

Is CSMA/CA really efficient against interference in a Wireless Control System? An experimental answer

M. Bertocco, G. Gamba, A. Sona

University of Padova

via G. Gradenigo, 6/B 35131 Padova, Italy,

Phone: +39 049 8277743, Fax: +39 049 8277699

{matteo.bertocco,giovanni.gamba,alessandro.sona}@unipd.it

Abstract

The deployment of a wireless control system must cope with a number of effects usually negligible in a wired scenario. To this aim, an experimental analysis on suitable prototypes and testbeds can readably offer a valid and rapid method to forecast the behavior of a real network. In this paper, performance measurement of industrial wireless sensor network in the presence of various type of interference is provided. In particular a Bluetooth, WiFi and Zigbee disturbing network impairs a suitable testbed employing IEEE 802.15.4 wireless sensor nodes. The purpose is to optimize the effectiveness of the medium access control mechanism through the choice of a proper mode of use. The analysis will show that from the measurement of certain parameters (number of failed pollings, experimental cycle-time, and alarm latency) interference effects can be effectively recognized and the network setup optimized.

1 Introduction

Wireless sensor networks (WSNs) are nowadays a promising and powerful novelty in the scenario of industrial communications [1]. They provide distributed sensing features having interesting properties with respect to the wired networks, such as the absence of infrastructures, low cost, scalability and flexibility. Nevertheless, some key drawbacks are actually delaying a wide deployment of these systems for industrial applications, due to some technical problems still far to be completely solved. One critical issue is, for instance, the poor reliability of WSNs, when some types of in-channel radio interference occur. This situation is typical for WSNs which transmit on radio channels shared with other communication systems and hence crowded of radio disturbances, like for instance the 2.4 GHz industrial, scientific and medical (ISM) band (2.4 - 2.4835 GHz). In this band, several sources of interference can be encountered, including for example IEEE 802.15.4 [2], IEEE 802.15.1 (Blue-

tooth) [3], and IEEE 802.11b/g [4] compliant devices, microwave ovens, cordless phones, etc. Moreover, severe timing constraints may be often required, meaning that relevant delays and/or uncertainties in data transfer might not be tolerated.

The most typical effect of interference is signal degradation, which occurs at the WSN receivers when the incoming useful signal is affected by interference. In order to reduce such effect, some communication standards employ sensing mechanisms like the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol. This mechanism enables multiple users to share the same physical medium, avoiding collisions and consequently signal degradation. A CSMA/CA compliant network node typically senses the in-channel power before transmitting. If the measured power is below a given threshold, meaning that the medium is actually not busy, then the transmission can begin. Otherwise, after a random backoff period, another attempt is performed. Common examples of wireless systems using CSMA/CA are IEEE 802.11 WLANs and IEEE 802.15.4 based WSNs. However, further effects of interference may anyway occur. In fact, if the interference at the WSN receivers is higher than a prefixed threshold, the channel is assumed as busy, and any node wishing to transmit is forced to wait and delay the delivery of data packets [1, 5].

In order to properly deal with these interference issues, a special effort is required just at the early design-stage. Helpful information can be found either theoretically, through the use of suitable simulation models, or experimentally, through measurements. Useful application notes can be also found in the scientific literature. For instance, in [6] the behavior of the IEEE 802.11 physical layer in an industrial environment is deeply investigated; in [7] a saturation analysis for IEEE 802.11 networks is reported; more generally, papers [1], [8] consider the usage of wireless technologies for industrial applications. Finally, papers [9], [10] are concerned with the use of IEEE 802.15.4. No analysis comparing the effects of different CSMA/CA operating modes against interference is instead available.

In this paper, the performance of a CSMA/CA-based WSN for industrial monitoring in the presence of real-life interference is investigated through a set of experimental tests. In particular, the tests have been performed on a testbed enlisting a real life IEEE 802.15.4 WSN for which a specific industrial monitoring protocol is used [5], capable both of cyclic polling and acyclic alarm management. Different kind of interfering networks have been deployed, comprising IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth) and IEEE 802.15.4 (ZigBee) two-node networks. The outline of the paper is as follows. Section 2 provides a brief overview of the IEEE 802.15.4 standard and of interfering sources. In Section 3, the key features of the monitoring protocol used for tests are presented. Section 4 describes the setup of the testbed and the measurement process. Section 5.1 and 5.2 describe the outcomes of an extensive experimental session carried out for the cyclic polling task, whereas, Section 5.3 does the same for the acyclic alarm task. Finally, Section 6 highlights some meaningful obtained results.

