in the architecture.

4.1 B&R Devices

4.1.1 X20 CPU

The B&R X20CP-1484 CPU is a device tailored for industrial systems based on PLC and EPL communication, in particular it could operate as Managing Node in these networks. The device, whose frontal view is shown in Figure 4.1, has an Intel processor and an internal memory embedded, but it also requires an initial configuration in order to operate, which is realized in Automation Studio and it is provided by: *USB (IF4 and IF5), Ethernet (IF2)* or *CompactFlash memory*. The user must, therefore, use Automation Studio, a B&R proprietary software, to load the configuration data in the memory devices (USB or CompactFlash) or to activate the transfer of data through Ethernet. In detail, to configure the CPU the user must forward the following data to

the CPU: the application project, realized by the user in Automation Studio, and the Automation Runtime, the software kernel which allows applications to run on a target system.

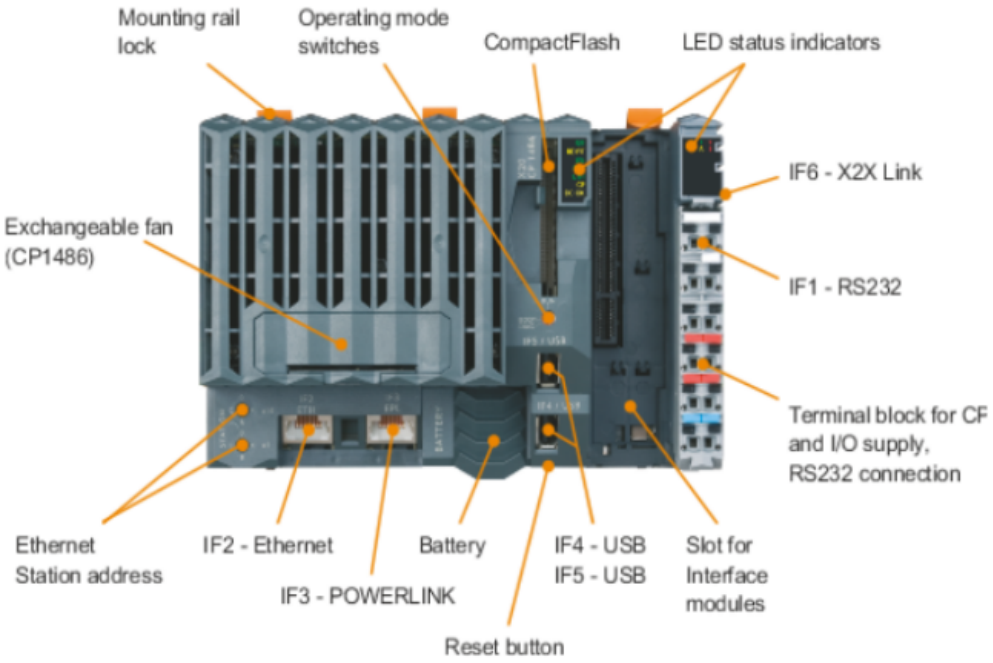


Figure 4.1: Operating and connection elements of the CPU, taken from [1].

As can be seen in Figure 4.1, the CPU has a *POWERLINK interface (IF3)* by which it is possible to link together MN and the CNs so as to accomplish a real-time EPL communication.

The operating mode of the CPU is set using the related *Operating Mode Switch*, in particular there are three possibilities:

- BOOT, the default Automation Runtime is activated.
- RUN, the application project is performed cyclically.
- DIAG, diagnostic mode.

Finally, the station number of POWERLINK interface, can be set using the two *Ethernet Station address* switches, where the MN number is expressed in hexadecimal format.

4.1.2 X20 Bus Controller

The B&R device used as Controlled Node is composed by:

X20BC-0083 : the bus controller base, which frontal view is shown in Figure 4.2. This device has two POWERLINK Interfaces with RJ45 connectors, therefore it could operate as Controlled Node in EPL networks.

X20PS-9400 : the bus controller supply module.

X20Dx-xxxx : terminal block. It is an I/O module equipped with 12 inputs for 1-wire connections.

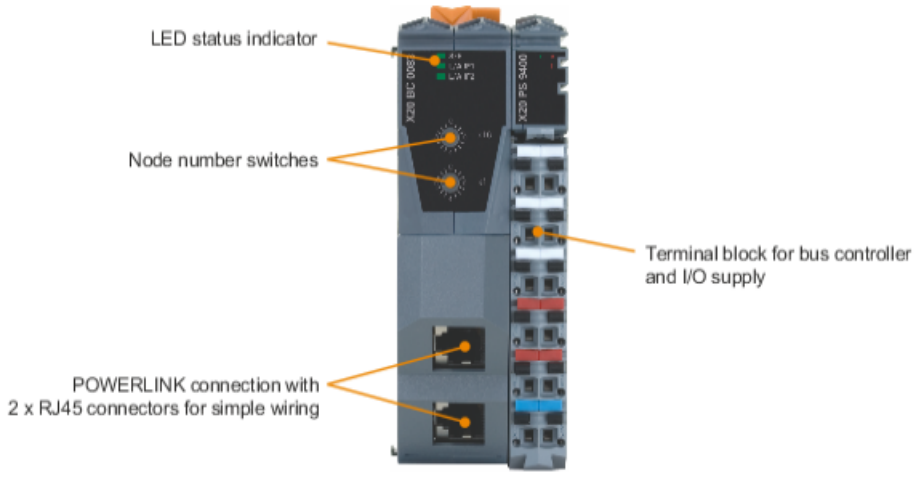


Figure 4.2: Operating and connection elements of the bus controller, taken from [1].

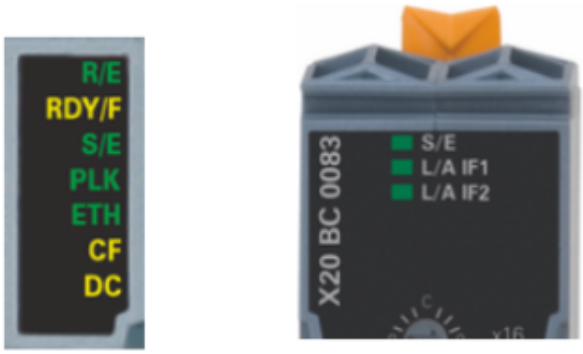
The station number of POWERLINK interface is set using the two hex switches.

4.1.3 LED Status Indicator

Each device has a LED display, namely the *LED Status Indicator*, which shows the operating mode of it. According to the typology of the device there are different LEDs which can be seen in Figures 4.3a and 4.3b.

In particular, the "S/E" Status/Error LED, available in both the devices, is a green/red dual LED and it represents the POWERLINK operating mode of the devices. The "S/E" Status/Error LED could be:

- Red: the module is in an error mode.
- Off: The module can be switches off, starting up, not configured correctly in Automation Studio, defective or it is in NOT_ACTIVE mode.



(a) (b)

Figure 4.3: Status LEDs of CPU 4.3a and bus controller 4.3b, taken from [1].

The bus is being monitored for POWERLINK frames. If a corresponding frame is not received within the defined time frame (timeout), then the MN will not be started, instead the CN switches to BASIC_ETHERNET mode. If POWERLINK communication is detected before the time expires then the MN switches immediately to PRE_OPERATIONAL_1 mode instead, the CN switches to PRE_OPERATIONAL_1 mode.

- Green flickering: the module is in BASIC_ETHERNET mode. The interface is being operated as an Ethernet TCP/IP interface. The MN has to be restarted to change this state, instead the CN changes state if detect a POWERLINK communication.

- Green single flash: the module is in PRE_OPERATIONAL_1 mode. The CN waits for the reception of a SoC frame.

- Green double flash: the module is in PRE_OPERATIONAL_2 mode. The MN begins cyclic communication and the CN can be configured.

- Green triple flash: the module is in READY_TO_OPERATE mode. The MN continue with the cyclic and asynchronous communication and the CN sends its data.

- Green On: the module is in OPERATIONAL mode. Cyclic data is being evaluated.

- Blinking: the module is in STOPPED mode.

4.2 PC

In the test bench there are also three desktop PCs, namely the Dell Optiplex model 745, 755 and 960, running Ubuntu 14.10, kernel Linux 3.16.0 and equipped with a an Intel Ethernet Controller I210 and a Wireless Interface card (WNIC) by TP-LINK. PCs could operate as either MN or CN if they are appositely configured using openPOWERLINK.

I210 It is an Ethernet Controller characterized by the possibility to do hardware timestamp. The timestamp logic is located on transmit and receive paths as close as possible to the PHY interface. The timestamp is captured at the beginning of the packet, in such a way to keep the latency between the captured timestamp and transmission time as deterministic as possible.

WNIC The wireless card is based on Atheros AR9227 chip, fully compliant with IEEE 802.11n. Moreover, it is a "SoftMAC" device, i.e. a card that does not implement the MAC layer in hardware, rather it expects the drivers to implement the MAC layer. Specifically, they are managed by the open source ATH9K Linux driver.

4.3 Hub

Figure 4.4: 0AC808 Ethernet hub, taken from [1].

The B&R 0AC808 Ethernet hub, shown in Figure 4.4, is a Standard Class II hub, moreover it is a device that can be used universally as a level 2 hub

in standard Ethernet or POWERLINK networks. The devices has 8 ports, all are equipped with auto-crossover.

4.4 Software

4.4.1 Automation Studio

B&R Automation Studio is the project development environment used specifically for B&R automation components. The controller, drive, communication, and visualization can all be configured in one environment. The programmed and configured machine parts can then be assigned to different hardware configurations. All relevant IEC 61131/3 languages and ANSI C can be used and also be combined to create projects. The visualization system integrated in Automation Studio is an effective tool that can be used to create line displays as well as control integrated.

Automation Runtime, is an integral component of Automation Studio. This runtime environment offers numerous important advantages:

- guaranteed highest possible performance for the hardware being used;
- hardware independence of application;
- applications porting between B&R target systems.

Figure 4.5 shows the Automation Studio interfaces, a description of the software can be found in [18].

4.4.2 openPOWERLINK

openPOWERLINK is an Open Source Industrial Ethernet stack, a complete protocol solution, implementing the POWERLINK protocol. It is programmed in ANSI C, hence this implementation can be easily ported to any target system. The documentation of the openPOWERLINK protocol stack can be found in [38]. For building openPOWERLINK the build utility CMake [39] is used.

For configuration of a POWERLINK network the Open-Source configuration tool openCONFIGURATOR can be used. The project consists of a core library implementing the configuration algorithms and an Eclipse plugin based user interface.

Figure 4.6 shows the openCONFIGURATOR workspace.

According to the manual [40], openCONFIGURATOR creates four files which can be used by the openPOWERLINK stack and application:

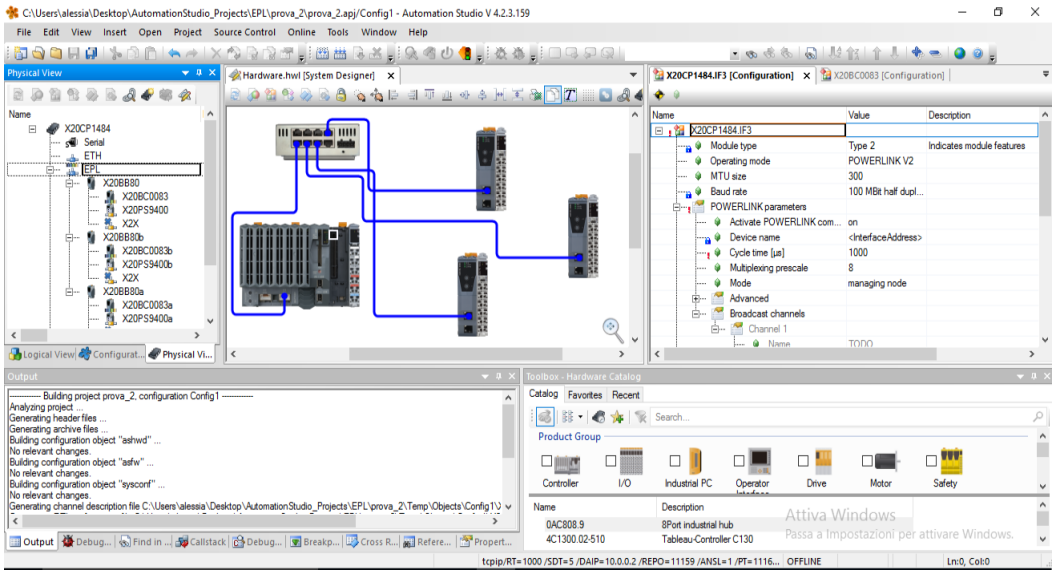


Figure 4.5: B&R Automation Studio workspace.

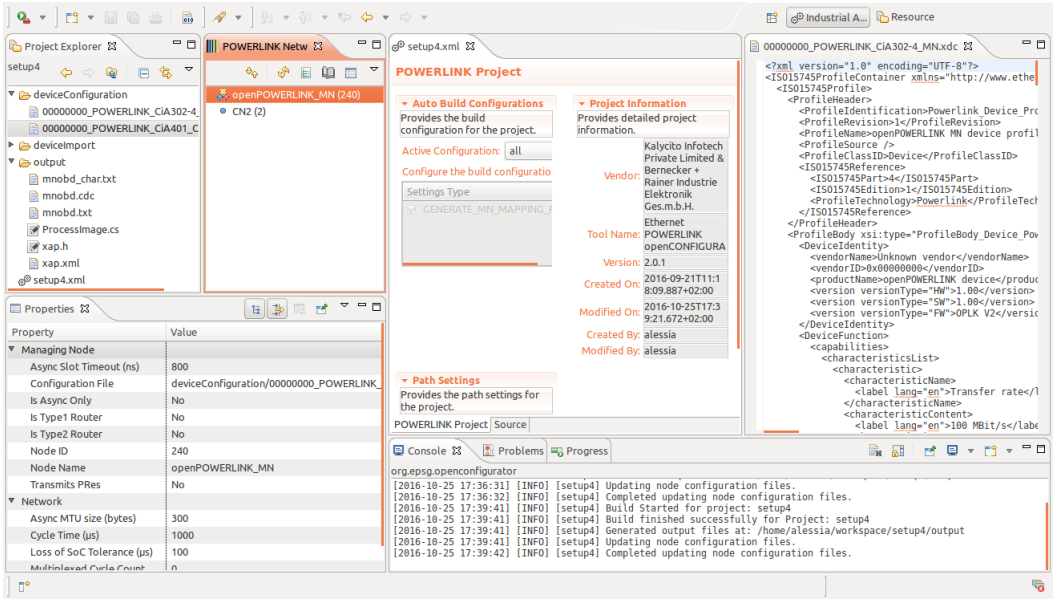


Figure 4.6: openCONFIGURATOR project editor.

- `mnobd.cdc` This file is used to configure the MN stack. It includes all configuration data of the MN and all CNs including the network mapping information.

- `mnobd.txt` This file describes the stack configuration in human-readable

format.

- `xap.xml` The xml file contains the structure definition of the process image. It depends on the available data fields of the CNs used in the application.
- `xap.h` The header file contains structure definition of the process image in the form of two ANSI C structures.

4.4.3 Wireshark & Matlab

Wireshark is a popular network protocol analyzer, its typical interface can be observed in Figure 4.7.

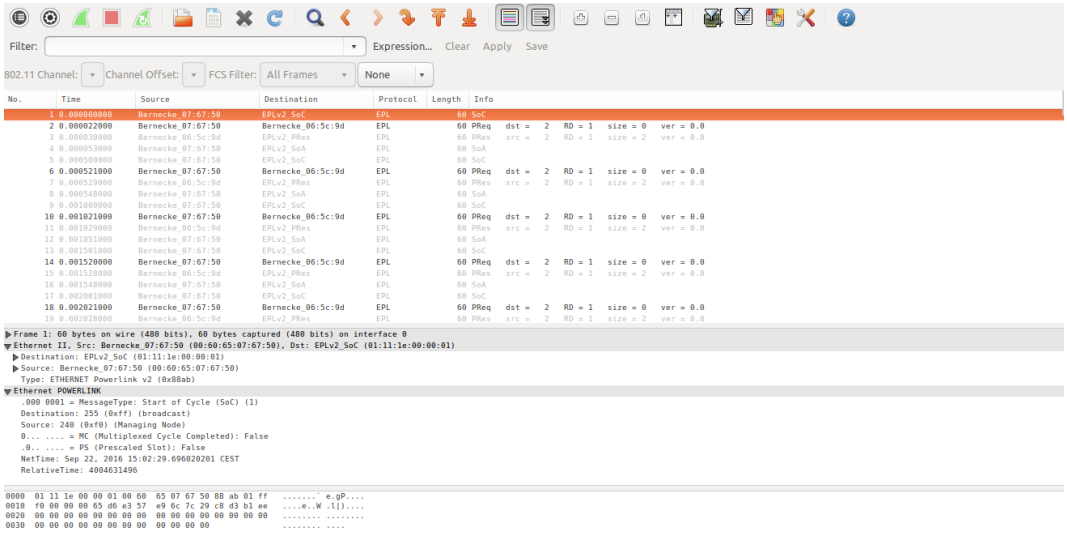
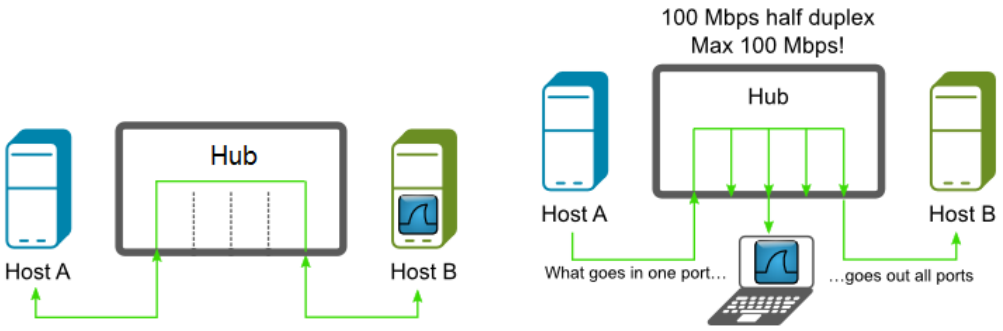


Figure 4.7: Wireshark interface.

Ethernet capture setup It is possible to capture network traffic between the machine running Wireshark and other machines on the network as Figure 4.8a shows, without special setup. However, in this way the traffic among other machines possibly connected to the network is not captured.

Therefore, a method to circumvent this problem is to use an Hub to connect the Ethernet nodes together as Figure 4.8b shows, meaning all packets could be received by all nodes on that network. Therefore, if a machine on such a network is configured into promiscuous mode, it will be able to monitor all the packets transmitted in the network and, hence, to analyze them.



(a) (b)

Figure 4.8: Wireshark configurations: Wireshark on a network machine 4.8a and Wireshark on an external machine 4.8b.

WLAN capture setup Similarly to the Ethernet setup it is posssible capture network traffic between a machine running Wireshark and other machines on the network.

Conversely to the Ethernet setup, to capture all the network traffic, also the one among other machines, the sniffer machine has to be configured in "monitor mode". The sniffer configured in monitor mode captures data, managment and control packets.

Finally, Matlab is used to analyze the data from Wireshark, an example of its typical interfaces can be observed in Figure 4.9.

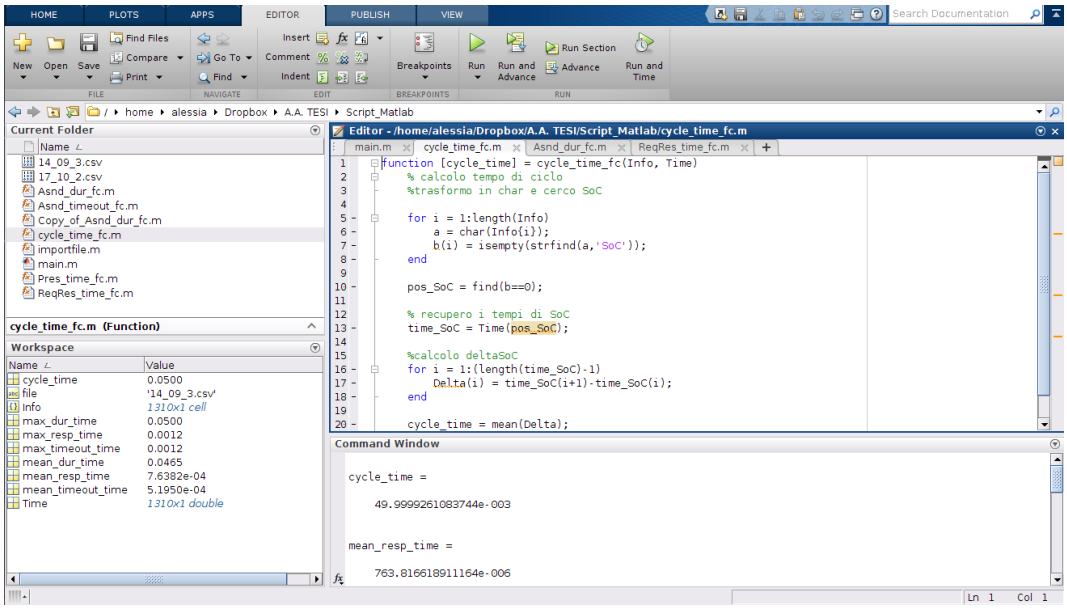


Figure 4.9: Matlab interface.

Chapter 5

Theoretical performance analysis

In order to analyse the behaviour of the experimental configurations and to correctly configure the communication protocol, a theoretical study of the communication system is carried out. The goal of this analysis is to provide an evaluation of the performance that can be reached by the system given its expected behaviour and some hardware-related constraints.

In particular, with reference to the polling procedure in an hybrid EPL network, a precise assessment of the time necessary to query a device is needed to correctly configure the poll response timeout and, hence, to ensure the timely polling of all the nodes. Specifically, we focus on the worst-case scenario, with the goal of estimating a maximum time limit in which the polling procedure is definitely completed, in such a way to ensure a real-time behaviour. Furthermore, the possible sources of variability and randomness, introduced by the IEEE 802.11 protocol, are discussed and appropriate confidence intervals of the parameters are given.

During the off line EPL network configuration phase, the user can set the values of some parameters such as Cycle Time, Poll Response Timeout and Asynchronous Timeout. Consequently, in order to correctly configure an EPL network it is important to carefully choose them according to a theoretical performance analysis.

5.1 POWERLINK Timing

In this section we will consider a wired EPL network, where one MN is connected through an hub to the CNs (see Figure 5.1). To begin, a simple network configuration composed by one MN and two CNs is considered but, whenever is possible, also a formulation for the general case is given.

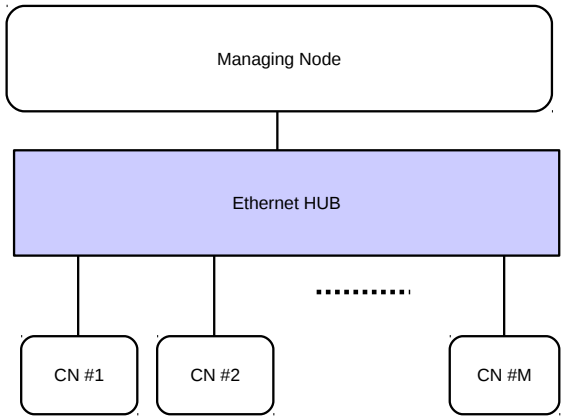


Figure 5.1: POWERLINK network architecture.