2 Standards Overview

2.1 IEEE 802.15.4 standard

IEEE 802.15.4 is a standard defined for Low Rate Wireless Personal Area Networks, characterized by low cost of the components, reduced coverage area, low power transmission, bit rate and energy consumption [2]. Such characteristics make the standard very suitable for WSN applications.

Similarly to the other networks standardized by the IEEE 802 committee, IEEE 802.15.4 defines only the first two layers of the open system interconnection stack, namely: Physical Layer (PHY) and Medium Access Control (MAC) [2]. In the PHY layer, the following functions are carried out: (i) in-channel power energy detection, (ii) link quality indication, (iii) communication channel selection, (iv) Clear Channel Assessment (CCA), and (v) packet transmission and reception through the radio channel. The standard also defines three bands: 868 - 868.6 MHz, 902 - 928 MHz, and 2400 - 2483.5 MHz, in which 26 separate channels are available. In particular, the latter (ISM band) is the most commonly used band and includes 16 channels, each of 3 MHz bandwidth, uniformly spaced as shown in Fig. 1(a).

The MAC layer is responsible for regulating channel access. In particular, two different types of operating schemes are available. The first one, called *beacon mode*, is based on CSMA/CA with a superimposed time-slot allocation, which schedules the network access. The second one, called *non-beacon mode*, is based on a pure CSMA/CA protocol. As it will be clarified in the following, due to the characteristics of the sensor nodes deployed in experiments, attention will be hereinafter exclusively focused on the *non-beacon mode* scheme.

In *non-beacon mode*, each station wishing to transmit:

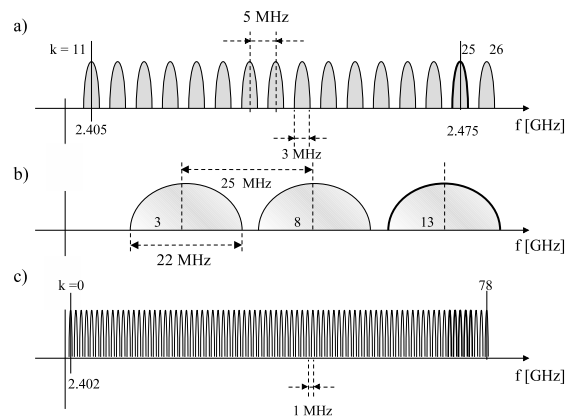


Figure 1. Frequency channels of (a) IEEE 802.15.4, (b) IEEE 802.11 and (c) IEEE 802.15.1 inside the ISM band.

- waits for a random *backoff* period uniformly distributed in the interval $(0, 2^{BE} - 1) \cdot BP$, where BE is a term called Backoff Exponent and BP is a basic period;
- senses the status of the channel (free or busy) through a Clear Channel Assessment (CCA) procedure;
- if the channel is free, transmits;
- else, if the channel is busy, then increments BE by one, up to a maximum BE level;
- if the retransmission number (NB) overpasses a pre-assigned maximum threshold, then signals failure, else restarts from the beginning.

The CCA procedure can be carried out in three different modes:

1. *CCA mode 1*: medium is assumed busy if the measured power level (Received Signal Strength Indication, RSSI) is higher than a prefixed threshold, CCA_{TH} .
2. *CCA mode 2*: medium is assumed busy if at least one signal with the modulation and spreading characteristics of IEEE 802.15.4 is detected.
3. *CCA mode 3*: medium is assumed busy if both the above assumptions are detected.