In this theoretical analysis we will take into account the times necessary to send the frames as well as the elaboration delays typical of each node. In particular, we will consider an EPL network where:

- all CNs are connected using cables with the same length (e.g. 2 meters).
- All the CNs have the same response time.
- Only minimum size Ethernet frames (i.e. 64 bytes) are supposed to be exchanged, as this is usual for industrial communication applications. Indeed, the size of process data is typically small (few bytes) and the EPL protocol adds only 3 bytes on its own.
- There are ideal operating conditions, i.e. a complete absence of transmission errors so that each transmitted frame is received correctly by the intended destination(s).

5.1.1 Typical Parameters

In the following theoretical analysis several parameters will be used. All the values relative to the B&R devices are taken from the datasheet [17] and from

the Automation Studio documentation [1], the other are taken from the literature, in particular from [3], [15], [13] and [9]. Particularly, we must focus on the worst-case of the ideal operating conditions scenario, to give a maximum time limit which can be used in the off line configuration setup to ensure a real-time behaviour. These parameters are summarize in Table 5.1 and afterwards described.

Symbol	Value	Description
T_F	$5,12\mu s$	Transmission time of an IEEE 802.3 frame (at 100 Mb/s).
T_H	$0,68\mu s$	Hub delay.
T_{sync}	$45\mu s$	Waiting time for the first PReq.
T_P	$10ns$	Cable delay.
T_R^{MN}, T_R^{CN}	$8\mu s$	Node latency of the B&R devices.
T_R^{MN}, T_R^{CN}	$16\mu s$	Node latency of the PCs desktop when configured as POWERLINK devices.

Table 5.1: Worst case values of the parameters.

Transmission time of a frame It is the time necessary to transmit an IEEE 802.3 frame. In our case all frames have the same length of 64 bytes, therefore:

$$T_F = b \cdot t_{tx} \tag{5.1}$$

where b is the length of frames and t_{tx} is the time necessary to transmit one byte ($0.08\mu s$ at a transmission rate of $100Mbit/s$).

Hub runtime Ideally, when an Ethernet frame arrives at one port of the hub, it is passed on to all the other ports at the same time it is being received. In reality, when a frame passes through a hub it is delayed, therefore the hubs has a direct effect on the POWERLINK cycle time.

Initial synchronization The EPL specification indicates $T_{sync} = 45\mu s$ as typical value, where it is included a safety margin (T_W).

Node latency The node latency is the time required to react to an event on the input with a change to the output. This process is composed by several sequential steps which can be observed in Figure 5.2.

Furthermore, in a POWERLINK communication when the CN-i receives the PReq frame from the MN it waits a time T_R^{CN} before sending the

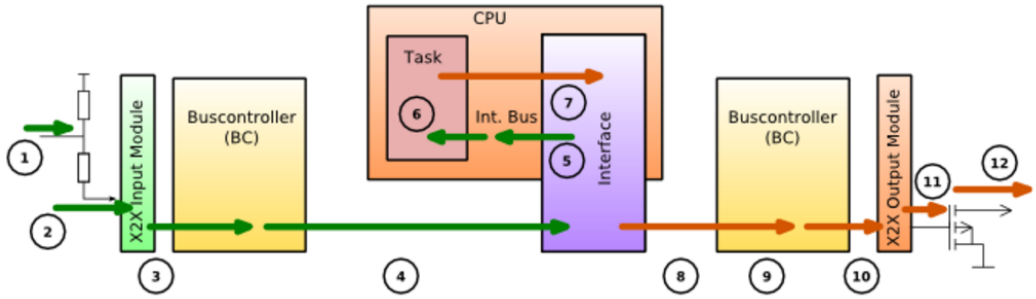


Figure 5.2: The node latency.

PRes. Similarly, the MN waits a time T_R^{MN} before sending the PReq to the following CN-i+1.

5.1.2 Cycle Phase

The cycle time duration is a configuration parameter for the managing node. The EPL cycle time, henceforth T_{cycle} stands for it, represents the maximum total duration of a POWERLINK network communication, in other words it is the period within which the real-time transmission of data between the MN and the CNs using the EPL protocol must conclude. Indeed, it is mandatory that T_{cycle} is not exceeded during the network operation phase. The EPL cycle can be separated into three phases, therefore it is given by:

$$T_{cycle} = T_{iso} + T_{asy} + T_{idle} \tag{5.2}$$

where $T_{iso}, T_{asy}, T_{idle}$ are, respectively, the duration of the isochronous, asynchronous and idle period. The fist term T_{iso}, which represents the isochronous time period, is the only term that can be a priori computed and, in addition, the user must compute it in the off line phase in such a way to correctly setup the EPL network. Conversely, the other terms have not an fixed duration. T_{asy}, which represents the asynchronous time period, depends on the communication needs of the particular network. Instead, T_{idle}, which represents the idle time period, changes depending on the previous phases duration.
In conclusion, this analysis focuses on the isochronous phase, and in particular we characterize T_{iso} for the different configurations that will be considered in the experimental measurements. In this section we focus on the wired configuration, whereas in the next section we will characterize T_{iso} for the wireless configuration.

5.1.3 Isochronous Phase

The duration of the Isochronous phase is, in general, influenced by many different factors such as:

- the network topology, in particular the number of CNs;
- the cyclic data size, which affects the frame transmission time;
- the elaboration delays introduced by the MN, the CNs and the network interconnection devices;
- the transmission/propagation delays.

The duration of the isochronous period is given by:

$$T_{iso} = T_{sync} + M \cdot T_{poll} \tag{5.3}$$

where T_{sync} is the synchronization time, M is the number of CNs and T_{poll} is the polling time of one CN.
During the synchronization period the MN transmits to all CNs a frame where it communicate to all that it is ready to start the EPL cycle of communication. Whereas, during the polling period the MN polls, one by one, each CN in the network by sending a PReq frame and waiting for a PRes.
In order to understand the mathematical formalization of T_{iso} a brief description of the operations involved in the isochronous period follows and furthermore Figure 5.3 shows these operations by a space-time diagram for the basic case considered (one MN and two CNs).
In the synchronization phase a SoC frame, which has a transmission time of T_{SoC}, is sent by the MN through the cable to the hub and then through the cable in parallel to all the CNs, this procedure introduce a delay of $2T_P + T_H$. Moreover, a waiting time, T_W, is required from the EPL protocol.
In the Poll Request activity a PReq frame is sent to the relative CN through hub and cables, the duration of this sequence of operations is $T_{PReq}+T_H+2T_P$. In addition, the CN introduces a waiting time T_R^{CN} due to the CN latency.
Similarly, for the Poll Response activity a PRes frame is sent broadcast through hub and cables. It is worth pointing out that, since the frame is broadcast, the total time for this operation is the same of the Poll Request activity, indeed the broadcast operation is done in parallel, hence there are not further time expenditure. Again, the MN introduces a waiting time T_R^{MN} due to the MN latency.
According to the above observations each term of equation 5.3 is composed by:

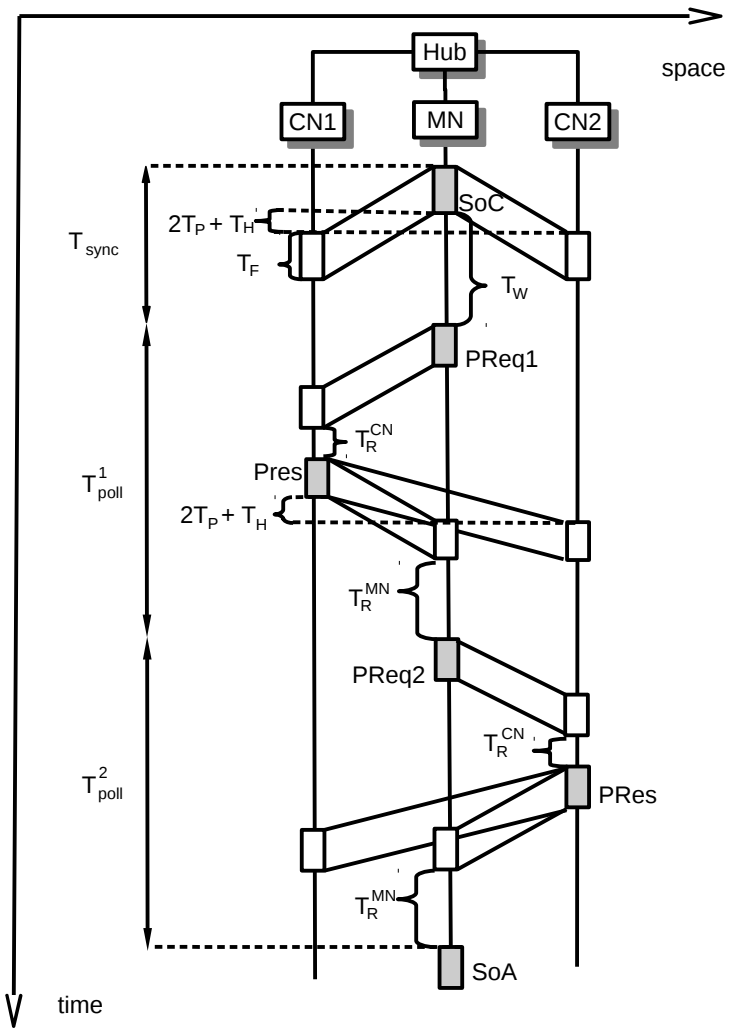


Figure 5.3: The space-time diagram of the isochronous period of an EPL network communication.

$$\begin{cases} T_{sync} = T_{SoC} + 2T_P + T_H + T_W \\ T_{poll} = T_{PReq} + T_{PRes} + T_{delay} \\ \qquad = T_{PReq} + T_{PRes} + [2T_H + 4T_P + T_R^{CN} + T_R^{MN}] \end{cases} \tag{5.4}$$

where: $T_{PReq} + T_{PRes}$ are respectively the PReq and PRes frame transmission time and T_{delay} is the sum of the delays associated to this procedure. In detail, in our case T_{delay} is composed by T_P, the cable propagation delay, T_H, the repeating hub delay, T_W, the waiting time required for all CNs to receive and

process the SoC frame, T_R^{CN}, the CN latency, and T_R^{MN}, the MN latency.

Symbol	Maximum Value
T_{sync}	$45\mu s$
T_{poll}	$44\mu s$

Table 5.2: Worst case values of both the syncronization time and the polling time.

Table 5.2 shows the worst case values of the synchronization period and of the polling time, they are computed using the values of Table 5.1. Therefore, as an example, in the basic case of one MN and two CN the isochronous time has to be set such that:

$$T_{iso} \geq 45\mu s + 2 \cdot 44\mu s = 133\mu s \tag{5.5}$$

In a general network, where there are M CNs, the expressions 5.3 become:

$$\begin{aligned} T_{iso} &= T_{sync} + M(T_{PReq} + T_{PRes} + T_R^{CN} + T_R^{MN}) + 2M(2T_P + T_H) \\ &= T_{SoC} + T_W + M(T_{PReq} + T_{PRes} + T_R^{CN} + T_R^{MN}) + \\ &\quad + (2M + 1) \cdot (2T_P + T_H) \end{aligned} \tag{5.6}$$

Moreover, in the particular case of all frame with the same transmission time T_F, the expression 5.6 becomes:

$$T_{iso} = M(T_R^{CN} + T_R^{MN}) + (2M + 1) \cdot (T_F + 2T_P + T_H) + T_W \tag{5.7}$$

5.1.4 Poll Response Timeout

The response timeout is a configuration parameter for the controlled node. During the polling phase the MN sends a poll request frame, PReq, to the first CN, which responds with a Poll Response frame, PRes. Then, the MN repeats this operation for all the CNs. This process can take variable times depending on several factors:

- Signal runtime between MN and CN: it is determined by the length of the cable and the number/type of hubs between the MN and the CNs.

- Response time of the CN, which is a property of the device used.

In a general network composed by M CNs, the MN manages the polling procedure dividing the available polling time into M slots, one for each CN. In other words, during each slot the MN sends the PReq and waits for the PRes, if no poll response is received within the end of a defined amount of time, i.e. the poll response timeout, the addressed CN is considered to have failed and the MN pass to the next CN. As an example, Figure 5.4 shows the difference between a slot and the PRes timeout in the case of an EPL network with the MN and two CNs. The CN slot is exactly the sum of the PRes timeout and the MN latency. Since the MN latency could be exactly quantified, the PRes timeout is exactly defined.
This parameter is very important for achieve a good communication, indeed:

- a response timeout that is too short causes a POWERLINK station to be considered as having failed, although a response arrives from the station shortly thereafter.
- If a station fails, the POWERLINK cycle continues with the next station as soon as the response timeout expires. If this response timeout is too long, the required cycle time might increase, with the consequent occurrence of a cycle time violation.

Especially, the poll response timeout is measured exactly from the start of the PReq frame transmission to the beginning of the PRes reception. Figure 5.4 shows the space-time diagram of the polling operation of an EPL communication in the basic case considered. As can be observed in the Figure, the poll response timeout is:

$$T_{PRes-timeout} = 2 \cdot [T_H + 2T_P] + T_R^{CN} + T_{PReq} + T_{PRes} \tag{5.8}$$

In this analysis, the poll response timeout is computed taking into consideration the CN with the worst value of poll response, i.e. the one with the highest value of T_R^{CN}. If we consider the particular case of all frames with the same transmission time T_F. The maximum value of $T_{PRes-timeout}$ is:

$$\begin{aligned} T_{PRes-timeout}^{max-th} &= \max_{i=1..M}(T_{PReq-PRes}(i)) \\ &= 2 \cdot [T_H + 2 \cdot T_P] + \max_{i=1...M}(T_R^{CN}(i)) + 2T_F \end{aligned} \tag{5.9}$$

According to the considered parameters in our experiments (see Table 5.1), we have taken $\max_{i=1...M}(T_R^{CN}(i)) := 16\mu s$ because the CN made of a desktop PC has the maximum value of node latency.
Moreover, since the procedure may be influenced by other transmission delays, typical of non ideal operating conditions, in order to ensure the correct operation of the protocol, an appropriate safety margin, T_S has to be added to the

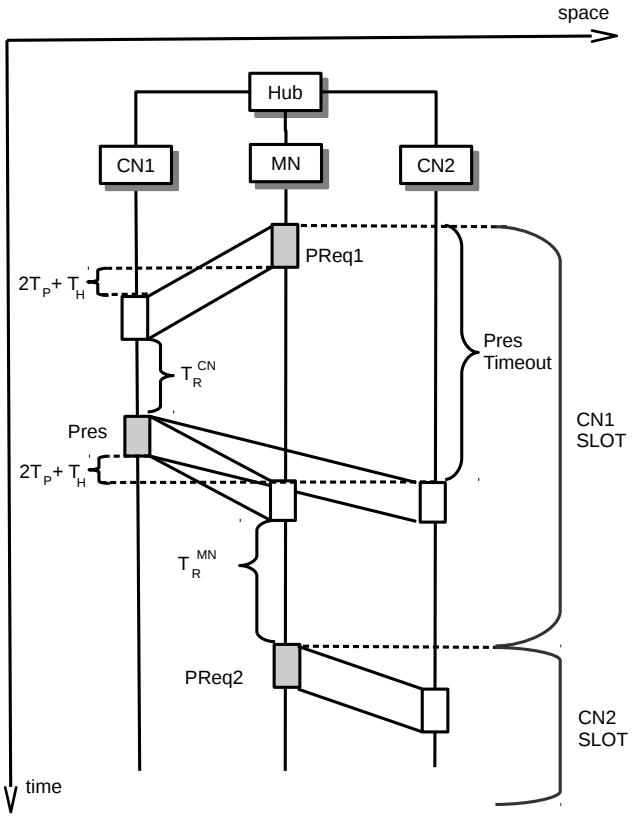


Figure 5.4: The space-time diagram of the polling time procedure of an EPL network communication.

value of the theoretical $T^{max-th}_{PRes-timeout}$ obtained from equation 5.9. We consider as the minimum $T^{max}_{PRes-timeout}$ the value obtained from equation 5.9 increased by 30%, which seems to be a reasonable safety margin. Table 5.3 shows the mathematical worst case value of the poll response timeout resulting from the theoretical analysis.

$$T^{max}_{PRes-timeout} \geq T^{max-th}_{PRes-timeout} + T_S \tag{5.10}$$

Symbol	Minimum Value
$T^{max-th}_{PRes-timeout}$	$23\mu s$
$T^{max}_{PRes-timeout}$	$30\mu s$

Table 5.3: Poll response timeout values, mathematical and with a safety margin.

5.1.5 Asynchronous Timeout

The asynchronous timeout is a configuration parameter for the managing node. During the asynchronous period the MN determines which station can start the asynchronous transmission and includes this information in the SoA frame, no more than one station per cycle is allowed to send asynchronous data. In general, the chosen station could send in the asynchronous period any Ethernet frame, provided that the frame length is less or equal of the MTU size, i.e. the maximum size of the net data of an asynchronous POWERLINK frame. This parameter can be selected in the offline phase and normally MTU := 300 bytes.

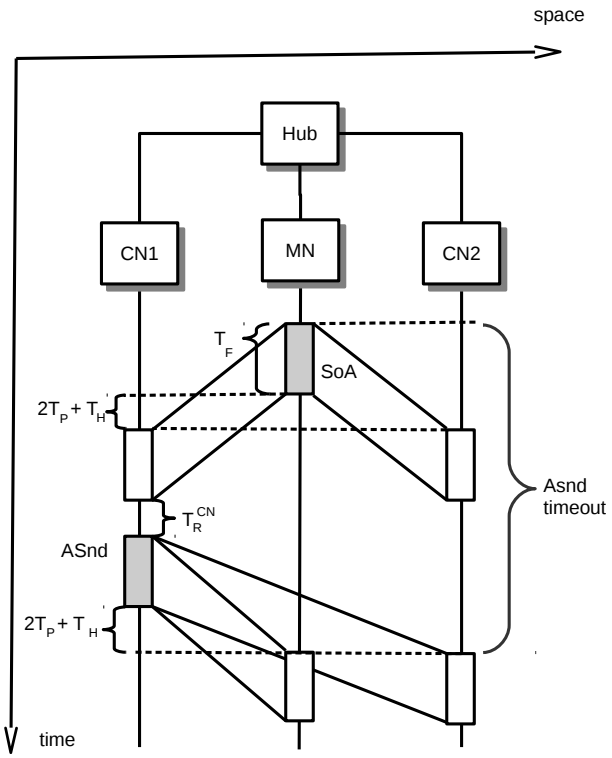


Figure 5.5: The space-time diagram of the asynchronous period of an EPL network communication.

To make it possible to control the cycle time and to calculate the duration CycleIdleTime in the managing node, the maximum runtime for asynchronous communication should be set, namely the Asynchronous timeout parameter. This may be necessary for network ranges over 10 hub levels. Figure 5.5 shows the space-time diagram of the asynchronous period of an EPL communication

in the basic case considered. The asynchronous timeout is calculated like the poll response time, indeed it is measured from the start of the SoA frame to the reception of the ASnd frame. For the configuration with a single hub we have:

$$T_{ASnd-timeout} = 2[T_H + 2 \cdot T_P] + T_R^{CN} + T_{SoA} + T_{ASnd} \tag{5.11}$$

where T_{SoA} is the frame which marks the beginning of asynchronous communication and T_{ASnd} is the frame transmitted during the asynchronous communication. The dimension of the T_{ASnd} depends on the communication necessity. In the worst-case:

$$T_{ASnd} = 300bytes \cdot 0.08\mu s = 24\mu s \tag{5.12}$$

For B&R devices, the same CN response times apply. As this parameter is similar for the whole network, it shall be set to at least the highest of all ASnd Timeout values of the stations in the network. Moreover, we consider the transmission time of a SoA frame equal to T_F. The maximum value of $T_{ASndtimeout}$ is:

$$\begin{aligned} T_{ASnd-timeout}^{max-th} &= \max_{i=1..M}(T_{ASnd-timeout}(i)) \\ &= 2[T_H + 2 \cdot T_P] + \max_{i=1..M}(T_R^{CN}(i)) + T_F + T_{ASnd} \end{aligned} \tag{5.13}$$

In the way that we have done for the Poll response timeout, also for the asynchronous timeout a safety margin, T_S, has to be added to the theoretical timeout value. Table 5.12 shows the mathematical worst case value of the asynchronous timeout resulting from the theoretical analysis.

$$T_{ASnd-timeout}^{max} = T_{ASnd-timeout}^{max-th} + T_S \tag{5.14}$$

Symbol	Minimum Value
$T_{ASnd-timeout}^{max-th}$	$41\mu s$
$T_{ASnd-timeout}^{max}$	$54\mu s$

Table 5.4: Asynchronous timeout values, mathematical and with a safety margin.

5.2 POWERLINK Wireless Extension Timing

The extension at the data link layer of a POWERLINK network communication 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 weakly hard real-time requirements, the above limitations may be well tolerated and, hence, wireless extensions of EPL at the data link layer may be profitably employed. In order to correctly configure the EPL network in the case of the wireless extension it is important to recompute a theoretical analysis. The user has to take into account the time necessary to transmit the frame in a wireless channel and the randomness which is introduced using the new medium access technique CSMA/CA.

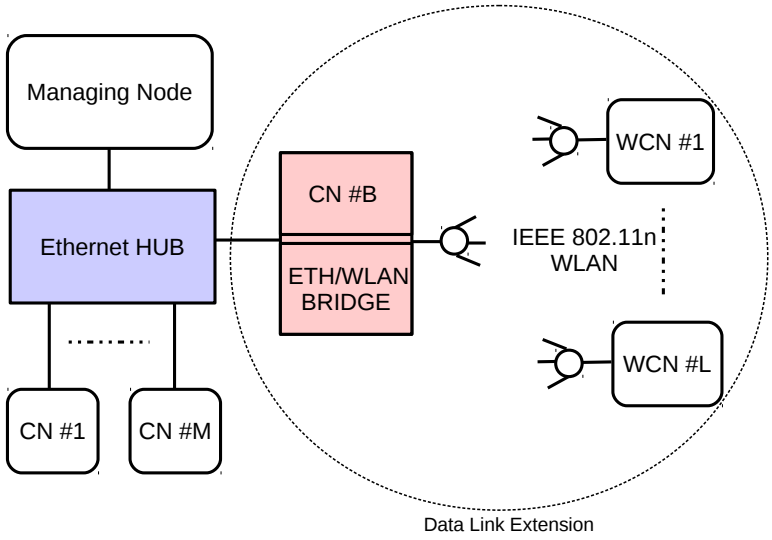


Figure 5.6: POWERLINK wireless extension architecture

In the following analysis we will consider a general network composed by a wired segment composed by M CNs and a wireless segments composed by L WCNs. This network is shown in Figure 5.6, where a bridge realizes the connection between the wired segment and the wireless one. In particular, the following analysis will consider a wireless extension of an EPL network where:

- all CNs are connected using cables with the same length (i.e. 2 meters)

and performance.

- All WCNs are at the same distance and the wireless channels have the same performance.
- All the CNs have the same response time.
- All the WCNs have the same response time, greater than the one of the CNs.
- Only minimum size EPL frames (i.e. 64 bytes) are supposed to be exchanged.
- The wired segment follows an ideal operating condition, whereas the possible sources of variability and randomness of the wireless communication are discussed.