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

2.2 IEEE 802.15.1 standard

Another standard deployed for wireless personal area networks (WPANs) is IEEE 802.15.1 [3], in the following simply referred with the commercial term Bluetooth. This standard defines the first two layers, PHY and MAC of the protocol stack. Bluetooth systems operate in the 2.4 GHz ISM band. In a majority of countries, the range of this frequency band is 2400 MHz - 2483.5 MHz. As depicted in Fig. 1(c), the frequency spectrum is divided into 79 channels, spaced 1 MHz apart and characterized by carrier frequencies f_k , with $f_k = 2402 + k$ MHz and with $k = 0, \dots, 78$. The most interesting feature of Bluetooth systems is the transmission technique. In fact Bluetooth employs a technique called Frequency Hopping Spread Spectrum (FHSS) whereby the carrier frequency of the transmitter changes up to 1600 times per second.

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

2.3 IEEE 802.11 standard

The IEEE 802.11 [4] is a wireless local area network (WLAN) standard defining a total of 14 frequency channels, each of which characterized by 22 MHz bandwidth. As sketched in Fig. 1(b), these channels are partially overlapped, and only three channels at a time, *e.g.* 3, 8, and 13, can be used without mutual interference.

In a typical communication, an AP transmits periodically a frame called *beacon*, which contains information like a network identifier (ID), the beacon and channel parameters, and other traffic information. Each network station receives the beacon, and if it is intentioned to access the network, it sends a request for authentication. Once authenticated, a station may communicate to the AP or *vice versa* according to a CSMA/CA protocol. Usually, to assess the channel status, the Clear Channel Assessment (CCA) technique in *mode 1* is implemented [4]. In this mode, the channel is assumed idle if the channel power level is below a given user selectable threshold, called CCA threshold, otherwise it is assumed busy.

3 Industrial Monitoring protocol

An high layer protocol, based on a master-slave relationship, has been designed and implemented on a wireless sensor network for industrial monitoring. The protocol performs two classic tasks: a periodical polling of each slave for receiving data (*e.g.* readings of temperature, luminosity, humidity, pressure, rotation angle, etc.) from monitoring sensors (cyclic task), and an asynchronous alarm transmission, able to handle critical events from peripheral sensors (acyclic task). Without loss of generality, throughout the paper only one master and one slave are

used: for a more general analysis of the proposed solution the reader is referred to [5].

3.1 Cyclic task

The cyclic task consists in a round robin polling of each slave. In the case of a single slave, the master node queries the slave every *polling time*, T_p maintaining a sort of time-slot division. The master samples a virtual sensor that is physically mounted on the slave: this task is basically an abstraction of the physical sampling. A key feature of a sampling process is to assure an almost fixed sampling period (T_p), especially if the industrial monitoring application is involved in a control chain. In Fig. 2, a schematic representation of a master/slave communication is shown (n indicates the sequence number of the corresponding cycle).

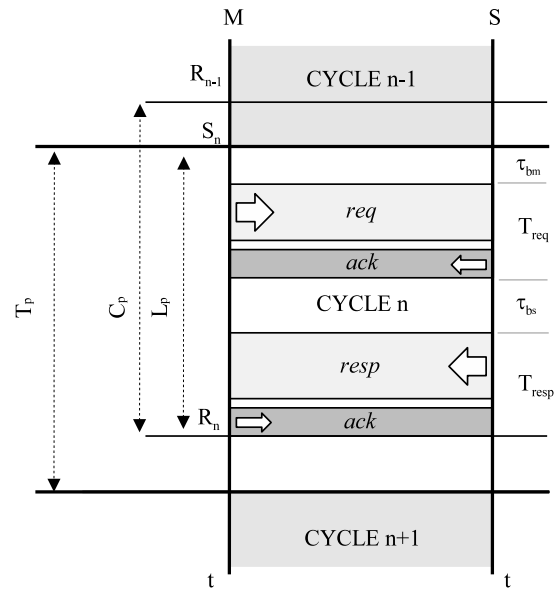


Figure 2. Packets exchange between master and slave inside a polling time.

Once gained the channel, the master begins querying the slave with a *request* packet (containing information like the destination node address, the sequence number of the performed cycle and time stamps). The slave, after a fixed delay defined by the IEEE 802.15.4 standard (12 symbol periods), replies with an ACK, and then sends a message containing the process data, after a second carrier sensing. In the case of correct reception, the master issues an ACK and waits for the expiration of the polling time before restarting the next cycle. If the handshake is not performed correctly (*i.e.* either the master or the slave does not receive the ACK frame), then the polling of that slave is considered failed and no retransmission is attempted. Within a cycle, the entire polling process is consists of the following steps.