Cycle time and Poll response timeout are the parameters which more influence a successful EPL communication. The main performance index considered in this analysis is the Polling time, indeed the cycle time results from this. In the case of the presence of wireless CNs (WCNs), the polling time is the time needed to successfully communicate with a wireless node, thus it is the time to transmit a PRes and receive the corresponding PRes. Clearly, this amount of time includes also the transmission time over the wireless channel and the bridge delay. To summarize, the Polling time of an hybrid EPL network is influenced by: the EPL protocol, the 802.11 protocol, the bridge delay and the intrinsic behaviour of the components and the condition of the wireless channel. All these factors have to be taken into account to provide a proper estimation of the range of values that the polling time can assume. Considering the network shown in Figure 5.6, the main elements that affect the time required for the polling procedure can be identified. In particular, the polling time T_{poll} turns out to be characterized by two 802.11 specific elements, summarized as follows:

1. the time necessary for the transmission of an 802.11 frame.
2. The time necessary to gain access to the wireless medium, according to the CSMA/CA retransmission protocol.

In the following analysis, firstly we develop a theoretical representation for these elements, finally we could characterize the Cycle time, the Poll response time and asynchronous timeout in the same way as it is done in the previous section.

5.2.1 Typical Parameters

In the following theoretical analysis several parameters will be used. The main 802.11 parameters, taken from the 802.11 standard [6] and from [8], are summarized in Table 5.5 and afterwards described. Again, we focus on the worst case, to give maximum time limit of the parameter which can be used in the offline configuration setup to ensure performance very close to the configured EPL ones.

Symbol	Value	Description
T_{WP}	10 ns	Wireless medium propagation delay.
T_{DIFS}	$28\mu s$	Duration of an IEEE 802.11 Distributed Interframe Space.
T_{SIFS}	$10\mu s$	Duration of a Short Interframe Space.
t_{slot}	$9\mu s$	IEEE 802.11 slot time.
N_{max}	7	Max number of MAC-layer retries.
T_{ACK}	34 μs	Transmission of an IEEE 802.11 ACK frame.
T_{ACK-TO}	45 μs	ACK- timeout time.
T_B	$10\mu s$	Bridging delay.

Table 5.5: Worst case values of the 802.11 parameters which will be used in the following theoretical analysis.

ACK timeout time The standard states that:

$$T_{ACK-TO} = T_{SIFS} + T_{ACK} \tag{5.15}$$

Bridging Delay It is the delay introduced by the bridge when it polls the i-th WCN. The delay is mainly due to latency and queueing. Although a general value for the latency can not be specified, we selected $10\mu s$ as the latency of the bridge model used in the simulations.

5.2.2 Transmission time of a frame

It is worth observing that the size of an IEEE 802.11 EPL frame, essential to compute the transmission time, is obtained analysing the bridge procedure.

The bridge encapsulates the Ethernet EPL frame in the playload field of an 802.11 frame. Therefore, the total size is obtained by adding to the length of the application playload both the size of the upper layer headers, summarized in Table 5.6, and the size of a preamble. Therefore, an 802.11 EPL frame is composed by 112 Bytes, except in the case of asynchronous frames which, in the worst-case of an EPL frame equal to the MTU (300 bytes), is composed by 348 Bytes.

Layer	Protocol	Header size
Data link	MAC	24 Bytes
Logical link layer	LLC	8 Bytes
Preamble	-	16 Bytes

Table 5.6: Header sizes for under network layers protocol.

Version	Transmission rate [Mb/s]	T_{WF} [μs]	T_{WASnd} [μs]
IEEE 802.11 g	54	17	52
IEEE 802.11n	13.5	67	206
	27	33	103
	40.5	22	69
	54	17	52
	81	22	34
	108	9	26
	121.5	8	23
	135	7	21

Table 5.7: Transmission times of an IEEE 802.11 EPL frame.

With these assumptions, the transmission time of a frame on the wireless channel is a deterministic quantity, that can be computed for different versions of the IEEE 802.11 standard and with different choices of the transmission rate, as reported in Table 5.7.

$$\begin{aligned} T_{WF} &= \frac{112 \cdot 8}{rate} \frac{bit}{Mb/s} \\ T_{WASnd} &= \frac{348 \cdot 8}{rate} \frac{bit}{Mb/s} \end{aligned} \tag{5.16}$$

5.2.3 Transmission on a wireless channel

The time needed for the transmission of messages on the wireless channel is a random variable with a wide range of possible values. Among the factors that influence its behaviour, there are the MAC protocol operations, the messages sizes, the SNR and the external interference. The following analysis focuses on T_{wifi}, the time a bridge needs to successfully poll a WCN on a wireless channel. In detail, T_{wifi} represents the time between the instant in which the 802.11 PReq frame start to be transmitted by the bridge on the wireless channel and the instant in which the bridge receives the 802.11 PRes frame from the polled WCN. Moreover, the following analysis is done intersecting the ones performed in [3] and [25].

1. To begin, a first ideal analysis can be carried out assuming that no other sources are transmitting, hence it is reasonable assuming that the message transmission is always complete with success at the first attempt, without the intervention of the retransmission mechanism.

   Looking at the Figure 5.7 the transmission time on the wireless channel in absence of retransmissions can be expressed as:

   $$T_{wifi} = T_{W-PReq} + T_{W-PRes} + 2 \cdot (T_{DIFS} + T_{WP}) + T_R^{CN} \tag{5.17}$$

   where T_{DIFS} is a fixed value which depends on the version of the IEEE 802.11 standard, T_{W-PReq}, T_{W-PRes} are the transmission times of the PReq and PRes frames, T_{WP} is the wireless propagation delay and T_R^{CN} is the CN response time.

2. The exchange of frame illustrated by Figure 5.7 represents an ideal case, in which all frames are successfully delivered at the first attempt. However, wireless transmissions could fail. In such a case, the IEEE retransmission mechanism kicks in. Figure shows the packet flow at MAC layer in the case of one transmission failure.

   With respect to the Figures 5.8a and 5.8b the transmission time on the wireless channel in the case of one transmission fail of the PReq or PRes frames can be expressed as:

   $$\begin{aligned} T_{wifi} = 2T_{W-PReq} + T_{W-PRes} + 3 \cdot (T_{DIFS} + T_{WP}) + T_{ACK-TO} + \\ + T_{BO}(1) + T_R^{CN} \end{aligned} \tag{5.18}$$

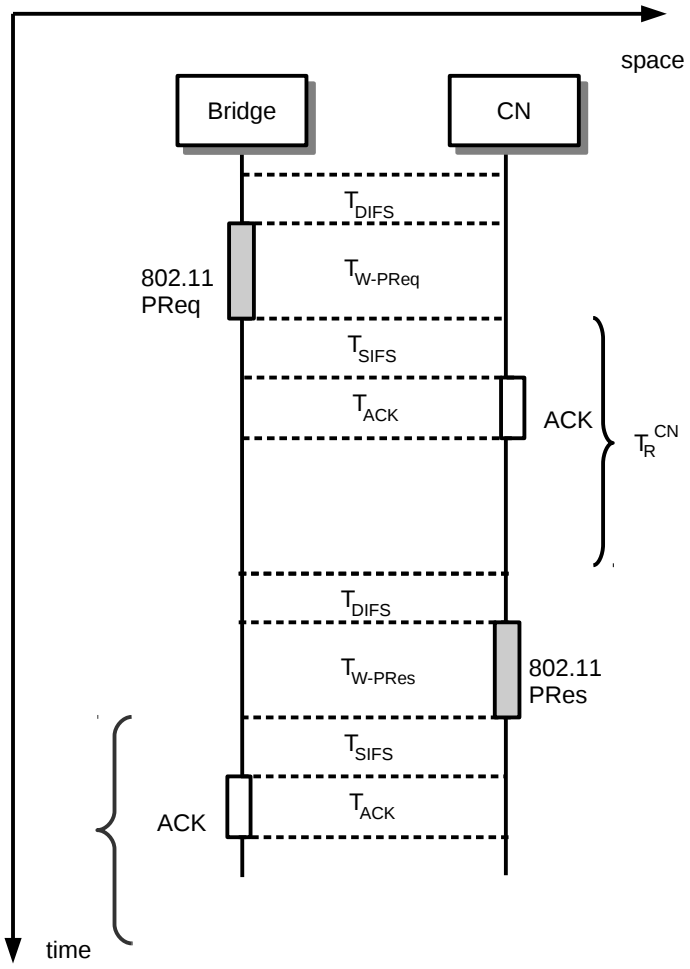


Figure 5.7: Exchange of packets in the ideal case.

where T_{ACK-TO} is the maximum time that an IEEE 802.11 STA waits for the ACK before declaring the loss of the last frame sent, T_{BO} is the random backoff time that a STA waits before attempting again the transmission. It is worth remarking that, according to equation 5.6 T_{BO} is a random variable. Hence, the whole T_{wifi} value it is no longer deterministic.

3. The more realistic case that can be considered, however, implies that both the command frame and the state frame could be retransmitted an arbitrary number of times. First of all, the computation of the wireless

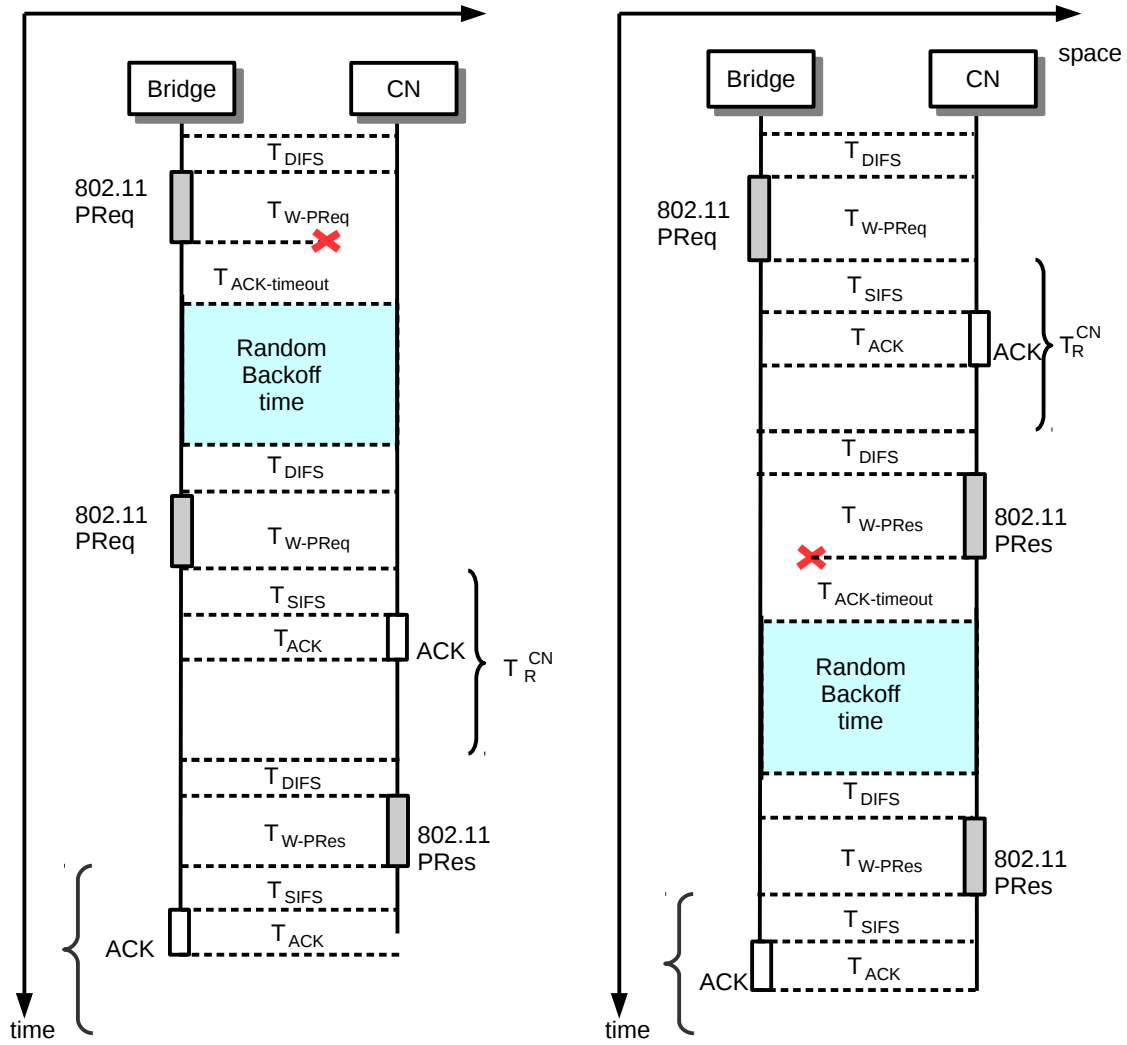


Figure 5.8: Exchange of packets in the case of one transmission fail of the PReq frame, Figure 5.8a, or of the PRes frame, Figure 5.8b.

transmission time is done in case that only one frame, for example PREq, is retransmitted n times, for a total of n + 1 transmissions. In this case, the total time required for the transmission of frames on the wireless channel is:

$$T_{wifi} = (n+1) \cdot T_{W-PReq} + T_{W-PRes} + (n+2) \cdot (T_{DIFS} + T_{WP}) + \\ + n \cdot T_{ACK-TO} + \sum_{i=1}^{n} T_{BO}(i) + T_R^{CN} \tag{5.19}$$

An even more general case is considered if also the state messages can be retransmitted. The number of retransmissions for the PReq frames is indicated again with n, while that of PRes frames is indicated with m. The total wireless transmission time hence becomes:

$$\begin{aligned} T_{wifi} =& (n+1) \cdot T_{W-PReq} + (m+1) \cdot T_{W-PRes} + \\ &+ (n+m+2) \cdot (T_{DIFS} + T_{WP}) + (n+m) \cdot T_{ACK-TO} + \\ &+ \sum_{i=1}^{n} T_{BO}(i) + \sum_{j=1}^{m} T_{BO}(j) + T_R^{CN} \end{aligned} \tag{5.20}$$

This case will be used in the following theoretical analysis, indeed it represents the worst-case possible during the transmission on the wireless channel.

5.2.4 Cycle Phase

According to the poll response time also the cycle time can be set equal to its maximum value, to achieve the best performance for what concern communication timeliness and therefore cycle loss never happens. Actually, to avoid the effect of components non-idealities and possible approximations in the theoretical model, the cycle period will be set slightly above the computed maximum value of the polling time.

5.2.5 Isochronous Phase

In an ideal scenario, T_{iso} 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.

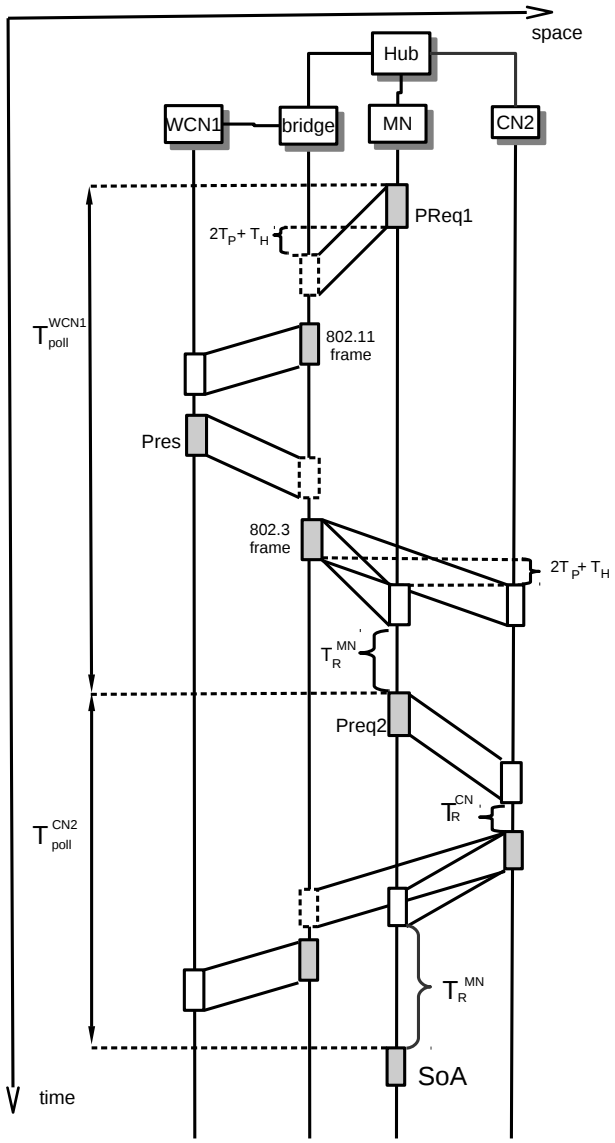


Figure 5.9: Space-time diagram of the polling procedure of an EPL hybrid network communication.

The general expression for the duration of the isochronous period of an hybrid network is given by:

$$T_{iso} = T_{sync} + M \cdot T_{poll}^{CN} + L \cdot T_{poll}^{WCN} \tag{5.21}$$

where T_{sync} is the synchronization phase (equation 5.4) and the other terms build the polling time. In particular, M is the number of the wired CN, L is

the number of the wireless CNs (WCNs), T_{poll}^{CN} is the polling time for a CN (equation 5.6) and T_{poll}^{WCN} is the polling time for a WCN.
As an example, Figure 5.9 shows the space-time diagram of the polling procedure for an EPL hybrid network composed by one WCN and one CN.

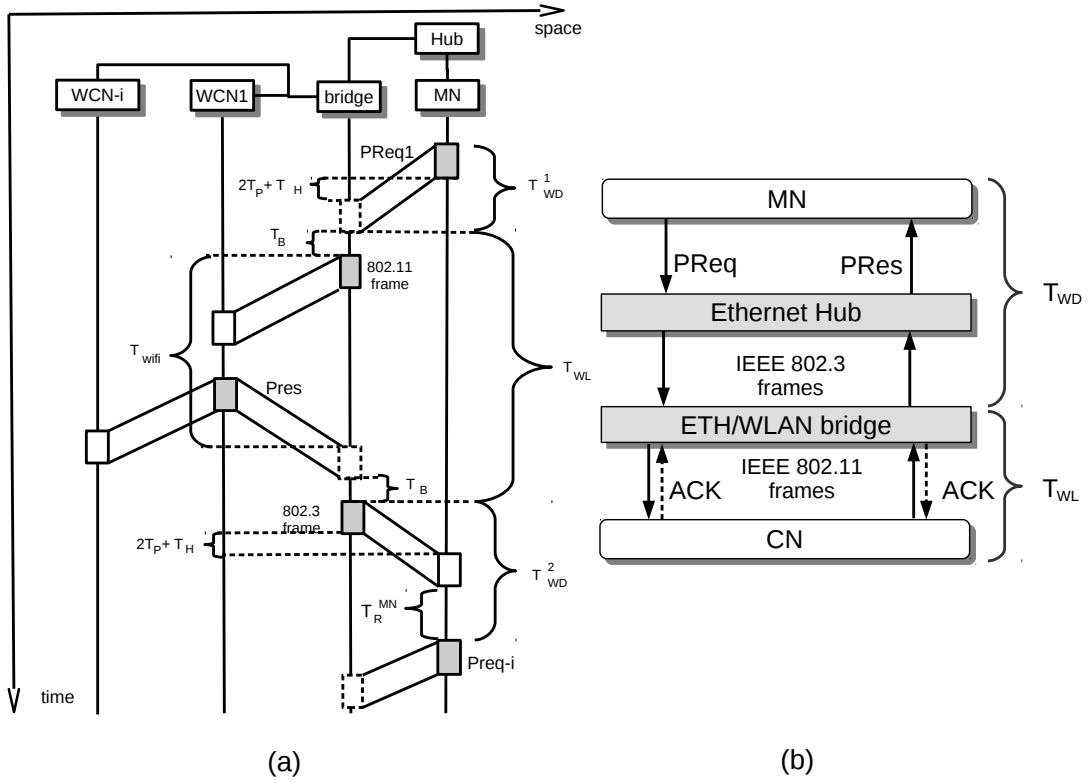


Figure 5.10: Space-time diagram (Figure 5.10a)and sequence of the operations (Figure 5.10b) of the polling procedure of a WCN.

In order to evaluate T_{iso} the only term which has to be characterized is the polling procedure of a WCN, i.e. T_{poll}^{WCN}.
As can be seen in 5.10b, 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_{poll}^{WCN} is given by:

$$T_{poll}^{WCN} = T_{WD} + T_{WL} \tag{5.22}$$

where T_{WD} is the time needed to transmit the frames on the wired segments, i.e. the transmission of the PReq from MN to bridge and the transmission of the PRes from the bridge to the MN. Instead, the second term accounts for

the transmission of the frames on the wireless segments, i.e. the transmission of the 802.11 PReq frame from the bridge to the CN and the transmission of the 802.11 PRes frame from CN to bridge.
We characterize T_{WD} and T_{WL} as follows with refer to Figure 5.10a, which shows the space-time diagram of the polling procedure of one WCN.

$$\begin{cases} T_{WD} = T_{WD}^1 + T_{WD}^2 \\ \qquad = T_{PREq} + T_{Pres} + 2 \cdot (2T_P + T_H) + \cdot T_R^{MN} \\ T_{WL} = 2T_B + T_{wifi} \end{cases} \tag{5.23}$$

where T_B is described in Table 5.5, whereas T_{wifi} is defined by equation 5.17.

5.2.6 Statistical characterization of WCNs Polling time

The detailed analysis of all polling time components carried out in this Section can be exploited to obtain a statistical characterization of the behaviour of the random variable polling time T_{poll}^{WCN} . In particular, the interest is on the maximum and minimum value that this variable can assume. Substituting equations 5.23 and 5.20 in equation 5.22, we obtain:

$$\begin{aligned} T_{poll}^{WCN} =& T_{WD}^1 + T_{WD}^2 + T_{WL} \\ =& [T_{PReq} + T_{PRes} + 2 \cdot (2T_P + T_H) + T_R^{MN}] + (n+1) \cdot T_{W-PReq-} + \\ & (m+1) \cdot T_{W-PRes} + (n+m+2) \cdot (T_{DIFS} + T_{WP}) + \\ & + 2T_B + (n+m) \cdot T_{ACK-TO} + \sum_{i=1}^{n} T_{BO}(i) + \sum_{j=1}^{m} T_{BO}(j) + T_R^{CN} \end{aligned} \tag{5.24}$$

From the performed analysis, it is evident that the randomness of T_{poll}^{WCN} lies entirely on the transmission time on the wireless medium. More specifically, the only source of randomness is the random backoff time, T_{BO}, and in particular whose distribution depends on the number of retransmissions of PReq and PRes frames.

- To compute the *minimum value* assumed by T_{poll}^{WCN} , the ideal case is considered, where all frames are transmitted successfully at the first attempt. In this case we use $T_{BO} = T_{BO}^{min} = 0$ and $n = m = 0$, therefore and T_{poll}^{WCN} become:

$$T^{WCN}_{poll_{min}} = [T_{PREq}+T_{Pres} + 2 \cdot (2T_P + T_H) + T^{MN}_R]+ \\ +T_{W-PReq} + T_{W-PRes} + 2 \cdot (T_{DIFS} + T_{WP}) + 2T_B + T^{CN}_R \tag{5.25}$$

Moreover, in the particular case of all frame with the same transmission time, respectively T_F for an 802.3 frame and T_{WF} for an 802.11 frame, the expression 5.25 becomes:

$$T^{WCN}_{poll_{min}} = [2\cdot(T_F + 2T_P + T_H) + T^{MN}_R]+ \\ +2 \cdot (T_{DIFS} + T_{WF} + T_{WP}) + 2T_B + T^{CN}_R \tag{5.26}$$

- Conversely, the *maximum value* of T^{WCN}_{poll} is obtained when the maximum allowed number of retransmission attempts ($N_{max} = 7$) is performed for both the frames and, each time the backoff time is randomly selected, the worst-case value is picked.

  According to the analysis done in [25]:

$$T^{max}_{BO} = t_{slot} \cdot \left[2 \sum_{i=1}^{N_{max}} \left(2^{i-1}(CW_{min} + 1) - 1\right)\right] \\ = 2t_{slot} \cdot \left[(CW_{min} + 1) \cdot (2^{N_{max}} - 1) - N_{max}\right] \tag{5.27}$$

  and

$$T^{WCN}_{poll_{max}} =[T_{PReq} + T_{PRes} + 2 \cdot (2T_P + T_H) + T^{MN}_R] + (N_{max} + 1)\cdot \\ \cdot (T_{W-PReq} + T_{W-PRes}) + (2N_{max} + 2) \cdot (T_{DIFS} + T_{WP})+ \\ + 2T_B + 2N_{max} \cdot T_{ACK-TO} + T^{max}_{BO} + T^{CN}_R \tag{5.28}$$

  Moreover, in the particular case of all frame with the same transmission time, respectively T_F for an 802.3 frame and T_{WF} for an 802.11 frame, the expression 5.28 becomes:

$$T^{WCN}_{poll_{max}} = [2\cdot(T_F + 2T_P + T_H) + T^{MN}_R] + (2N_{max} + 2) \cdot (T_{DIFS} + T_{WF}+ \\ + T_{WP}) + 2T_B + 2N_{max} \cdot T_{ACK-TO} + T^{max}_{BO} + T^{CN}_R \tag{5.29}$$

Version	Data rate $[Mb/s]$	$T^{WCN}_{poll_{min}}[\mu s]$	Maximum $T^{WCN}_{poll_{max}}[\mu s]$
IEEE 802.11 g	54	153	19560
IEEE 802.11n	13.5	252	20365
	27	186	19834
	40.5	164	19657
	54	153	19569
	81	142	19480
	108	136	19436
	121.5	134	19421
	135	133	19409

Table 5.8: Maximum and minumum value assumed by polling time for different IEEE 802.11 versions and data rates.

As a first result, Table 5.8 shows both the minimum and maximum values assumed by T^{WCN}_{poll} in correspondence of the chosen value of data transmission rate on the wireless channel. Moreover, in the case of IEEE 802.11n Figure 5.11 shows the evolution of both the maximum and the minimum values of the Polling time versus the values of the transmission rate.

As can be seen, the values of the polling time lie in the range $[T_p^{min}, T_p^{max}]$, where:

$$\begin{aligned} T_p^{min} &= 133\mu s \\ T_p^{max} &= 20360\mu s \end{aligned} \tag{5.30}$$

Moreover, it is evident that the smaller the data rate the higher the polling time, therefore it can be concluded that the choice of data rate has quite an impact. However, using a higher data rate allows to speed up the transmission process but also worsens the robustness of communication, possibly leading to errors which may require more retransmissions, thus causing an increase in polling time.

It is also interesting to analyse the behaviour of the maximum value of the polling time versus the number of consecutive retransmissions, for a given version of the IEEE 802.11 standard and choice of data rate. As an example, Figure 5.12 shows the behaviour of the polling time for the IEEE 802.11n version of the standard with data rate 135 Mb/s (the highest possible choice).

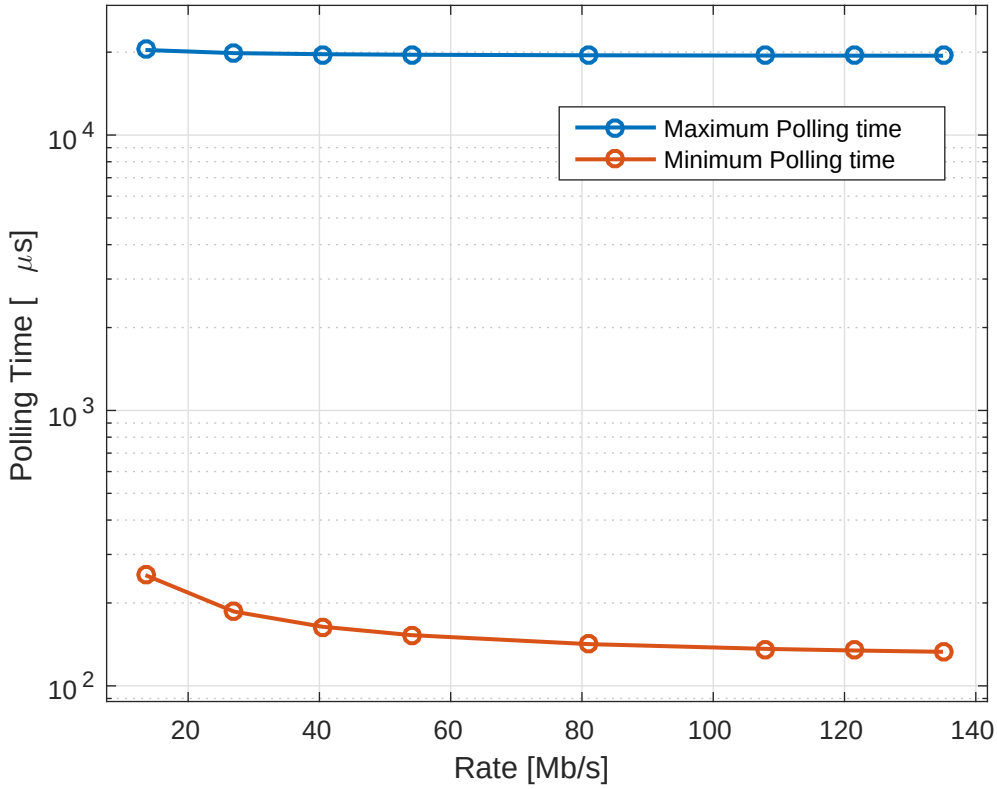


Figure 5.11: Evolution of the polling time versus the data rates for IEEE 802.11n, for the maximum polling time values and the minimum one.

In particular, Figure 5.12 describes the case where the maximum number of retransmission attempts varies among $N_{max} = N = [1, 2, 3, 4, 5, 6, 7]$ and it is performed for both the PReq and PRes frames. Moreover, each time the backoff time is randomly selected, the worst case is picked. The values are also reported in Table 5.9.

Looking at Figure 5.12 and Table 5.9, it is evident that the smaller the maximum number of retransmission, the smaller the polling time, therefore it can be concluded that the choice of N_{max} has quite an impact on the Polling time and, hence, on the isochronous period duration. As a consequence, using a smaller N_{max} allows to speed up the transmission process. However, such a choice worsens the robustness of communication, since it leads to possible frame losses.
As an example, the choice N_{max}=3 seems to be an good choice. indeed, al-

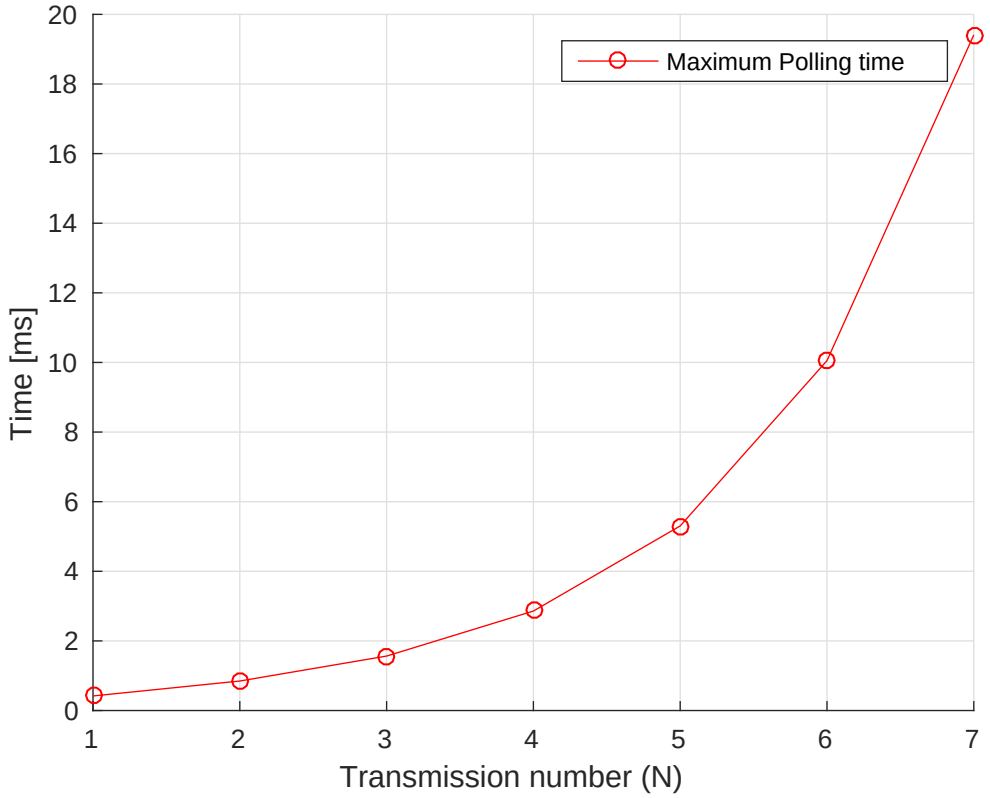


Figure 5.12: Evolution of the polling time versus the number of retransmission for a network with one WCN.

though the possibility of three attempts improves the robustness of communication, this also ensures that the polling time does not exceed the 1.56 ms of duration.
In conclusion, the maximum value of the isochronous period results:

$$T_{iso} = T_{sync} + M \cdot T_{poll}^{CN} + L \cdot T_{p}^{max} \tag{5.31}$$

As an example, for an hybrid EPL network with one CN and one WCN the isochronous time has to be greater or equal to the worst-case value (with N_{max} = 7):

$$T_{iso} \geq 45\mu s + 44\mu s + 20365\mu s \approx 21ms \tag{5.32}$$

Now, we can compare this result with the one obtained in the previous section. We remark that in a wired network composed by two CNs $T_{iso} \geq 133\mu s$.
It is interesting to analyse how the value of the isochronous period changes with the number of stations. T_{iso} is directly proportional to the number of

both the CNs and the WCNs, however since the polling of a WCN in general requires more time than one of a CN, the number of WCNs has a greater impact on T_{iso} than the number of CNs.

Number of retransmission N	$T^{WCN}_{poll_{max}}[ms]$
1	0.42
2	0.84
3	1.56
4	2.85
5	5.3
6	10
7	19.4

Table 5.9: Maximum value assumed by polling time for IEEE 802.11n versions and data rate 135 Mb/s varying the maximum number of retransmission N_{max}.

5.2.7 Poll Response Timeout (EPL)

The poll response timeout is a POWERLINK porotocol parameter. Especially, it is the amount of time that the MN waits for the PRes, after having sent the PReq, of the CNi before considering this station as having failed and polls the CNi+1. According to the previous definition, the poll response timeout is measured exactly from the start of the PReq frame transmission to the beginning of the PRes reception, moreover, in the case of a WCN polling during this amount of time there must be also the transmission time over the wireless channel.
The MN could experience time-outs in querying the WCNs due to the randomness of the polling time. Such a situation reveals dangerous since several consecutive time-outs may lead to the exclusion of the WCN from the isochronous period. This problem, however, may be solved (at least in principle) setting suitable time-out values after a careful evaluation of the time requested to poll the WCNs.
In this theoretical analysis, we are interested on the highest value from all of the individual stations should be used. Obviously the station with the high value have to be searched among the wireless stations (WCNs). Considering this situation, Figure 5.13 shows the space-time diagram of the polling operation

of a WCN. As can be observed from the figure, the WCN poll response timeout is exactly the T_{poll}^{WCN}, reported in equation 5.24. In order to consider the worst case, we take the maximum value which the polling time could have T_p^{max}, computed in the particular case of all frame with the same transmission time from equation 5.28.

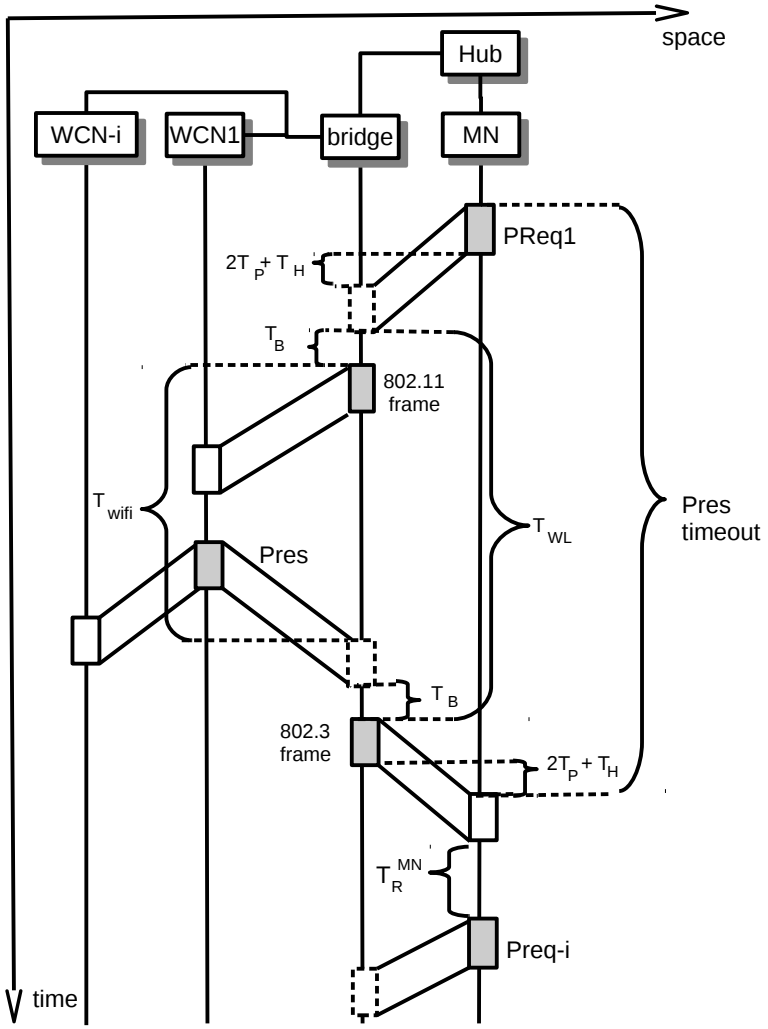


Figure 5.13: Space-time diagram of the polling procedure of a WCN during an EPL communication.

Therefore, taking the Poll response timeout greater to T_p^{max}=20.36 ms seems to be a reasonable choice in order to ensure the polling of whatever WCN station.
In particular, an appropriate safety margin, T_S, has to be added, which corre-

sponds to the 30% of the theoretical value. Table 5.10 shows the mathematical worst case value of the poll response timeout resulting from the theoretical analysis.

$$T_{Pres-timeout}^{maxWCN} = T_p^{max} + T_S \tag{5.33}$$

In detail, as can be seen in Figure 5.13 the timeout time does not include the first transmission time of the PReq and the MN response time, which instead are included in T_p^{max}, however since we would like to add a safety margin the adding of these terms is not a problem.

Symbol	Minimum Value
$T_{PRes-timeout}^{maxWCN}$	$27ms$

Table 5.10: Poll response timeout values, mathematical and with a safety margin.

5.2.8 Frame delivery Timeout (RSIN)

The RSIN technique is a dynamic rate selection algorithm introduced in [8] and briefly described in Section 3. This algorithm is based on an optimization problem, solved before any packet transmission, which can be formulated as:

$$\begin{aligned} &\min_{n \leq N_{max}, r_i \in R} \mathcal{L}(L, S, n, r_1, r_2 ... r_N) \\ &\max_{n \leq N_{max}, r_i \in R} \mathcal{D}(L, S, n, r_1, r_2 ... r_N) \leq D \end{aligned} \tag{5.34}$$

In other words, the algorithm finds the number of attempts $n \leq N_{max}$ and the relevant sequence of rates $r_1, r_2 ... r_N$ (where r_i is the rate selected for the i-th attempt between R=[13.5 27 40.5 54 81 108 121.5 135] for IEEE 802.11n) to be used for the transmission of a packets within a deadline D. Moreover, the algorithm also minimizes the residual transmission error probability $\mathcal{L}$ for a packet with a playload of L bytes (in our case L=64), transmitted to a receiver which perceives an SNR level of S dB.

Considering the wireless extension of EPL, the RSIN technique may be adopted, with the same configuration, by both the bridge and the CNs involved in the wireless communication.

The device wishing to transmit on the wireless channel solves the optimization problem before any packet transmission. Therefore, the algorithm parameters

could be computed in the offline phase. In order to be able to use this algorithm we need to compute the frame delivery time, $\mathcal{D}$, and its maximum, D, in the considered EPL real-time communication scenario. From a practical point of view, $\mathcal{D}$ is defined as the period between the instant in which an IEEE 802.11 packet starts to be transmitted and the instant in which the transmitter receives the correspondent ACK. On the other hand, the Frame delivery Timeout, D, is the amount of time within the bridge or the WCN transmits an 802.11 frame, then they consider the packet as having failed. Therefore, it is a RSIN parameter.
In the above analysis T_{wifi} is exactly the time that a bridge needs to successfully poll a WCN on the wireless channel. Therefore, T_{wifi} is the sum of the PReq frame delivery time and the PRes frame delivery time and it can be formulated as:

$$T_{wifi} = T_{PReq,W} + 2T_{WP} + T_R^{CN} + T_{PRes,W} \tag{5.35}$$

The maximum frame delivery, D, is determined under the worst-case assumption that $n = m = N_{max}$ retransmission are needed to successfully send both PReq and PRes frames. The worst-case is represented by T_{wifi}^{max}, so:

$$T_{PReq,W} + T_{PRes,W} \leq T_{wifi}^{max} - 2T_{WP} - T_R^{CN} \tag{5.36}$$

where, we can define a delivery frame timeout $T_{TO,W}$ as:

$$T_{TO,W} = T_{wifi}^{max} - 2T_{WP} - T_R^{CN} \tag{5.37}$$

Since in the previous section we had calculated the value of T_p^{max}, we compute T_{wifi}^{max} from T_p^{max}. Indeed, it our case follows that:

$$T_{wifi}^{max} = T_p^{max} - 2(T_F + 2T_P + TH) - T_R^{MN} - 2T_B \tag{5.38}$$

Substituting equation 5.38 in equation 5.37, we obtain:

$$T_{TO,W}^{th} = T_p^{max} - 2(T_F + 2T_P + T_H) - T_R^{MN} - 2T_B - 2T_{WP} - T_R^{CN} \tag{5.39}$$

and so:

$$T_{TO,W} = T_{TO,W}^{th} - T_S \tag{5.40}$$

where an appropriate safety margin, T_S, has to be added, which corresponds to the 30% of the theoretical value.
Consequently, assuming that all frames have the same length, we can conclude that in our analysis the frame delivery and its maximum are:

$$\begin{aligned} 2\mathcal{D} &= T_{PReq,W} + T_{PRes,W} \\ D &= \frac{T^{th}_{TO,W}}{2} \end{aligned} \tag{5.41}$$

Table 5.11 shows the mathematical worst case value of D resulting from this theoretical analysis.

Symbol	Minimum Value
D	$8ms$

Table 5.11: Maximum frame delivery on a wireless channel.

5.2.9 Asynchronous Timeout

Similarly to the analysis of the previous section, we are interested on the highest value from all of the individual stations, and it has to be searched among the wireless stations.

$$\begin{aligned} T^{max-th}_{ASnd-timeout} &= \max_{i=1..(L+M)}(T^{WCN,CN}_{ASnd-timeout}(i)) \\ &= \max_{i=1..L}(T^{WCN}_{ASnd-timeout}(i)) \end{aligned} \tag{5.42}$$

The asynchronous timeout is calculated like the poll response time, indeed it is measured from the start of the SoA frame to the reception of the ASnd frame. However, to consider the worst-case we have to use T_{W-ASnd} with the lower transmission rate. Therefore, in this analysis of the worst case we can not use T^{max}_{p} in the same way of the previous section.
To compute an appropriate analysis we use Figure 5.14, the space-time diagram of the asynchronous procedure of a WCN during an EPL communication. The time between the end of the SoA frame and the reception of the ASnd is composed by:

$$\begin{aligned} T^{WCN}_{ASnd-timeout} =& [T_{SoA} + T_{ASnd} + 2 \cdot (2T_P + T_H)] + (N_{max} + 1) \cdot \\ & \cdot (T_{W-ASnd} + T_{W-SoA})) + (2N_{max} + 2) \cdot (T_{DIFS} + T_{WP}) + \\ & + 2T_B + 2N_{max} \cdot T_{ACK-TO} + T^{max}_{BO} + T^{CN}_{R}] \end{aligned} \tag{5.43}$$

where T_{SoA} and T_{W-SoA} are time needed to transmit respectively an 802.3 SoA frame and an 802.11 SoA frame and T_{ASnd} and T_{W-ASnd} are time needed to transmit respectively an 802.3 ASnd frame and an 802.11 ASnd frame. In this analysis we consider $T_{SoA} = T_F$ and $T_{W-SoA} = T_{WF}$. Conversely, T_{ASnd} and

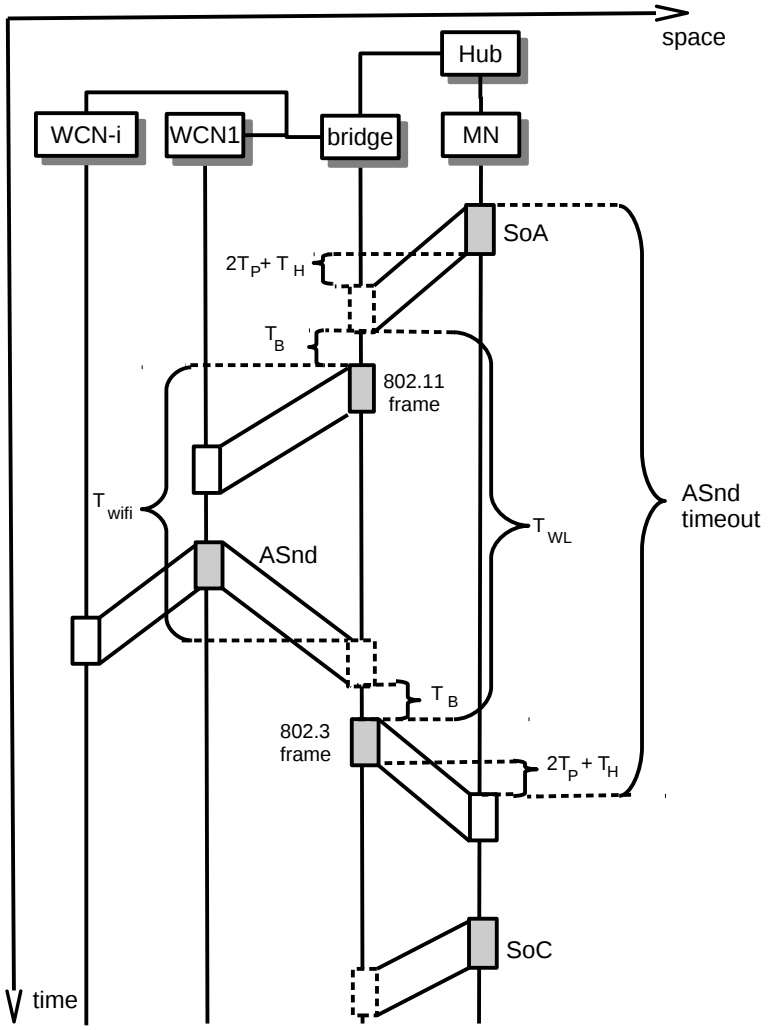


Figure 5.14: Space-time diagram of the asynchronous procedure of a WCN during an EPL communication.