At the instant S_n , the master starts the *request*: a first variable period, τ_{b_m} , is used to gain the channel and the transmission of the request packet plus the ACK takes another fixed period, T_{req} . After that, the slave performs several tasks and, once obtained the channel (after a variable τ_{b_s} period), sends the *response* to the master (the transmission of the response plus the ACK takes a fixed period, T_{resp}). Hence, τ_{b_m} and τ_{b_s} are the *initial backoff* period needed to access the channel by the master and the slave, according to the CSMA/CA protocol.

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

$$R_n = S_n + \tau_{b_m} + T_{req} + \tau_{b_s} + T_{resp}. \quad (1)$$

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

$$L_p = R_n - S_n = \underbrace{\tau_{b_m} + \tau_{b_s}}_{random} + \underbrace{T_{req} + T_{resp}}_{deterministic}. \quad (2)$$

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

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

$$C_p = R_n - R_{n-1} = \underbrace{S_n - S_{n-1}}_{deterministic} + \underbrace{(L_{p(n)} - L_{p(n-1)})}_{random}. \quad (3)$$

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

As shown in equation (3), C_p accounts for two different contributions: the first one, $S_n - S_{n-1}$ is deterministic since it represents the time elapsed between two subsequent transmissions of the request frame from the master to the same slave. The second contribution, $L_{p(n)} - L_{p(n-1)}$, is a random variable, since it is given by

the difference between L_p values evaluated for cycles n and $n - 1$.

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

3.2 Acyclic task

The acyclic task allows for the direct transmission of alarms from slave to master. When an alarm occurs at the slave side, it is put in a specific queue and the acyclic task is responsible of its transmission. Several techniques could be used by the acyclic task to access the physical medium in order to transmit alarms [8]. The proposed experiments rely on the *immediate policy* described in [8]. In practice, when an alarm has to be sent, the slave, irrespective of the activity carried out of the network, tries to gain access to the channel and, if successful, transmits the alarm. This is possible thanks to IEEE 802.15.4 CSMA/CA technique which makes each device able to autonomously access the network. Also this transmission is acknowledged by an ACK frame as in the case of polling task. The alarm can not be lost, so the transmission is retried until success. This queued policy does not allow to predict any upper bound to the *alarm latency*, L_a , that is the time that the slave takes to successfully deliver the alarm to the master.

4 Measurement System and Setup

The experimental session has been carried out by using the testbed sketched in Fig. 3, which comprises a two-nodes IEEE 802.15.4 network connected to a personal computer (PC) and a two-node interfering network. For the sake of simplicity, a single link network has been chosen since the monitoring protocol induces a time scheduling that gives similar results regardless of the number of slaves, as reported in [5]. The experiments have been performed in a non-anechoic room in order to emulate a real life environment. Preliminary measurements made with a Agilent E4407B spectrum analyzer inside the used band have been performed to assure the absence of external uncontrolled interferences.

The WSN enlists t-mote sky wireless sensors (motes) available from Moteiv [11, 12], based on the IEEE 802.15.4 communication radio system [13].

They are equipped with an universal serial bus (USB) port for programming, a 12-bits analog-to-digital converter, and a light sensor. A software named *WSN driver* (available free online [14]), purposely developed in Boomerang TinyOS IDE and Java, is used to implement the high layer protocol in non-beacon mode [15, 16]. The channel used is number 25 (centered at 2.475 GHz).

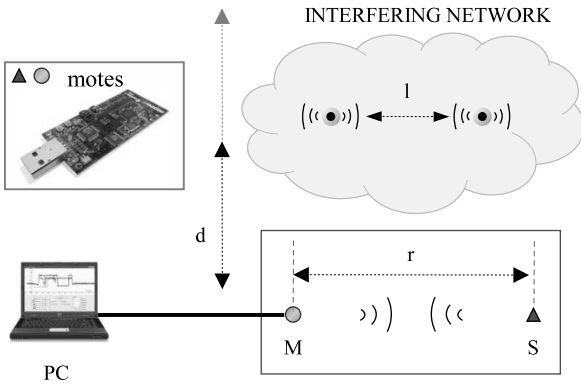


Figure 3. Testbed architecture.