T_{W-ASnd} are transmitted during the asynchronous communication, they can be equal to the other frame (64 bytes) or it can be greater. The duration of those frames depends on the communication necessity. In the worst-case the 802.3 ASnd frame is equal to 300 bytes, which is the maximum length allowed, and consequently the 802.11 frame is 348 bytes length.
Again, a safety margin, T_S, has to be added to the theoretical timeout value.

$$T^{max_{WCN}}_{ASnd-timeout} = T^{max-th_{WCN}}_{ASnd-timeout} + T_S \tag{5.44}$$

Symbol	Minimum Value
$T^{max-th_{WCN}}_{ASnd-timeout}$	$23ms$
$T^{max_{WCN}}_{ASnd-timeout}$	$30ms$

Table 5.12: Asynchronous timeout values, mathematical and with a safety margin.

Chapter 6

Experimental measurements

Some configurations of EPL networks, both wired and wireless, have been set up using the hardware devices presented in Chapter 4. These EPL networks have been subject to a series of experimental campaigns, in order to assess their performance and compare them with the results of the theoretical analysis conducted in Chapter 5. In particular, the polling time, whose expected behaviour has been analysed in detail, has been evaluated in different scenarios.

Industrial environments are often characterized by the presence of several sources of electromagnetic noise. Similarly, the wireless networks deployed in the laboratory are influenced by transmission errors and delays that may deteriorate their performance. Since the electromagnetic isolation was not achievable, we selected a channel which was not steadily used by other WLANs by monitoring the surrounding environment with a real-time spectrum analyzer. Moreover, several consecutive sessions have been performed, each one lasting long time, which is the expected working time for the system. The results have shown a common trend, so the outcomes presented in this Section all refer to the first period of operations in a specific session.

Two principal experimental campaigns have been performed to investigate the behaviour of the EPL real-time protocol. The first one regards the EPL wired communication performance, the second one is about the introduction of the EPL protocol in the IEEE 802.11 WLAN framework. There are several indicators that should be considered when evaluating POWERLINK real-time

Ethernet, however according to the provided theoretical analysis we focus on some of them. Specifically, in this work of thesis we focus on such as indicators: the cycle time and the polling time, because these parameters more influence a successful EPL communication. Particularly, the most important performance aspect considered is the polling procedure timing, indeed we have seen that considering an hybrid network, the 802.11 protocol more influence this parameter.
Figure 6.1 shows the laboratory environment where the experimental measurements will take place.

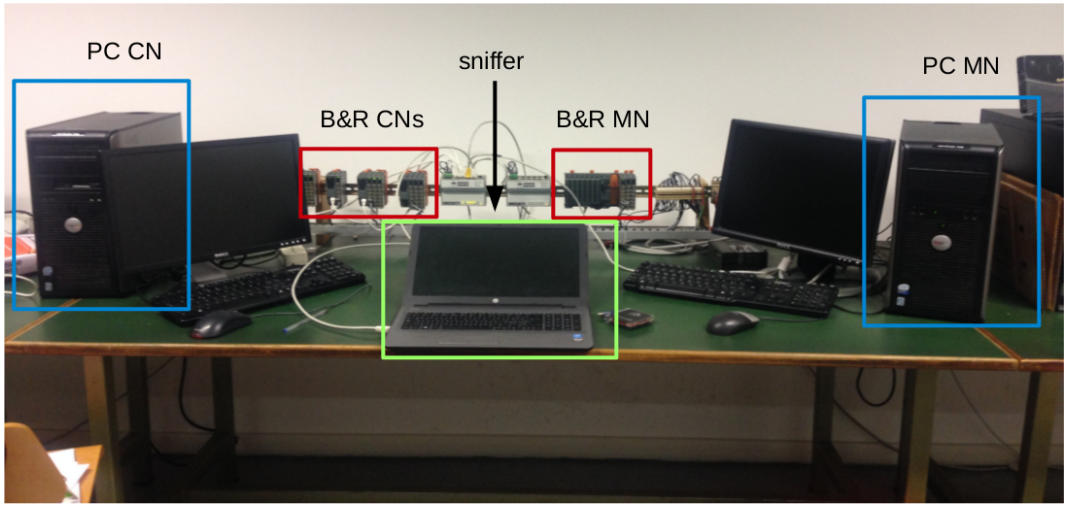


Figure 6.1: System for the experimental measurements.

6.1 Wired EPL configuration

This first measurement campaign is carried out in an environment which represents a real application scenario for the EPL protocol. We use the B&R devices: X20CP-1484 CPU, X20 Bus Controller, 0AC808 Hub. This allows to test the behaviour of the EPL protocol when facing the real-operating conditions that it may encounter during its industrial use.
Alternatively, a second measurement campaign is performed in an environment of research. In detail, during this session of measurements we will use both the B&R devices and the desktop PCs, that allow to manually configure the openPOWERLINK protocol using the openCONFIGURATOR tool.

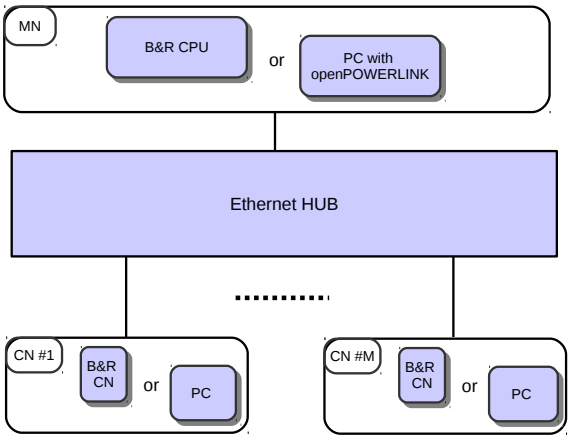


Figure 6.2: Ethernet POWERLINK one-level configuration.

6.1.1 Network description

We make a performance analysis for the network configurations shown in Figure 6.2, where the MN is connected with up to 5 CNs. As can be seen, both the B&R devices and desktop PCs with openPOWERLINK could implement a POWERLINK station.
Naturally, several other networks configurations could have been considered, however, the ones we choose are good representative of a large number of industrial communication applications. Indeed, as outlined in [41], the one-level configuration of Figure 6.2 based on a single Ethernet hub is typically employed at the device level of automation systems and/or by networked control systems.

6.1.2 Setup description

To begin, we have manually configured the hardware of the POWERLINK network. The configuration of a station depends on its features. Particularly, there is a clear distinction between legacy B&R nodes and PCs running the openPOWERLINK software.

B&R Automation Studio is the software needed to configure a B&R device. The use of this software is very easy because the programming interface of the program is very visual. The only thing that the user has to do is to specify one by one all the cards the Control Node or the Managing node has and to set up some variables. Figure 6.3 shows the setup procedure.

When a new project is created, the user can either use the Automation

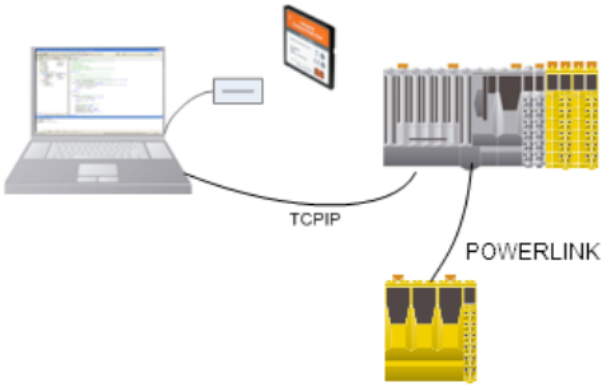


Figure 6.3: EPL network setup procedure using Automation studio.

Runtime simulation or define a new hardware configuration, selecting the X20CP-1484 CPU. Then, it can be possible to add a program, which can be written in several languages (such as Ladder, structured text, instruction list). Programs must be assigned to a CPU so that the B&R modules they contain can be generated and transferred to the target system when the project is compiled. Using the Object Catalog the user can choose the other hardware components, in our case the hub and the Bus Controllers X20BC-0083. Then in the `Hardware.hwl` the user must connect the devices. Using the POWERLINK configuration interface, shown in Figure 6.5, it is possible to configure the network parameters. When the network is correctly configured the user may build the project, in such a way the associated command list is created automatically and the generated file is displayed in the Automation Studio output window. In order to transfer to the target system all the information needed to work, the user must generate the CompactFlash card. Finally when the CompactFlash card is inserted in the X20CP-1484 CPU, this latter can be turned on.

openPOWERLINK Figure 6.4 shows the setup procedure, which will be described in detail in the following. Each device has a specific *XML Device Description File* (XDD) which can be purposely written or downloaded from [5]. Then this file is passed to the configuration tool, openCONFIGURATOR, which is used to characterize the EPL communication depending on the real-time capabilities of the devices. The user creates a project and defines the MN adding its XDD file. Then, the user can add the CNs according to the nodeID and their XDD files. Now it is possible to configure the performance parameters of the EPL network

using the Proprieties window, shown in Figure 6.6. When the network is correctly configured the user may build the project, in such a way openCONFIGURATOR creates a binary file `mnobd.cdc`, which represents a full network configuration including the network mapping information. The generated files are saved in the workspace output folder. The device running as MN needs the CDC file to correctly works, whereas the CNs will receive their configuration setting during the EPL communication from the MN.

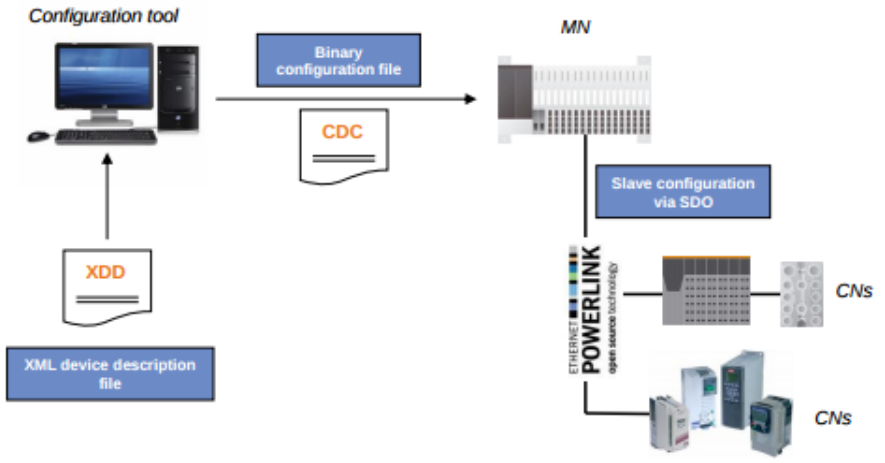


Figure 6.4: openPOWERLINK setup procedure.

Before starting the openPOWERLINK communication it is necessary to create a safety network between the MN and the CNs, which means to choose carefully the devices IP addresses according to the addressing rules described in 2.3.5.

openPOWERLINK_V2.3.2 is an open source ANSI C stack implementing the POWERLINK protocol. The openPOWERLINK stack itself is implemented via stack libraries. A stack library can either contain the whole stack (complete library), the user part (application library) or the kernel part (driver library). Therefore, the user has to build the Stack Libraries on a Linux system. In order to use the stack, Demo Applications are available. The user has to build the executables, namely `demo_mn_console.sh` and the `demo_cn_console.sh`. Running `demo_cn_console.sh` on the CN device, the user has to firstly choose the interface to be used for POWERLINK communication (eth0, eth1, wlan0) and then the application starts initializing the CN. Similarly for the MN, the user has to run `demo_mn_console.sh` linking the `mnobd.cdc`

created in openCONFIGURATOR.

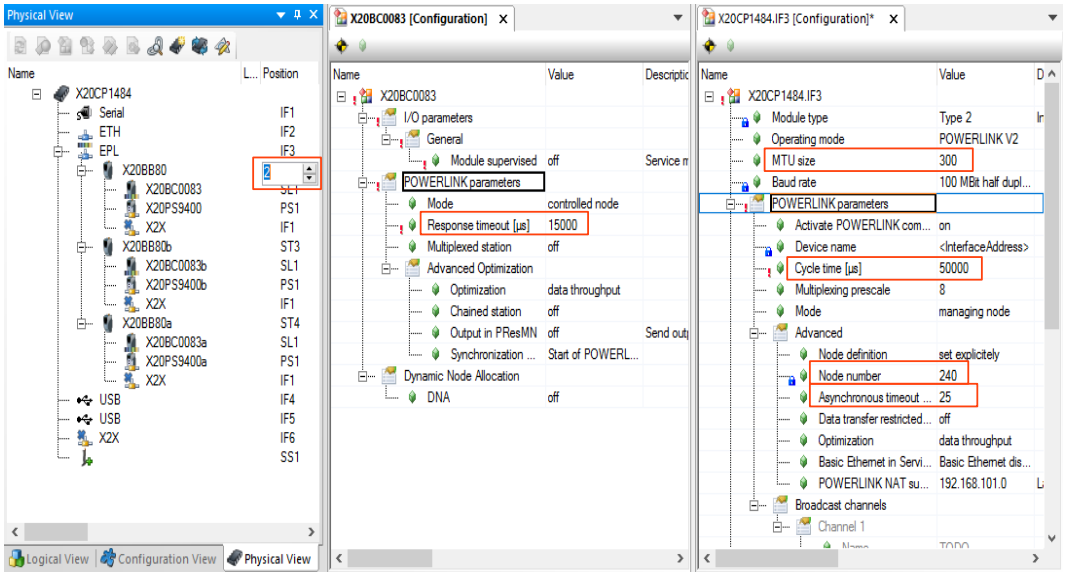


Figure 6.5: POWERLINK configuration interface of Automation Studio.

Mixed network Finally, [38] claims that it is possible to attach openPOWERLINK based nodes and other industrial devices (as the B&R modules), proving the absolute openness of EPL technology. Therefore, a mixed EPL network, composed by both B&R devices and openPOWERLINK ones, could be realized. In order to be obtain a correctly EPL communication between the devices of a mixed network, the user needs to follow only few recommendations. In this work of thesis we will analyse two particular mixed network:

1. openPOWERLINK MN + B&R I/Os.
   In order to set up this network the user has only to set the nodeID and the IP address according to the Automation Studio POWERLINK configuration interface.
2. B&R CPU + openPOWERLINK CN.
   For such a setup, the user has to carry some recommendations during the configuration of the MN in Automation Studio. Firstly, the user has to import the XDD file of the openPOWERLINK demo, which can be found in the stack's source code, in Automation Studio via "Import Fieldbus Device". Then it is possible to add a new "openPOWERLINK device" as CN and normally configure it. Finally, the user must activate the channels 0x6000/01 and 0x6200/02

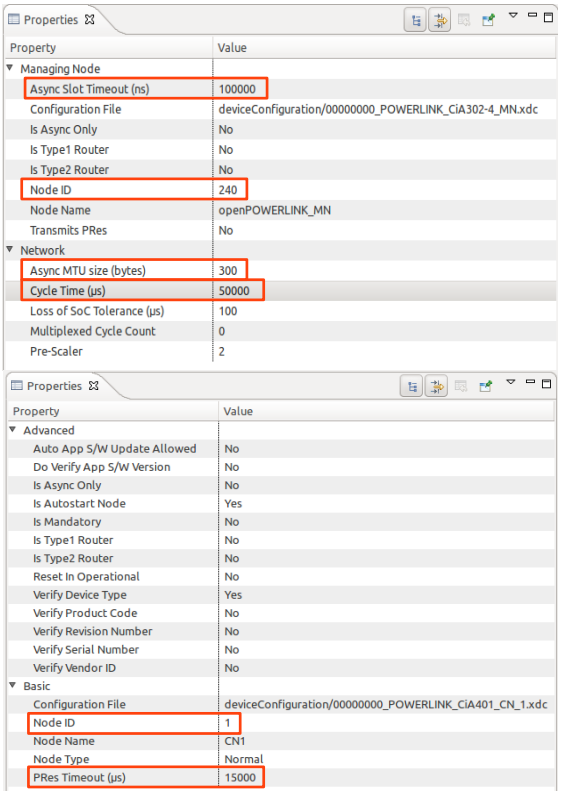


Figure 6.6: openPOWERLINK configuration interface in openCONFIGURATOR.

in the "I/O Configuration". The final results of this configuration phase is shown in Figure 6.7.

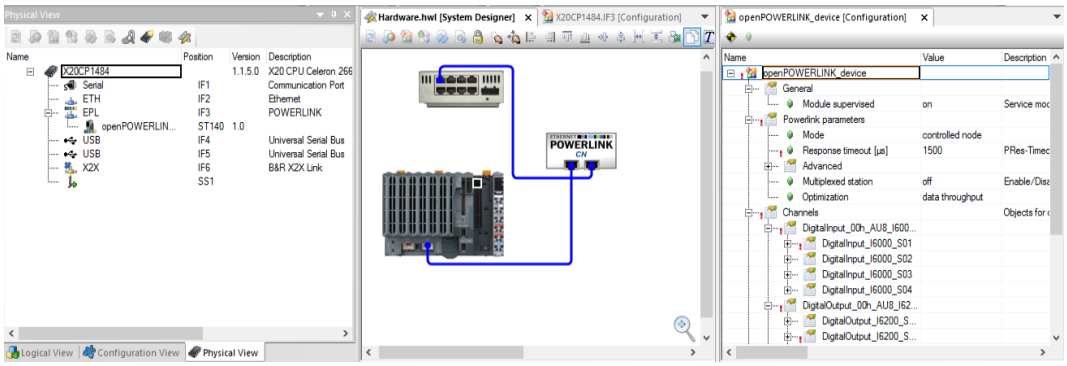


Figure 6.7: Configuration interface on Automation Studio for a mixed network.

6.1.3 EPL Parameters

In both the setup procedures of the EPL network, the user can set some parameters, among which the most important are:

- Network Architecture.
  The user could configure the number of the CNs and their nodeIDs. It is worth stressing that in case of B&R devices the nodeIDs must be set accordingly to the node switches, which are on the devices.

- Cycle time, (MN).
  It is a Managing node parameter. Automation Studio allows to configure a minimum cycle time of 200 μs, whereas openCONFIGURATOR require a cycle time value bigger or equal to 400 μs. The biggest value available for both the devices is 100 ms.

- Asynchronous MTU size, (MN).
  It is a Managing node parameter. The value range is from 300 - 1500 Bytes.

- Asynchronous Timeout, (MN).
  It is a Managing node parameter. The value range is from 5 - 100 ms, and 25 μs is the default value.

- Poll Response Timeout, (CN).
  It is a Controlled node parameter. The value range is from 1 - 30 ms, and 27 μs is the default value.

In particular the B&R devices parameters are configured using the POWERLINK interface configuration of Automation studio, shown in Figure 6.5. Whereas, the openPOWERLINK devices parameters are configured using the properties window of the tool openCONFIGURATOR, shown in Figure 6.6.

6.1.4 Experiment: Minimum cycle time

In the first experiment two EPL networks both composed by one MN and one CN are used, where one is made of only B&R devices, the CPU and the Bus Controller, whereas the other is made of PCs configured with openPOWERLINK. The aim of this experiment is to assess the differences between the two network types. In detail we will observe the cycle time. Since the B&R network is tailored for EPL application, we expect that it will have better performance than an openPOWERLINK network, thus smaller cycle time.

Theoretically, a POWERLINK cycle may reach the minimal value of 200 μs. But this value can be achieved only by using a specific hardware in the network components.
Indeed, it is worth underline that Automation Studio allows to configure a cycle time of 200 μs, whereas openCONFIGURATOR require a cycle time value bigger or equal to 400 μs. Therefore the available ideal minimal cycle time value is different with respect to the devices nature.
In order to verify whether there available minimal values could be effectively reached, we test one by one the networks.
In our implementation a B&R network achieves the minimal value of cycle time 200 μs, using this minimal value the network devices could have a correct EPL communication. Conversely, nevertheless the time cycle of an openPOWERLINK network could be set to a minimum value of 400 μs, using this minimum value the protocol never works because an `NmtEventNmtCycleError` cyclically occurs and the node has to be cyclically re-configured.

Cycle Time	B&R dev	openPOW dev
$200\mu s$	$\surd$	not available
$400\mu s$	$\surd$	$\times$
$3000\mu s$	$\surd$	$\times$
$3500\mu s$	$\surd$	unstable
$4000\mu s$	$\surd$	$\surd$

Table 6.1: EPL protocol operation state w.r.t the cycle time value.

Table 6.1 shows the outcomes of the tests, which evidenced that:

- in a B&R network the EPL always works, also with a cycle time value of 200 μs.

- Conversely, in an openPOWERLINK network configured with a cycle time value less than 3 ms the EPL protocol never works, instead using values between 3 ms and 4 ms the EPL protocol has a very unstable behaviour. More specifically, we see a blinking communication: periods of correct communication of variable duration and periods of no-communication during which the node is re-configured as a consequence of an `NmtEventNmtCycleError` error. Figure 6.8 shows the MN sequence of operations, cyclically repeated, during an openPOWERLINK no-communication due to a too smaller cycle time value. Finally, we

observed that using values greater or equal than 4 ms the EPL protocol properly works.

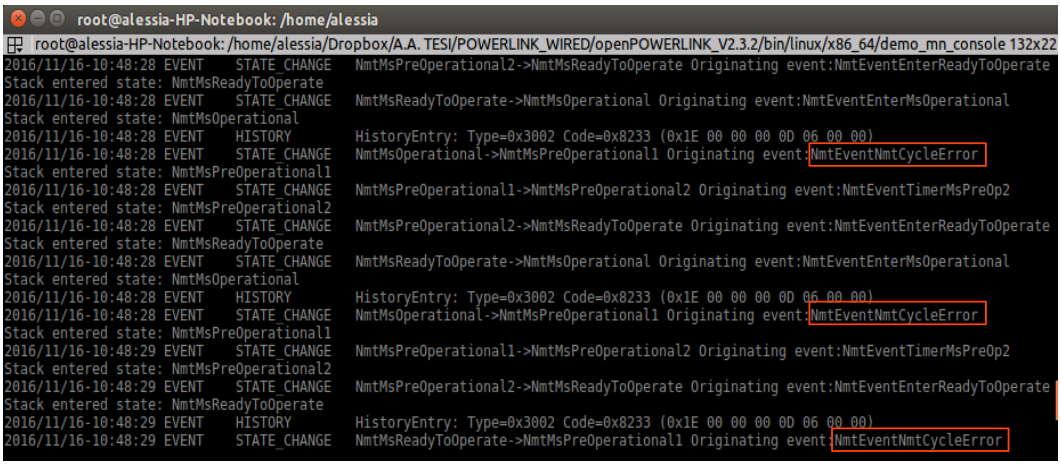


Figure 6.8: MN operations in an openPOWERLINK network configured with a too smaller cycle time value.

This behaviour is mainly due to the hardware composition of the device. A B&R node, since it is an hardware targeted at industrial communication, allows very fast data processing; whereas, an openPOWERLINK node, implemented on a desktop PC, is necessarily subjected to greater elaboration delays. For this reason an openPOWERLINK network needs longer cycle time values in order to correctly communicate using the EPL protocol.

6.1.5 Experiment: Polling Time

In this experiment the two formerly described EPL networks have been used. The aim of this experiment is to evaluate the differences between the two networks and in detail we will consider the polling time. Similarly to the previous experiment, we expect that a B&R device will have better performance than an openPOWERLINK network, thus smaller polling time.
In the theoretical analysis, we founded a mathematical worst case value of the Poll Response Timeout, nevertheless, both the experimental networks need a bigger Timeout in order to correctly communicate.
According to the analysis done in Chapter 5, in an EPL polling procedure the poll response time is, in the worst case, $23\mu s$, hence, a PRes Timeout value of $30\mu s$ should be an adequate choice. However, using this value of PRes Timeout in the B&R network the EPL protocol shows an unstable

behaviour: sometimes the node is correctly polled, thus the PRes frame is received by the MN before the PRes timeout, but some other times it happens that the node is considered to have failed. Conversely, the EPL protocol of an openPOWERLINK network always fails since the PRes frame is never received by the MN within the timeout.

As can be seen in Figure 6.9, when the PRes Timeout is too smaller there is the following sequence of packets:

SoC - PReq - SoA - PRes

this implies that the MN begins the Asynchronous period, with the SoA frame, before the reception of the PRes frame.

No.	Time	Source	Destination	Protocol	Length	Info
2	0.048693000	30:8d:99:f2:ca:e9	EPLv2_SoC	EPL	36	SoC
3	*REF*	30:8d:99:f2:ca:e9	IntelCor_1c:38:bd	EPL	60	PReq dst = 140 RD = 1 size = 0 ver = 0.0
4	0.000033000	30:8d:99:f2:ca:e9	EPLv2_SoA	EPL	54	SoA
5	0.001334000	IntelCor_1c:38:bd	EPLv2_PRes	EPL	60	PRes src = 140 RD = 1 size = 0 ver = 0.0
6	0.049951000	30:8d:99:f2:ca:e9	EPLv2_SoC	EPL	36	SoC
7	*REF*	30:8d:99:f2:ca:e9	IntelCor_1c:38:bd	EPL	60	PReq dst = 140 RD = 1 size = 0 ver = 0.0
8	0.000034000	30:8d:99:f2:ca:e9	EPLv2_SoA	EPL	54	SoA
9	0.001338000	IntelCor_1c:38:bd	EPLv2_PRes	EPL	60	PRes src = 140 RD = 1 size = 0 ver = 0.0
10	0.049956000	30:8d:99:f2:ca:e9	EPLv2_SoC	EPL	36	SoC
11	*REF*	30:8d:99:f2:ca:e9	IntelCor_1c:38:bd	EPL	60	PReq dst = 140 RD = 1 size = 0 ver = 0.0
12	0.000032000	30:8d:99:f2:ca:e9	EPLv2_SoA	EPL	54	SoA
13	0.001334000	IntelCor_1c:38:bd	EPLv2_PRes	EPL	60	PRes src = 140 RD = 1 size = 0 ver = 0.0

Figure 6.9: Wrong EPL communication cycle because of a small PRes Timeout.

In order to find the PRes Timeout value effectively needed in our implementation, several experimental sessions have been carried out for each network for different cycle time values, each one comprising more than 100 s of test. Then, we analyzed, among the received packets of a third PC able to monitor all the packets transmitted over the network, the period of time between the PReq frame and the PRes frame.
Particularly, a B&R network is tested with a cycle time value of 200 μs and 20 ms, instead an openPOWERLINK network is tested with 4 ms and 50 ms. It is worth remarking that the poll response time does not depend on the cycle time, whereas it depends only on the device nature. Moreover, we underline that in the considered case of one MN and one CN the polling duration is exactly the polling procedure of one CN.
Figures 6.10 shows the values of time between the PReq frame and the PRes frame for a B&R network, whereas Figure 6.11 shows the same values in the case of an openPOWERLINK one.
Focusing on the percentage of poll response, the Figure 6.10 highlights that the polling durations are mainly in the interval [5 , 9] μs. Moreover, as can be observed, each configuration has a most probable value of polling duration.

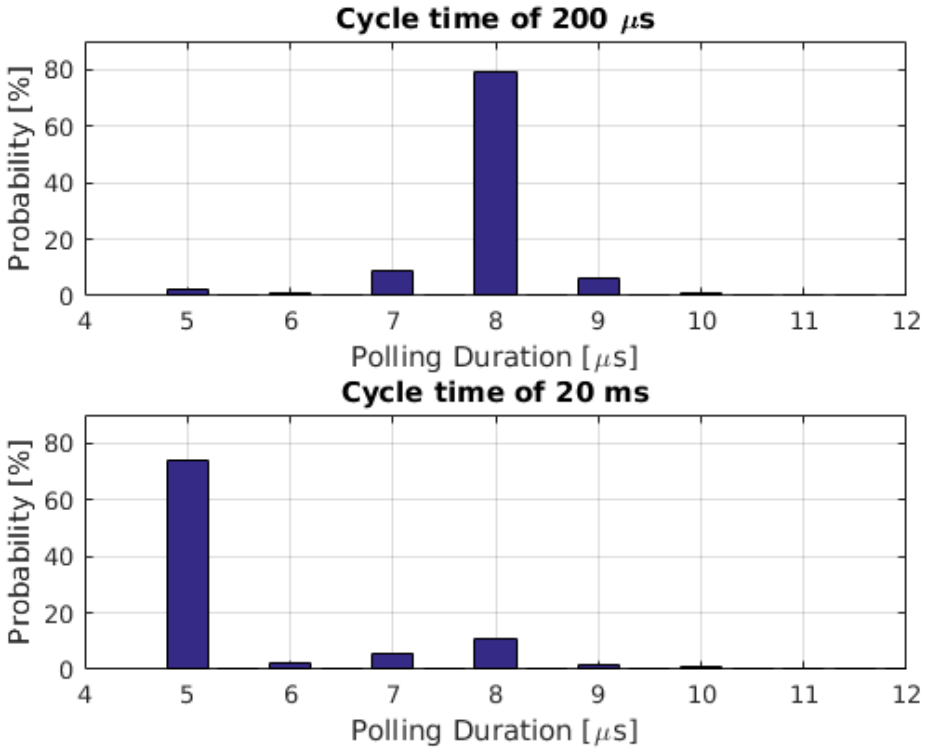


Figure 6.10: Histogram of the percentage of the poll response durations in a B&R network with a cycle time of 200 μs or 20 ms.

If the cycle time is set to 200 μs the most probable value, with a 80% of probability, of polling duration is clearly 8 μs; correspondingly if the cycle time is set to 20 ms the most probable value, with a 74% of probability, is 5 μs. Finally, as a more interesting result we considered the maximum value of the polling duration, it results 69 μs in the first case and 73 μs in the second case.

These experimental measurements are consistent with the observed unstable behaviour of the EPL protocol. Precisely, now we can conclude that, using 30 μs as PRes Timeout, almost always there will be a successful communication. Figure 6.11 summarizes the polling duration for a cycle time of 4 ms and 50 ms. Focusing on the percentage of poll response, the Figure highlights that the polling durations are mainly in the interval [1.04 , 1.11] ms. Moreover, the Figure emphasizes that each configuration has a set of probable values of polling duration. If the cycle time is set to 4 ms the most probable value, with a 20% of probability, of polling duration is in [1.08 1.09]; correspondingly

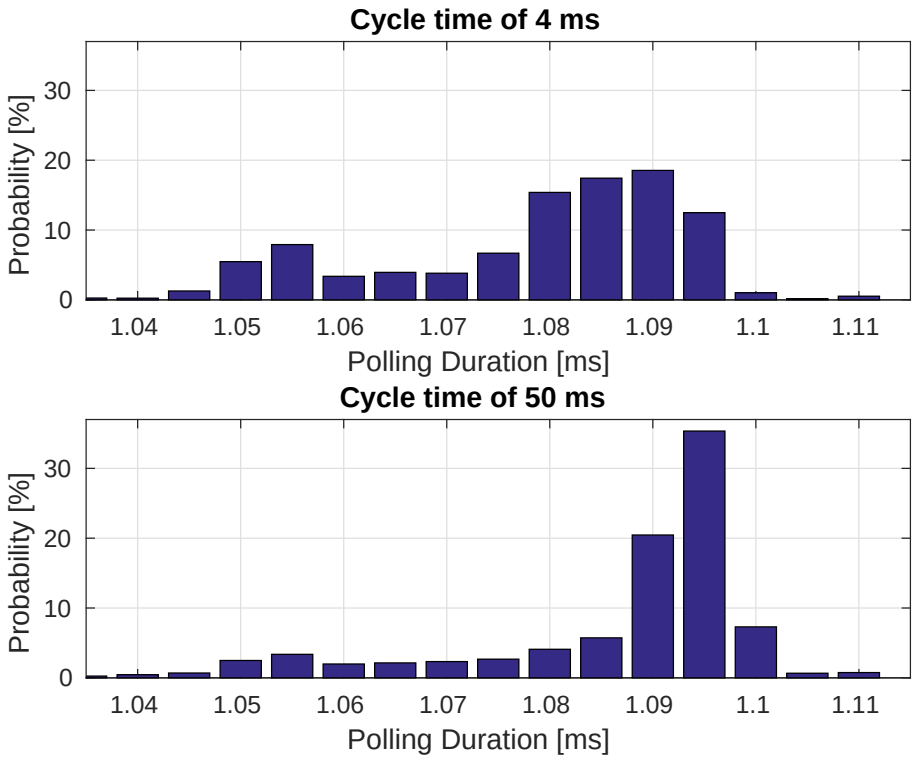


Figure 6.11: Histogram of the percentage of the poll response durations in an openPOWERLINK network with a cycle time of 4 ms or 6 ms.

if the cycle time is set to 50 ms the most probable value, with a 35% of probability, is 1.095 ms. Finally, as a more interesting results we considered the maximum value of the polling duration, it results 1.2 ms in the first case and 1.3 ms in the second case.

These experimental measurements are consistent with the claimed behaviour of the EPL protocol. Precisely, now we can conclude that, using 30 μs PRes Timeout, almost never there will be a successfully communication.
The results of Table 6.2 stress the fact that, in general, the cycle time does not influence the polling duration. Theoretically, the polling time value has to be exactly the same for each cycle time value. However, we observe two or more different most likely values of polling time (the difference is only of few μs), this is due to the few time-stamping precision of Wireshark.

Devices	Cycle time	Most likely	Max
B&R	200 μs	8 μs	69 μs
	20 ms	5 μs	73 μs
openPOWERLINK	4 ms	1.09 ms	1.2 ms
	50 ms	1.095 ms	1.3 ms

Table 6.2: Statistics of the polling duration in a network composed by B&R devices or openPOWERLINK devices.

Moreover, comparing the values summarized in Table 6.2 it is evident that the performance figures of both the cycle time values of an openPOWERLINK network are worse than those of a B&R network. Specifically, while the most probably polling duration of a B&R network is comparable with the ones obtained during the theoretical analysis, the openPOWERLINK results are, instead, considerably higher. Moreover, we can make out that the poll response values of a B&R device are mainly concentrate on one value, whereas the values of a openPOWERLINK device, are spread on an interval. This behaviour is again due to the hardware composition of the network.
Furthermore, according to this observation, in general the choice of the PRes Timeout will be a trade-off between the maximum value and the most probably one. In the first case the CN is always correctly polled at the expense of longer EPL cycles, on the contrary with a timeout value equal to the average Poll response time we may have smaller cycle times at the expense of some failed cycles.
To conclude, in general, the choice of the value of the Poll response timeout depends firstly on the hardware nature, and secondly on the resulted choice between reliability and speed.

6.1.6 Experiment: Mixed networks

In this experiment two EPL networks both composed by one MN and one CN are used. Particularly, in this experiment we will use:

- Mixed 1: network composed by an openPOWERLINK MN and a B&R CN.
- Mixed 2: network composed by a B&R MN and an openPOWERLINK CN.

The aim of this experiments is to recognize the difference of performances between these mixed networks w.r.t. the network considered before. In detail, we will observe the minimum cycle time which can be successfully used. Since the B&R devices has better performance, we expect that a Mixed 2 network will have better performance than an only openPOWERLINK based one.
We conducted some experimental measurements using the two mixed networks and varying the cycle time value among the interval of values [200 μs, 4 ms]. Table 6.3 summarizes the former results, with only B&R or openPOWERLINK network, and the results of this experiment.
In our implementation, both Mixed 1 and Mixed 2 networks have the same performances of an only openPOWERLINK based network. As a matter of fact. they are able to achieve a stable minimum cycle time value of 4 ms.

Cycle Time	B&R dev	openPOW dev	Mixed 1	Mixed 2
$200\mu s$	$\surd$	not available	not available	$\times$
$400\mu s$	$\surd$	$\times$	$\times$	$\times$
$3000\mu s$	$\surd$	$\times$	$\times$	$\times$
$3500\mu s$	$\surd$	unstable	unstable	unstable
$4000\mu s$	$\surd$	$\surd$	$\surd$	$\surd$

Table 6.3: EPL protocol operation state w.r.t. the cycle time value.

The results of Table 6.3 underline the fact that, in general, the minimum cycle time achieved by a network depends on the presence of openPOWERLINK devices in the network. Indeed, the degradation of the network performance is strictly related to the presence of at least one device which is not targeted for the industrial communication.

6.2 Wireless EPL configuration

The goal of this second measurement campaign is to provide an experimental assessment of the capability to communicate through a wireless channel using the EPL protocol. In the experiments, IEEE 802.11n is used for the wireless segment. Moreover, a specific customization for industrial applications has been adopted and, finally, the dynamic rate adaptation algorithm RSIN, introduced in [8], has been exploited by all the wireless devices.

To this aim, it is required the actual implementation of the RSIN algorithm on the real devices, which wishes to transmit over the wireless channel, as well as the deployment of an adequate prototype network, where a real-time communication is developed using the POWERLINK protocol. Especially, the experimental session we provide are done using both the B&R devices and the desktop PCs, where it is possible to manually modify the ANSI C stack openPOWERLINK in order to enable a wireless segment.

6.2.1 Network description

As can be seen in Figure 6.12, in this work a one-level EPL network configuration is used, which comprises two segments, one wired and the other wireless. The EPL MN and a set of (up to 4) CNs are connected to the same Ethernet hub in the wired segment. Some additional controlled nodes, which constitute the EPL wireless extension, could be located on the wireless segment and they are referred as Wireless Controlled Nodes (WCNs). In our implementation we consider only one WCN. The wireless segment is configured in infrastructure mode, where one Linux PC behaves like an Access Point (AP), while the WCNs are IEEE 802.11 stations (STAs) associated to the AP. Particularly, the Linux PC acts both as wired CN (CN #B) and Ethernet/WLAN bridge. As can be seen, the MN, the WCN and the CN#B have to be openPOWERLINK based devices, whereas the CNs could be both openPOWERLINK devices and B&R devices. The network is designed to emulate an industrial configuration, where a controller node (the AP) is in charge of polling the attached sensors/actuators (the STAs).
The wireless extension of EPL implemented at the data link layer allows for the direct inclusion of the WCNs in the EPL cycle.
The data flow between the two segments takes place transparently by means of an Ethernet/WLAN bridge device. In particular, a PReq (carried by an Ethernet frame) originated by the MN towards a WCN, crosses the bridge where it is encapsulated in a IEEE 802.11 Protocol Data Unit. Then, the addressed WCN responds with the PRes via the reverse path. For pure software solu-

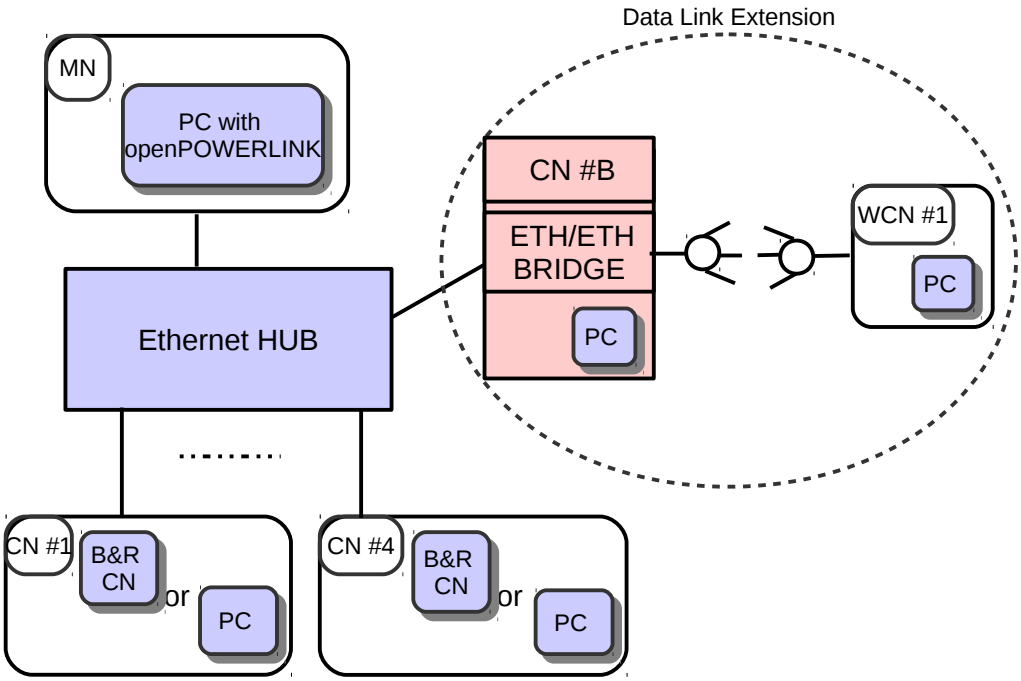


Figure 6.12: POWERLINK Wireless Extension at the data link layer using a bridge.

tions, POWERLINK is directly integrated on the application processor and uses a standard Ethernet controller as its bus connection.
In a further experimental session, we used the daisy chained configuration described in Figure 6.13. As can be seen, in this case an Ethernet/Ethernet bridge is adopted. Although such configuration clearly does not represent a wireless extension, it reveals particularly helpful to analyze the impact of the bride on the network performance.

6.2.2 Addressing

The EPL protocol defines the addressing system based on MAC address and Node ID, which is described in Chapter 2.3.5. Considering the wireless extension of the POWERLINK protocol we have to modify the default EPL addressing system to differentiate the IP addresses of WCNs and CNs. In other words we would like to recognize a wired node from a wireless one from the IP address. Since we can not use a different Net class, because all the IP

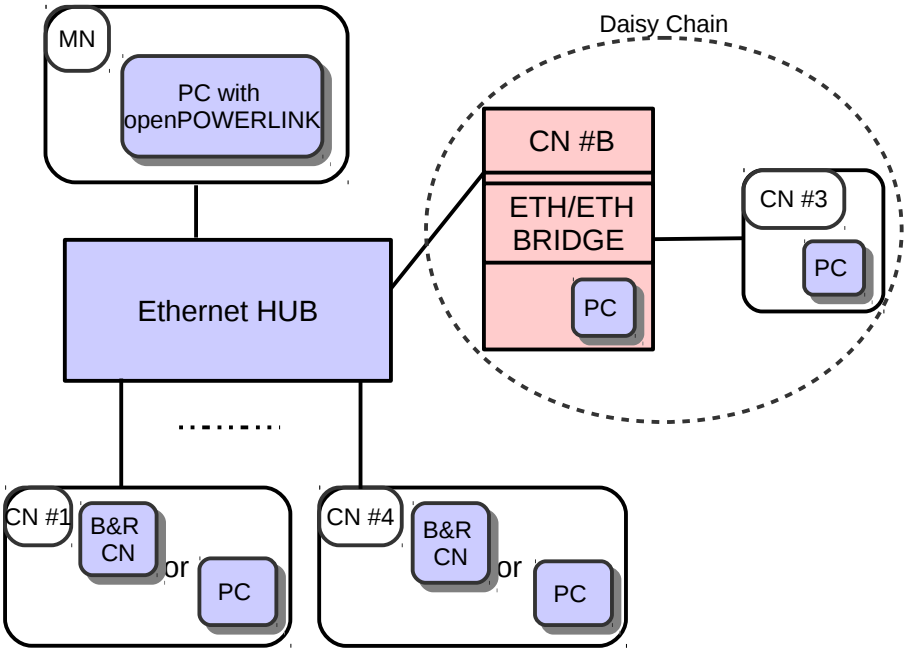


Figure 6.13: POWERLINK Extension using a daisy chained network.

addresses must belong to the same class 192.168.100.0, we put forward the following remedy based on the use of the subnet mask and the gateway. We recall that the EPL addressing rules allows the user to define a such elements, hence the solution is easily achievable.

As it is known, a class C TCP/IP network can be subnetted exactly into two parts by using a particular subnet mask, therefore, we can allocates a block of IP address for the wired CNs and the other for the WCNs. Using the subnet mask 255.255.255.128, our 192.168.100.0 network can be split in two networks:

$$\begin{aligned} \text{Net 1: } & 192.168.100.0 \\ \text{Net 2: } & 192.168.100.128 \end{aligned} \tag{6.1}$$

These two networks would have as valid host addresses:

$$\begin{aligned} \text{Hosts Net 1: } & 192.168.100.1 - 126 \\ \text{Hosts Net 2: } & 192.168.100.129 - 254 \end{aligned} \tag{6.2}$$

It is worth remembering that binary host addresses with all ones or all zeros

are invalid, so in our case we cannot use addresses with the last octet of 0, 127, 128 or 255. Under these assumptions we will have two independent networks, Net 1 and Net 2; one will be related to the wired nodes, and the other to the wireless ones.
When a subnet mask is associated to an IP address, it is used by the TCP/IP protocol to determine whether a host is on the local subnet or on a remote network.
Firstly, we use the MN IP address since the it manages the polling of the other nodes. Consequently, we can conclude that:

- since the MN IP address belongs to it, Net 2 (129-239) is the local subnet, thus it represents the wired IP addresses range.
- Net 1 (1-126) is the remote network, hence, it represents the wireless IP addresses range.

The MN can obviously communicate with the nodes that belong to Net 2, however, we must allows it to communicate with also the hosts of Net 1.
As it is known, if a TCP/IP node (the MN) needs to communicate with a host on another network (Net 1), it will usually communicate through a device called a router. In TCP/IP terms, a router that is specified on a host, which links the host's subnet to other networks, is called a default gateway. When a host attempts to communicate with another device using TCP/IP, it performs a comparison process using the defined subnet mask and the destination IP address versus the subnet mask and its own IP address. The result of this comparison tells the node whether the destination is a local host or a remote host. If the result of this process determines the destination to be a local host, then the node will simply send the packet on the local subnet. If the result of the comparison determines the destination to be a remote host, then the node will forward the packet to the default gateway defined in its TCP/IP properties. It is then the responsibility of the router to forward the packet to the correct subnet.
In our configuration the IP address of the MN default gateway has to be the same of the ethernet interface of desktop PC configured as bridge, i.e. eth0. Figure 6.14 summarizes the new EPL addressing rules needed to realize the wireless extension.
Moreover, the same subnet mask has to be applied also to the WCNs IP addresses, because they must answer to the MN back to front through the same network. Since each IP address of the WCNs belongs to the Net 1, the Net 2 is the remote network and a new gateway has to be defined. In our configuration the IP address of the WCNs default gateway has to be the same of the wireless interface of desktop PC configured as bridge, i.e. wlan0. Figure

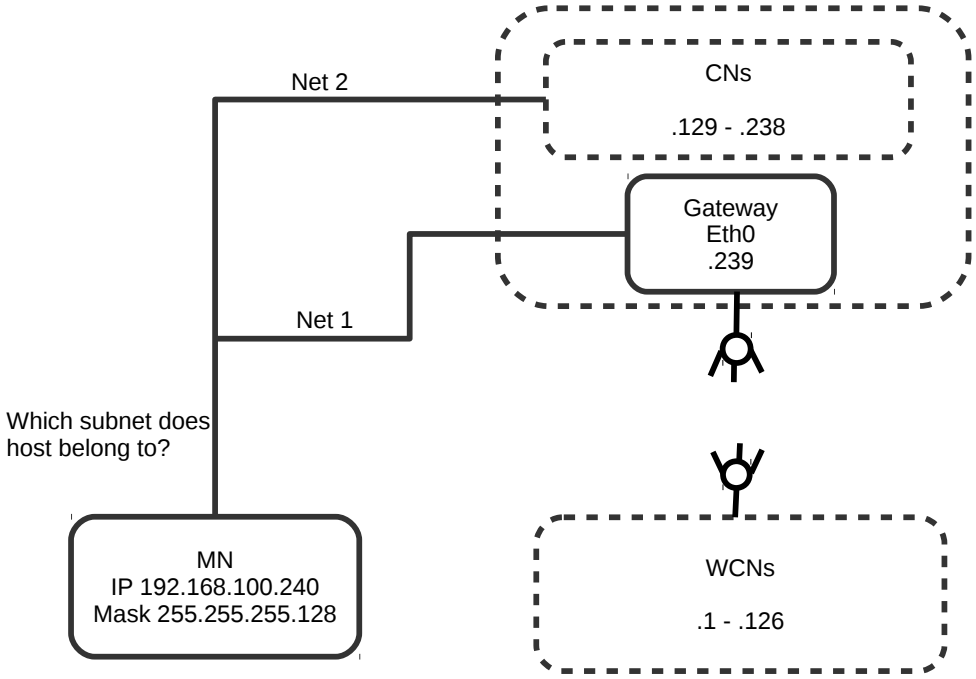


Figure 6.14: New EPL addressing rules for the MN for the EPL wireless extension.

6.15 summarizes the new EPL addressing rules needed to wireless extend the network.

6.2.3 Setup description

To begin, we have to manually configure the hardware of the wireless extension of this POWERLINK network.

openPOWERLINK MN The user has to configure the device according to the previous setup procedure, with few alterations in order to introduce the new EPL addressing rules for the MN.

Firstly, the user has to configure the MN network interface in Linux, choosing IP address: 192.168.100.240 and Subnet mask: 255.255.255.128. The process of creating a virtual network interface in Linux involves a single execution of the `ifconfig` command.

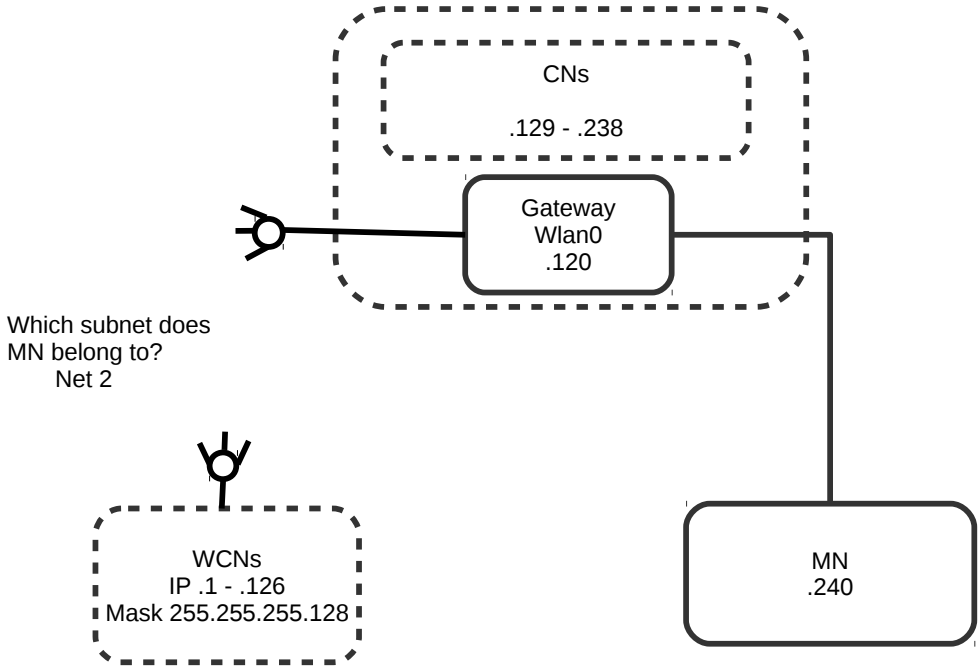


Figure 6.15: New EPL addressing rules for the WCNs for the EPL wireless extension.

Secondly, the user has to modify the variables SUBNET_MASK and DEFAULT_GATEWAY in the file `main.c`, which can be found in openPOWERLINK_V2.3.2/apps/demo_mn_console, before to create the executables of the demo application.

It is worth to pointing out that is this case the DEFAULT_GATEWAY value is: 192.168.100.239.

openPOWERLINK bridge The configuration procedure of this device can be divided into two principal parts, indeed the user has to characterize the network interfaces as well as to configure the wireless communication.

This device must have two physical network interfaces, the ethernet eth0 and the wireless wlan0, which will both operate as default gateway. Consequently, the user must configure these two interfaces in Linux, again using the `ifconfig` command. In our implementation, we configure the interfaces according to the following indication:

- Ethernet eth0
  IP address: 192.168.100.239
- Wireless wlan0
  IP address: 192.168.100.126

On this device the user has to create the interconnection between wired and wireless networks segments using a Linux bridge, which can built with the `brctl` package.

The wireless interface wlan0 can be turned into access points by using the `hostapd` (Host access point daemon) package. The user has to configure both the network parameter through the configuration file `hostapd.conf` and the RSIN parameters. Finally, the RSIN algorithm and the access point could be enabled.

Moreover, the device configured in such a way could also be a CN in the EPL communication. The user must define the Subnet mask: 255.255.255.128 for the eth0 network interface. Furthermore, the user has to modify the variables SUBNET_MASK in the file `main.c`, which can be found in openPOWERLINK_V2.3.2/apps/ demo_cn_console, before to create the executables of the demo application.

B&R CNs The user has to configure these devices according to the previous setup procedure, with the only recommendation about the Node ID which has to be taken among the wired IP address range apart from the Gateway IP address (129-238).

openPOWERLINK CNs The user has to configure these devices according to the previous setup procedure, with alteration, similarly to the MN, in order to introduce the new addressing rules due to the subnet mask. The user has to configure the CN network interface in Linux, choosing IP address among the wired IP address range apart from the Gateway IP address (129-238), and the Subnet mask: 255.255.255.128.

Secondly, the user has to modify the variables SUBNET_MASK in the file `main.c`, which can be found in openPOWERLINK_V2.3.2/apps/ demo_cn_console, before to create the executables of the demo application.

openPOWERLINK WCN The user has to configure the device according to the previous setup procedure, with few alterations in order to introduce the new EPL addressing rules for the WCNs.

Firstly, the user has to configure the CN network interface in Linux, choosing IP address among the wireless IP address range apart from the Gateway IP address (1-125), and the Subnet mask: 255.255.255.128.

Secondly, the user has to modify the variables SUBNET_MASK and DEFAULT_GATEWAY in the file `main.c`, which can be found in openPOWERLINK_V2.3.2/apps/demo_cn_console, before to create the executables of the demo application. It is worth to pointing out that is this case the DEFAULT_GATEWAY value is: 192.168.100.126.

The user has to configure and enable the RSIN algorithm and then the wireless interface wlan0 can be connected to the access points, created by the bridge, by using the `iwconfig` command.

Daisy Chained network A daisy chained network is composed by the MN and one openPOWERLINK bridge and CN.

The MN set up is equal to the one shown for an EPL wireless network, whereas, the openPOWERLINK bridge and CN have configurations similar to the one just described skipping over the steps concerning the wireless interfaces. Especially, in order to set up the bridge, the user has to configure the network interfaces eth0 and eth1 (instead of wlan0), and add them to the Linux bridge.

- Ethernet eth0
  IP address: 192.168.100.239
- Wireless eth1
  IP address: 192.168.100.126

6.2.4 IEEE 802.11n & RSIN Parameters

All the experiments will be carried out on an IEEE 802.11n network, where the configuration parameters were set according to the analysis provided by [8]. The main network parameters adopted in these tests are summarized in Table 6.4.

Description	Value
Channel Frequency	2.4 GHz
Channel width	both 20 and 40 MHz
Channel number	13
Modulation and Coding Schemes (MCS)	0-7
Transmission rates	13.5, 27, 40.5, 54, 81, 108, 121.5, 135 Mbit/s
Deadline D	variable
Playload size	64 bytes
Max retransmission attempts N_{max}	7

Table 6.4: IEEE 802.11n & RSIN parameters.

Figure 6.16 shows the configuration file of the access point, `hostapd.conf`. The `Hostapd` packet allows to configure the access point as an unique service set identifier (SSID) according to the configuration parameters of Table 6.4.

The real devices liable for the wireless communication could adopt several rate selection algorithms, thanks to the Multirate Support option. In this work of thesis we have taken into consideration the widespread Minstrel RA algorithm [42] which is commonly adopted by several general purpose WLAN devices. We are not interested on the behaviour of this algorithm, it is only used to underline why it is necessary the introduction of a real-time targeted algorithm. Since Minstrel was not designed for industrial application, as it will be underline in the following experiment, we will use the RSIN technique, introduced by [8].
In general, at the beginning of a packet transmission procedure, the RA algorithm has to provide a list of the rates to be used for each subsequent transmission attempt. In the case of RSIN, such a list resulted from the solution of the optimization problem. We remark in brief how the RSIN technique works. Given a packet to be transmitted within a deadline D, a specific transmitter-receiver pair and a set of transmission rates, the optimization problem is to find the number of attempts and the relative sequence of rates to be used for the transmission of this packet, with the twofold aim of minimizing the residual packet error probability and ensuring the deadline is inviolate. In the analysis of paper [8], an estimation of the algorithm processing delays is 50 μs, therefore the computation burden of the RSIN algorithm may impact on the

performance of the stations that use it.
The main RSIN configuration parameters are: the set of transmission rates, the playload size of the IEEE 802.11n frame, the maximum number of retransmission N_{max} and the deadline D. Table 6.4 summarizes their values. The first tree terms are fixed, in detail they are set accordingly to the configuration of [8]. In the theoretical analysis done in Chapter 5 we have defined D as the maximum frame delivery time of a wireless transmission. In other words D is the period between the instant in which an IEEE 802.11 packet starts to be transmitted and the instant in which the transmitter receives the correspondent ACK. Therefore, D is computed considering the worst case of successful transmission over the wireless channel, thus when N_{max}=7 unsuccessful retransmission are needed. The value of this parameter will be varied during the experiment.

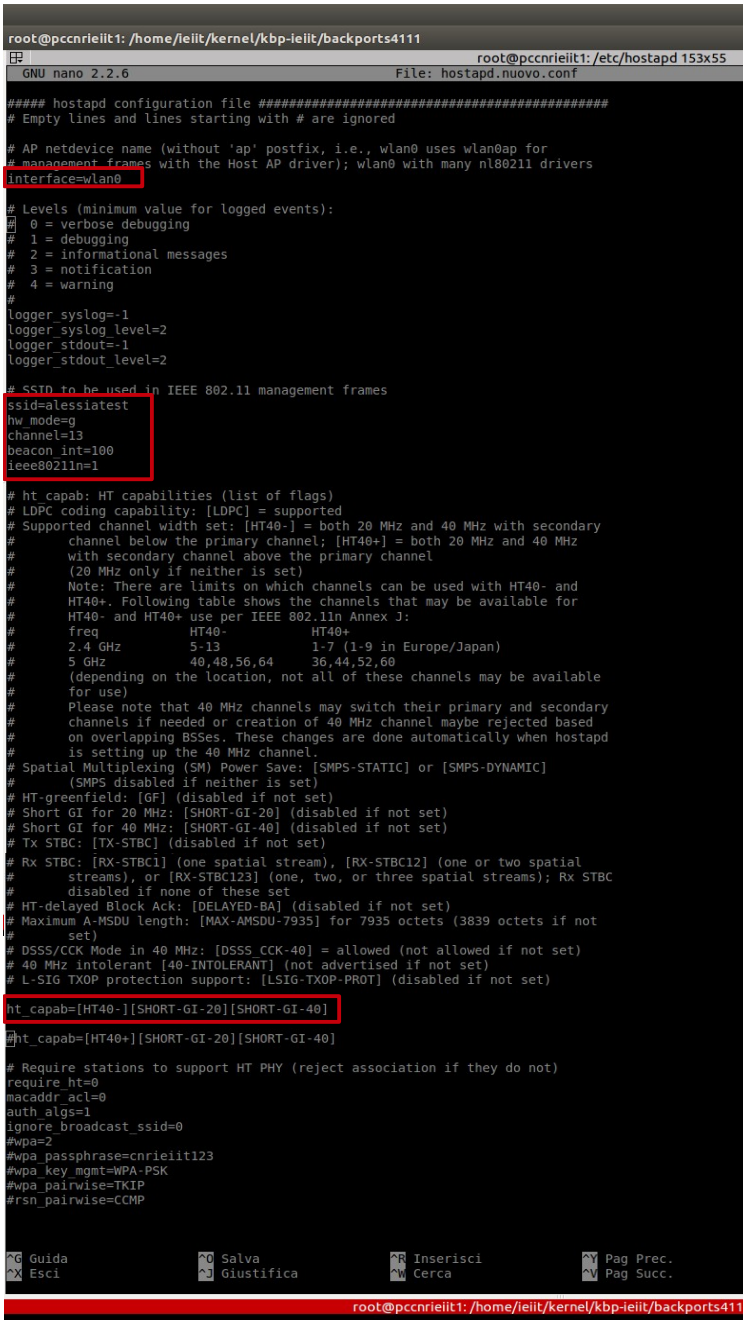


Figure 6.16: Hostapd configuration interface.

6.2.5 Experiment: Subnet Mask

In the first experiment an EPL network composed by the MN and one wired CN is used, where both the MN and CN are openPOWERLINK based devices. In particular, a subnet mask is enabled in both the devices and moreover, the CN ID Node is selected among the wired hosts range, i.e. 129-238.
The aim of this experiment is to verify the possibility of effectively setting up such a network, and verify if the subnet mask introduces a degradation of the EPL performance. In detail, we will pick out the minimum cycle time which can be effectively reached, thus for which the EPL protocol will not have an unstable behaviour. Theoretically, such a network may reach the same performance of the one without the subnet mask configuration. In order to verify if this is true, we test the network.

As a result of the a first experimental session we have pointed out that such a network could not be implemented using B&R CNs. Indeed, the EPL communication fails for each value of cycle time. It follows that in order to use a subnet mask the network must be assembled only with openPOWERLINK based devices, both the MN and the CNs. This is probably due to the fact that we can not modify the B&R EPL protocol adding the sub netmask, hence, different subnet mask configurations cause errors in the EPL communication.
Turning to the performance, the subnet mask does not influence the minimum cycle time which ensures a correctly communication among the network stations. In detail, as can be seen in Table 6.5, the EPL protocols works properly with a minimum cycle time value of 4 ms.
In conclusion, an EPL network configured using the subnet mask has the same performance of the one configured without it, provided that all the station network interfaces are properly configured.

Cycle Time	Normal	Subnet Mask
$200\mu s$	not available	not available
$400\mu s$	$\times$	$\times$
$3000\mu s$	$\times$	$\times$
$3500\mu s$	unstable	unstable
$4000\mu s$	$\surd$	$\surd$

Table 6.5: EPL protocol operation state w.r.t. the cycle time value.

6.2.6 Experiment: Daisy Chain

In this experiment a daisy chained EPL network composed by the MN and a wired CN connected in daisy chain is used, as shown in Figure 6.13.
The aim of this experiment is to analyze the impact of the Ethernet/Ethernet bridge on the network performance. In detail we will observe the cycle time and the polling time. Since the daisy chained network introduces delays, due to the presence of the bridge and of more cable, we expect that it will have worse performance than the previous network, thus higher minimum cycle time and polling time.
To begin, we performed some experimental measurements varying the cycle time value among the interval of values [4 ms, 8 ms]. Table 6.6 summarizes the former results, with a normal openPOWERLINK network and a network where the subnet mask is configured, and the results of this experiment.

Cycle Time	Normal	Subnet Mask	Daisy Chain
$4000\mu s$	$\surd$	$\surd$	$\times$
$7000\mu s$	$\surd$	$\surd$	unstable
$8000\mu s$	$\surd$	$\surd$	$\surd$

Table 6.6: EPL protocol operation state w.r.t. the cycle time value.

In our implementation a daisy chained EPL network achieves a minimum cycle time value of 8 ms. This behaviour is mainly due to the presence of the bridge, which introduces a delay due to elaboration of data. For this reason, an EPL daisy chained network works properly only if the cycle time is roughly twice the one of a normal EPL network.

Then, we analyzed, among the received packets of a third PC able to monitor all the packets transmitted over the network, the period of time between the PReq frame and the PRes frame. Especially, we tested an EPL daisy chained network configured with a cycle time value of 8 ms, however, as we have proved, the poll response time does not depend on the cycle time. Therefore, this configuration allows an analysis of the polling performance, without sacrificing the generality of the obtained results.
Figure 6.10 shows an histogram of the values of time between the PReq frame and the PRes frame for such a network. Focusing on the percentage of poll response durations, the Figure highlights that they are mainly in the interval [1.5 , 1.8] ms. Differently from results of 6.11, the Figure shows that there are more than one most probable value, indeed, the polling duration is included,

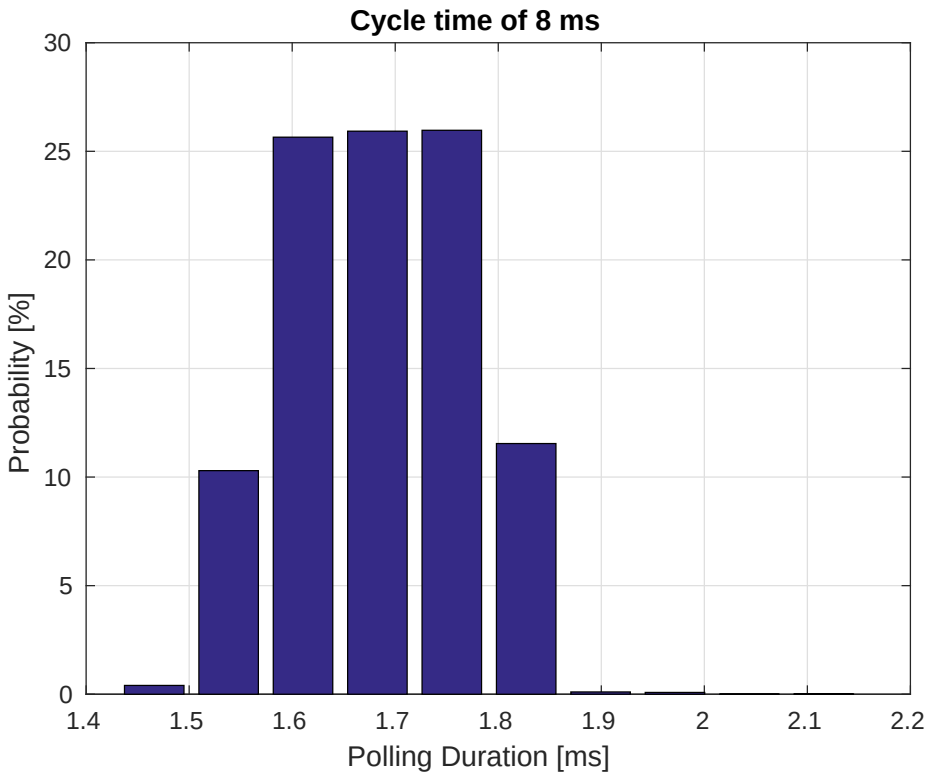


Figure 6.17: Histogram of percentage of the poll response duration in a EPL daisy chained network with a cycle time of 8 ms.

with an equally 25% of probability, among the interval [1.6, 1.75] ms. Finally, as a more interesting result, we considered the maximum value of the polling duration and the mean value, they respectively result 2 ms and 1.7 ms.

Network	Most likely	Max
Normal	1.09 ms	1.2 ms
Daisy Chained	1.6-1.75 ms	2 ms

Table 6.7: Statistics of the polling duration in EPL daisy chained network.

Comparing the results of Table 6.7 we can conclude that in our implementation the Ethernet/Ethernet bridge introduces a delay of roughly 0.7 ms in the polling time. It is worth pointing out that in the theoretical analysis we setted a bridging delay value of 10 μs, therefore, this is one motivation of why the theoretical values underestimate the real ones.

In conclusion, we validated the possibility to use a desktop PC both as ETH/ETH bridge and as a normal CN #B in the EPL communication. The experimental result highlights that this activity do not introduce further delays, indeed, the CN #B behaves as a normal wired openPOWERLINK CN, and the performance of the network remains invariant.

6.2.7 Experiment: Wireless Communication

In this experiment a wireless EPL network composed by the MN and one WCN is used. This experiment is pursued using an IEEE 802.11n network and the default Minstrel rate adaptation algorithm [42]. This workstation, developed in a prototype network, allows an effective control of the devices and of the wireless medium, without sacrificing the generality of the obtained results.
The aim of this experiment is to remark how the performance figures of a traditional EPL network are influenced by the introduction of the wireless communication.
During this experiments we validate the analysis done in [14], which individuates the main problem in the retransmission procedure of IEEE 802.11n. As can be seen in Figure 6.18, if during the query of a wireless CN, the MN does not receive the PRes frame within the Poll response time-out, then it moves on the asynchronous period. However, if the N_{max} on the node, that issued the relevant IEEE 802.11n frame, is not exceeded, then, at the data link layer of that node the transmission will continue; ever after the EPL polling time-out has expired. As a consequence, the PRes frame is received during one of the following cycles and the forthcoming and devastating consequence is the collapse of the EPL communication.
Figure 6.19 shows another case of critic retransmission due to the IEEE 802.11n protocol. Similarly to the PRes, also the ASnd frame could be retransmitted exceeding the asynchronous timeout. As a consequence, the ASnd frame is received during one of the following cycles and the consequence is again the collapse of the EPL communication. Since these critic cases are often present during the the experimental measurements, we can conclude that a wireless implementation based on IEEE 802.11n and Minstrel RA algorithm is not adequate to ensure a successful EPL communication.

A possible compromise to this problem is introduced and validated in the analysis done in [13]: the retry limit N_{max} has to be set in such a way that, when it is reached, the Poll response time-out must not be expired yet. Especially, the maximum time requested by N_{max} unsuccessful polling attempts of a WCN must be smaller than the EPL PRes timeout of a fixed quantity, which is the sum of the node latencies and the wired communication delays. Clearly,

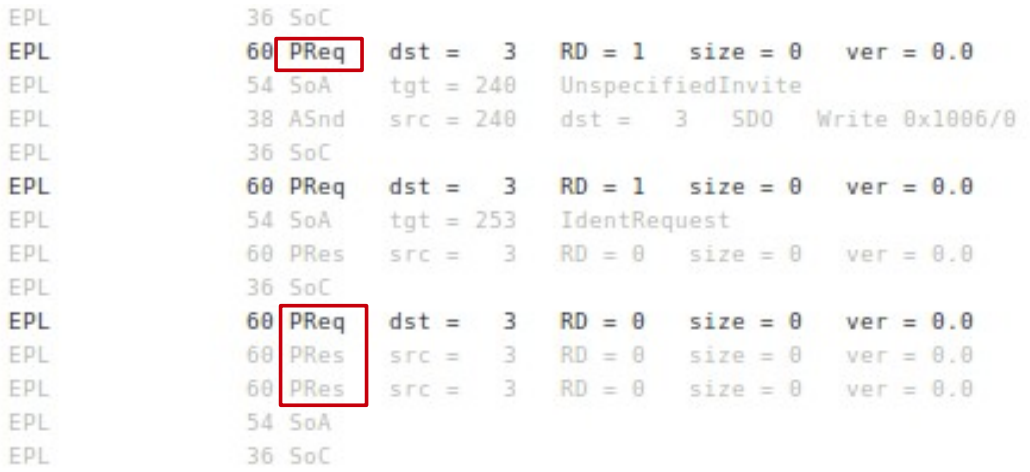


Figure 6.18: First example of inusual beaviour in the EPL communication due to the instrinsec randomness of IEEE 802.11.

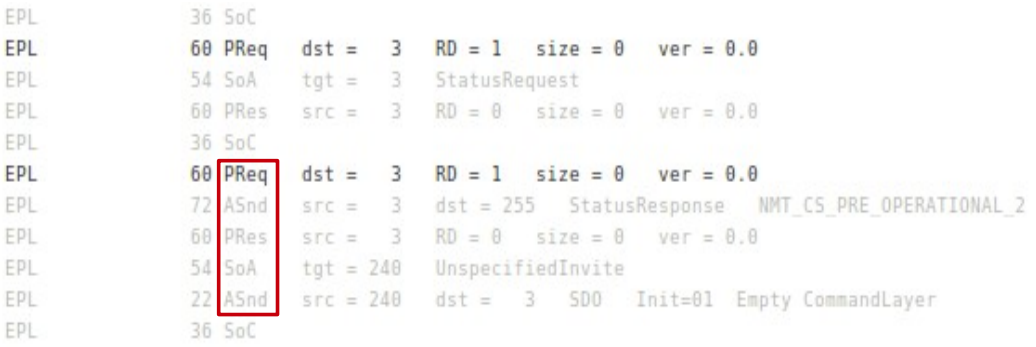


Figure 6.19: Second example of inusual beaviour in the EPL communication due to the instrinsec randomness of IEEE 802.11.

the lower the retry limit, the greater the probability of packet loss during the communication and, hence, of polling failures. However, since the EPL protocol is able to recover from polling failure, indeed it resumes its operation correctly, this compromise seems to be reasonable.

6.2.8 Experiment: Wireless Communication & RSIN

In this experiment a wireless EPL network composed by the MN and one WCN is used. This experiment is carried out on a IEEE 802.11n network which use the real-time tailored RSIN rate adaptation algorithm, introduced by [8].

In this work of thesis we suggest the use of RSIN RA algorithm as a solution to the retransmission problems detected in the previous experiment. In particular, our solution to the intrinsic randomness of the IEEE 802.11n is to harmonize the RSIN parameter D with the EPL parameter PRes timeout. There is a strictly relationship between D and N_{max}. However, an appropriate setting of D allows to ensure no collapse of the EPL communication even if N_{max} is set to high values, ensuring a lower probability of packet loss. It follows that, our propounded solution and the one proposed and tested in [13] are comparable, moreover our seems a more promising alternative.

The aim of this experiment is firstly to provide an experimental validation and a performance assessment of the voted solution to implement a wireless extension of the EPL protocol. In detail, we will evaluate both the minimum cycle time and the polling duration.

To begin, we performed some experimental measurements in order to check whether RSIN actually solves the retransmission problem observed in the previous experiment. After an accurate analysis of the results, we were able conclude that the problem never occurred when RSIN was used.

Then, we conducted several experimental measurement in order to find a cycle time value that corresponds to a stable EPL communication. However, the communication seems to be extremely disturbed even if the environmental conditions are good. Even if, for some high cycle time values the EPL communication is more stable, the promising results of the previous experiment were never achieved. Unfortunately, both Wireshark trace and POWERLINK operation sequence do not provide an noticeable explanation to the presence of errors. Indeed, they only shows the re-configuration of the node as a consequence of an several errors.
However, as it is shown in Figure 6.20, a strange behaviour was recorded, namely the periodic presence of errors in the communication, that suggests to think that it is an EPL protocol fault. However, in order to validate this hypothesis further investigations about the wireless communication have to be carried out, which will be a purpose for the future works in this framework.

In the theoretical analysis, we founded a mathematical worst case value of the RSIN deadline, D, nevertheless both the experimental networks need a smaller deadline value in order to correctly communicate. According to the analysis done in Chapter 5, the polling procedure of a wireless node WCN

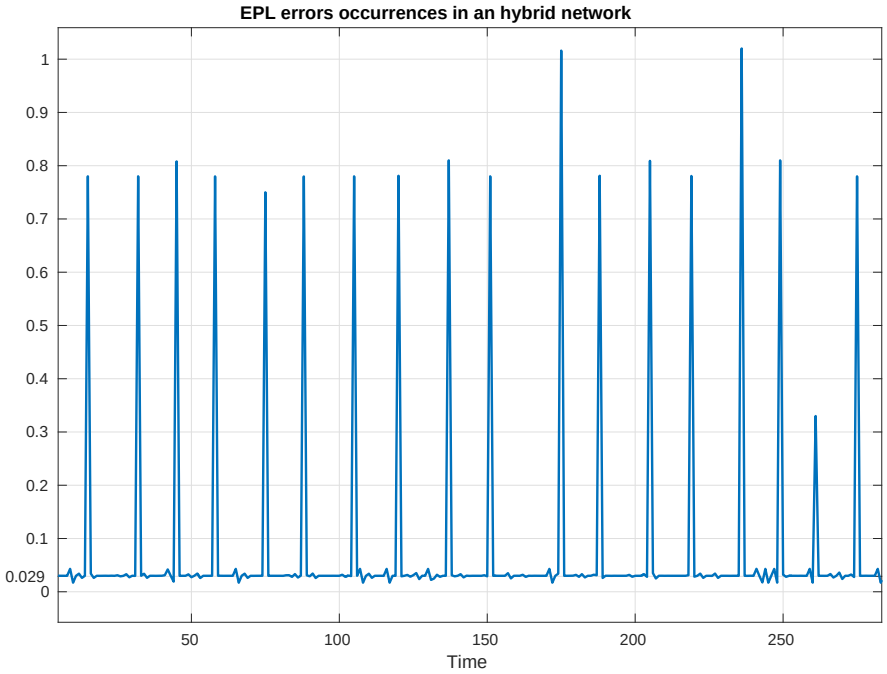


Figure 6.20: Inter times beetwen the restarts of the EPL protocol for the basic network considered.

takes, in the worst case, 20.36 ms, hence, a PRes Timeout value of 27 ms and a deadline D value of 8 ms should be adequate choices. It is worth remarking that this PRes timeout value is adequately greater than the maximum value found during the EPL daisy chained experimental measurement (i.e. 2 ms). In order to find the PRes Timeout value effectively needed in our implementation, several experimental sessions have been carried out for this basic network. We analyzed, among the received packets of the MN, the period of time between the PReq frame and the PRes frame.
Figure 6.21 shows the histogram of the results for such a network. Focusing on the percentage of poll response durations, the Figure highlights that the most likely value, with the 61% of probability, is 2.4 ms. Therefore, the polling of a wireless node requests a lower time than that expected in the theoretical case. This is mainly due to the fact that in the analysis we considered the worst-case value, which is quite unrealistic. Indeed, the experimental measurements are taken in a controlled environment and the distance between the devices is three meters, therefore, the occurrence of many retransmissions is unlikely.
In conclusion, we validated the possibility to use a desktop PC both as ETH/WLAN bridge and as a normally CN #B in the EPL communication.

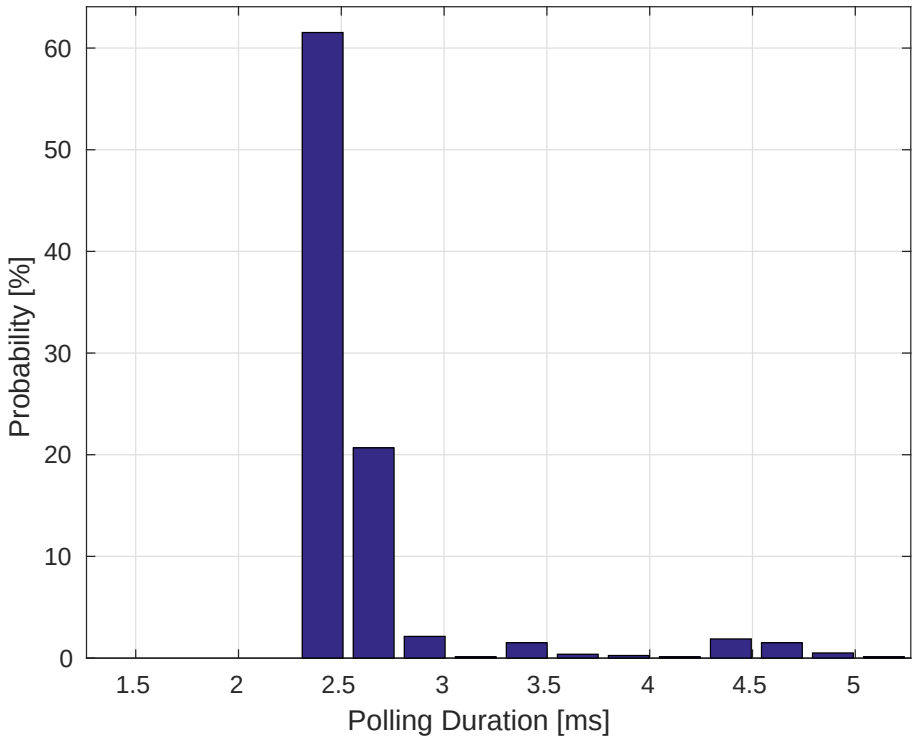


Figure 6.21: Histogram of percentage of the poll response duration in a wireless EPL network with a cycle time of 50 ms.

The experimental result highlights that this activity does not introduce further delays, indeed, the CN #B behaves as a normal wired openPOWERLINK CN, and the performance of the network remains invariant.

6.3 Review of the principal results

Several configurations of EPL networks, both wired and wireless, are constructed using the B&R real-time hardware and the PCs configured with the open source POWERLINK stack. In detail we had considered the following networks whilst they are communicating using the EPL protocol: only B&R networks, only openPOWERLINK networks, mixed networks; moreover, daisy chained networks and wireless network. These EPL networks have been subject to a series of experimental campaigns, in order to assess their performance and compare them with the results of the theoretical analysis conducted in Chapter 5.

Firstly, we investigate the minimum cycle time for which the EPL protocol does not present an unstable behaviour. The outcomes of the experiments showed that, apart from wireless network, the EPL protocol works well in each network configuration. Indeed, a successfully EPL communication could be performed with cycle time values acceptable w.r.t. the real-time requirement. Table 6.8 summarizes all the performances about the cycle time carried out during this Chapter. Generally, more complex the network is, greater the minimum cycle time has to be. It is worth pointing out that the network realizing the EPL wireless extension does not achieve acceptable performance for any cycle time value.

Typology of Network	Minimum Cycle time
only B&R network	0.2 ms
only openPOWERLINK network	4 ms
Mixed network	4 ms
Network with the subnet mask	4 ms
Daisy chained network	8 ms
Wireless network	never

Table 6.8: Principal results about the minimum cycle time achievable w.r.t. the typology of the network.

Secondly, we investigate the difference, concerning the polling duration w.r.t. the polled device. The outcomes of the experiments we performed are very encouraging, since they showed that not only the device tailored to industrial communication has good performance, indeed the ones realized by openPOWERLINK introduce more or less big delays but they can still be considered

reliable for a real-time communication. Table 6.9 summarizes all the results regarding the mean value of the Polling duration w.r.t. the typology of the node.

Typology of Node	Mean value of the Polling Duration
B&R	0.007 ms
openPOWERLINK	1.09 ms
Daisy chained	1.7 ms
Wireless	2.4 ms

Table 6.9: Principal results about the mean value of the Polling duration w.r.t. the typology of the node.

Chapter 7

Conclusion

In this paper we considered Ethernet POWERLINK, which is one of the most popular Real-Time Ethernet networks currently available.
As a first relevant aspect, the EPL protocol has been thoroughly analyzed, both from the hardware perspective, with a detailed description of all its components, and from the software perspective, with an overview of the adopted communication protocol. The time required for the isochronous period has been carefully computed, with a particular attention to the polling time.
Several experimental sessions have been performed on the proposed system, in order to characterize its real performance figures and compare them to those obtained through the theoretical analysis. In particular, we are interested in evaluating the performance of an EPL network composed also by desktop PCs, where it possible to manually configure the EPL protocol using the openPOWERLINK stack.
The outcomes of the experiments we carried out showed that the minimum cycle time achieved depends on which hardware there is in the network; and particularly, the degradation of the network performance is strictly related to the presence of at least one openPOWERLINK based device. Moreover, the assessments expose the critical aspects of an openPOWERLINK implementation, the polling time of such a node is considerable greater than one of a commercial node. These results are mainly due to the hardware composition of the devices. Clearly, an hardware targeted to industrial communication allows very fast data processing; whereas, an openPOWERLINK device, implemented on a desktop PC is necessarily subjected to greater elaboration delays.

The goal of the second part of the thesis is to provide an experimental assessment of the capability to communicate through a wireless channel using EPL protocol.
Firstly, the time required for the transmission and elaboration of data through the wireless channel has been carefully computed, with a particular attention to the components that introduced randomness in the communication.
Several experimental sessions have been performed on an EPL wireless extension which simply use the 802.11n protocol, in order to characterize its real performance figures and compare them to those obtained through the theoretical analysis. Such a network can be implemented by customizing the open source protocol stack of POWERLINK and introducing a Linux ETH/WLAN bridge as interconnection between the wired and wireless segments.
Since simply using the 802.11n protocol its intrinsic randomness often cause the collapse of the EPL communication, the purpose of this work of thesis is to put forth a solution. Particularly, a specific customization for industrial application of IEEE 802.11n is used and, finally, the dynamic rate adaptation algorithm RSIN has been exploited. RSIN algorithm leverages on the knowledge of the wireless channel status and, on this basis, selects the sequence of transmission rates to be used for packet transmission by solving a constrained optimization problem. Since the main reason of errors is the retransmission procedure of IEEE 802.11n we advance to harmonize the RSIN deadline parameter, D, with the EPL PRes Timeout.
In a first experimental session we assess the impact of the bridge on the network performance, introducing a daisy chained configuration. An EPL daisy chained network works properly only if the cycle time is roughly twice the one of a normal EPL network.
Then, the performance figures of the proposed RSIN algorithm during an EPL communication have been compared with those of Minstrel, based on the outcomes of an extensive measurements campaign conducted on real devices, on which all the aforementioned RA technique has been implemented. The analysis has highlighted that the retransmission problem never occurred when RSIN was used.
The outcomes of further experiments we carried out shows that the communication of the hybrid EPL network seems to be extremely disturbed even if the environmental conditions are good. Nevertheless, for some cycle time the EPL communication is more stable, the promising results of the previous experiment are never achieved. Moreover, a strange behaviour was recorded, namely the periodic presence of errors in the communication, that suggests to think that it is an EPL protocol fault. However, in order to validate this hypothesis further investigations about the wireless communication have to be

carried out, which will be a purpose for the future works in this framework.

During the course of this work, several software and hardware tools have been used to achieve the aforementioned results. The greatest part of work was dedicated to the measurement campaigns and to the elaboration of the collected data, mainly the capture provided by the packet-sniffing software Wireshark. The data taken from the measurements have been subsequently elaborated with Matlab to produce the plots visible in Chapter 6. Matlab has also been used to perform the theoretical analysis in Chapter 5.

7.0.1 Future works

A further interesting development in the framework of the POWERLINK protocol can be outlined.
We plan to evaluate the behaviour of the protocol for more performing not industrial device, in order to come to a more complete validation of the reason of the degradation of the performances related to these devices.

Some further interesting developments in the framework of the wireless extension of the POWERLINK protocol can be outlined.
As a natural extension of the experimental assessments of the EPL wireless extension, we plan to analyze in more detail the errors, in order to find the correct explanation to the problem. From one hand, we can use the open-POWERLINK operations story, and characterize one by one the EPL errors. From the other hand, we can use a third PC able to monitor, using Wireshark, all the packets transmitted over the wireless channel and observe if there are corrupted packets. It is worth pointing out that in the previous analysis we used only Ethernet Wireshark trace.
Moreover, the behaviour of the bridge and a more complete characterization of its influence on the performance should be investigated. An accurate estimation of the delay introduced by the bridge has to be carried out in order to ensure a more precisely knowledge of the performance issue of the EPL wireless extension. Then it can be compared with the experimental results observed using the daisy chained configuration.
Furthermore, different interconnections between wired and wireless network segments should be tested.

A general interesting extension of this work, for both the first and the second part, is to allow the connection of multiple nodes both wired (CNs) and

wireless (WCNs). In detail, a further analysis could be for example what is the maximum number of client devices that can be connected to ensure good performance figures.
Finally, both the networks should be tested again on the real application environment, to see if the satisfactory results obtained in the research laboratory still hold.

Bibliography

[1] *Help Explorer of Automation Studio - B&R Automation*, 2016. v. 4.2.3.71.

[2] S. Vitturi, *Dispense del corso 'Laboratorio di Automazione Industriale'*. 2015.

[3] L. Seno, *Real-Time Networks and Protocols for Industrial Automation*. Ph.D. dissertation, 2011.

[4] *IEEE 802.3 Standard for Information technology - Telecommunications and information exchange between systems - - Local and metropolitan area networks - Specific requirements*, 2012.

[5] "EPSG website." `http://www.ethernet-powerlink.org/`.

[6] *IEEE 802.11 Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements*, 2012. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[7] F. Tramarin, S. Vitturi, and L. Seno, "Industrial Wireless Networks: The Significance of Timeliness in Communication Systems," *Industrial Electronics Magazine, IEEE*, 2013.

[8] F. Tramarin, S. Vitturi, and M. Luvisotto, "A dynamic rate selection algorithm for ieee 802.11 industrial wireless lan," *IEEE Transactions on Industrial Informatics*, vol. PP, pp. 1–1, October 2016.

[9] L. Seno and S. Vitturi, "A simulation study of ethernet powerlink networks," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, pp. 740–743, Sept 2007.

[10] A. Soury, M. Charfi, D. Genon-Catalot, and J. M. Thiriet, "Performance analysis of Ethernet Powerlink protocol: Application to a new lift system generation," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–6, Sept 2015.

[11] H. Xu, Y. Gao, K. Liu, B. Zhu, and C. Zhang, "Research on Cross-communication Based on Real-time Ethernet POWERLINK," *Chinese Control and Decision Conference*, 2014.

[12] M. Knezic, B. Dokic, and Z. Ivanovic, "Improving the flexibility of the ethernet powerlink pollresponse chaining mechanism," in *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, pp. 1–4, May 2016.

[13] L. Seno, S. Vitturi, and C. Zunino, "Analysis of Ethernet Powerlink Wireless Extensions Based on the IEEE 802.11 WLAN," *IEEE Transactions on Industrial Informatics*, vol. 5, pp. 86–98, May 2009.

[14] L. Seno and S. Vitturi, "Wireless extension of Ethernet Powerlink networks based on the IEEE 802.11 wireless LAN," in *Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on*, pp. 55–63, May 2008.

[15] S. Segantin, *Analisi delle prestazioni di una rete di comunicazione ibrida (wired/wireless) per applicazioni industriali real-time*. Master thesis, 2009.

[16] Y.-C. Li, S. H. Hong, X. Huang, G. Chen, and X. Liang, "Implementation of a powerlink-wirelesshart gateway for industrial automation," in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pp. 1–6, June 2016.

[17] *Ethernet Powerlink Communication Profile Specification*, 2016. Draft Standard 301, v1.3.0.

[18] "B&R Automation website." `http://br-automation.com/`.

[19] "IEC Internation standards webstore." `https://webstore.iec.ch/home/`.

[20] "CAN in Automation (CiA) website." `http://can-cia.org/`.

[21] L. Lachello, P. Wratil, A. Meindl, S. Schnegger, B. S. Karunakaran, S. H., and S. Potier, "Ethernet Facts: the Five Major Technologies System Comparison," *EPSG, Vol. 2, pp. 14-25*, 2013.

[22] S. Ding, P. Zhang, S. Yin, and E. Ding, "An Integrated Design Framework of Fault-Tolerant Wireless Networked Control Systems for Industrial Automatic Control Applications," *IEEE Transactions On Industrial Informatics*, 2013. vol. 9, no. 1, pp. 462-471.

[23] F. Tramarin, *Industrial wireless sensor networks - simulation and measurement in an interfering environment.* Ph.D. dissertation, 2011.

[24] N. Benvenuto and M. Zorzi, *Principles of Communications Networks and Systems.* Wiley, 2011.

[25] M. Luvisotto, *Analysis of an IEEE 802.11-based protocol for real-time applications in agriculture.* Master thesis, 2014.

[26] F. Cali', M. Conti, and E. Gregori, "Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEEE/ACM TRANSACTIONS ON NETWORKING*, 2000.

[27] S. Vitturi, L. Seno, F. Tramarin, and M. Bertocco, "On the rate adaptation techniques of ieee 802.11 networks for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 198–208, Feb 2013.

[28] F. Tramarin, S. Vitturi, and M. Luvisotto, "Improved rate adaptation strategies for real-time industrial ieee 802.11n wlans," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, pp. 1–8, Sept 2015.

[29] Perahia and Stacey, "Next Generation Wireless LANs: Throughput, Robustness, and Reliability in 802.11n," 2008.

[30] Perahia and Stacey, "IEEE 802.11n Development: History, Process, and Technology," *IEEE Communications Magazine*, 2008.

[31] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "On the Use of IEEE 802.11n for Industrial Communications," *IEEE Transactions on Industrial Informatics*, vol. 12, pp. 1877–1886, Oct 2016.

[32] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, "The ieee 802.11n wireless lan for real-time industrial communication," in *Factory Communication Systems (WFCS), 2015 IEEE World Conference on*, pp. 1–4, May 2015.

[33] F. Tramarin, S. Vitturi, and M. Luvisotto, "Enhancing the real-time behavior of ieee 802.11n," in *Factory Communication Systems (WFCS), 2015 IEEE World Conference on*, pp. 1–4, May 2015.

[34] F. Tramarin, S. Vitturi, M. Luivisotto, and A. Tagliapietra, "Performance Analysis of IEEE 802.11 Rate Selection for Industrial Networks," *IEEE*, 2016.

[35] *IEEE 802.1D Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges*, 2012.

[36] M. Bertocco, C. Narduzzi, and F. Tramarin, "Estimation of the delay of network devices in hybrid wired/wireless real-time industrial communication systems," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, pp. 2016–2021, May 2012.

[37] "Linux Fondation Wiki webstore." `https://wiki.linuxfoundation.org/`.

[38] "openPOWERLINK 2.3.2, An open-source POWERLINK protocol stack webstore." `http://openpowerlink.sourceforge.net/web/openPOWERLINK.html`.

[39] "CMake webstore." `http://www.cmake.org`.

[40] Kalycito and B&R, *User manual for Ethernet POWERLINK openCONFIGURATOR eclipse plugin*, 2016.

[41] 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*, 2006.

[42] "Minstel rate adaptation algorithm webstore." `https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel`.