This channel has been chosen suitably far away from external interference due to near operating WiFi Access Points. The master, M, and the slave, S, are positioned at a height of 1 m from the ground floor, with a distance $r = 1$ m from each other, assuring at least a -50 dBm received signal power, *i.e.* far above the receiver's sensitivity. The PC is used both to setup the network parameters at the beginning of the test, and to collect all the data coming from M via the USB port during the test. The PC is also equipped with a purposely developed software interface providing a user-friendly way to control the network and to perform the monitoring-process [14]. The slave's clock is also synchronized to the master's one at each polling cycle, so that the polling and alarm latencies can be correctly measured.

Each experimental session consists of 10000 cycles, with a polling time $T_p = 30$ ms. This value guarantees a null packet loss in absence of alarms and interference [5]. Alarms are software generated by a Poisson process with mean alarm inter-arrival time of 1000 ms (*i.e.* a stressful condition of nearly one alarm/second). All the collected data are post processed with Matlab.

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

- **CCA 1:** the channel is sensed busy if an arbitrary interfering source causes: $RSSI \geq CCA_{TH}$, with $CCA_{TH} = -77$ dBm (default value).
- **CCA 2:** the channel busy only if another IEEE 802.15.4 transmission is revealed (Carrier detection).
- **NO CCA:** motes do not perform CCA, and the back-off basic period (BP defined in section 2.1) is set to zero, switching the backoff delay off.

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

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

5 Experimental Results

5.1 Packet Error Rate

A first set of results from the experiments have been obtained in terms of Packet Error Rate (PER). For every interference configuration described in section 4, three tests have been performed varying the CCA mode. The results are summarized in Fig. 4.

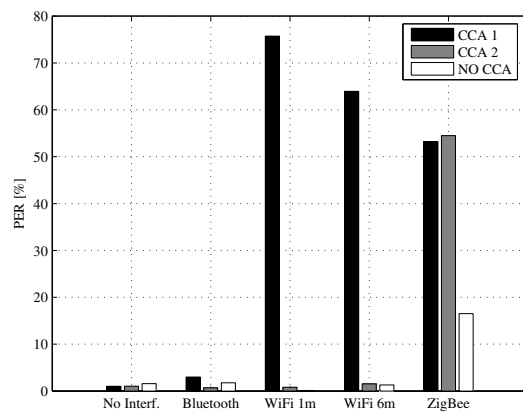


Figure 4. Measured PER for all the considered configurations.

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

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

5.2 Polling Cycle Time

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

As stated in section 3.1, the expected pdf in both CCA 1 and 2 modes is gaussian-like (cubic splines), due to the convolution of four equal distributed uniform random variables (due to the initial back-offs). The curve is centered at 30 ms, with a minimum value of 10 ms and a maximum value of 50 ms. In Fig. 5 the polling cycle pdf in the case of no interference is shown. As predicted, for CCA mode 1 and 2 the back-offs of CSMA/CA protocol lead to an identical shape of the two curves. With CCA disabled, the presence of an impulsive pdf centered exactly at 30 ms (the chosen T_p) means an almost constant and deterministic value of C_p .

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

Although not reported, even with a Bluetooth interferer the pdfs remain the same of Fig. 5. This interferer only increases the PER without modifying substantially the polling latency.

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

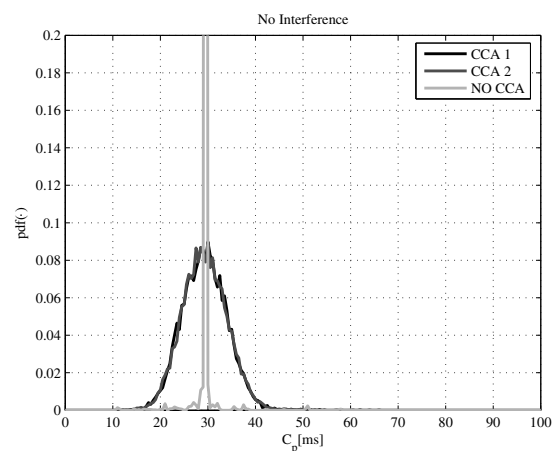


Figure 5. Probability density function for Polling cycle (C_p) with no interference.

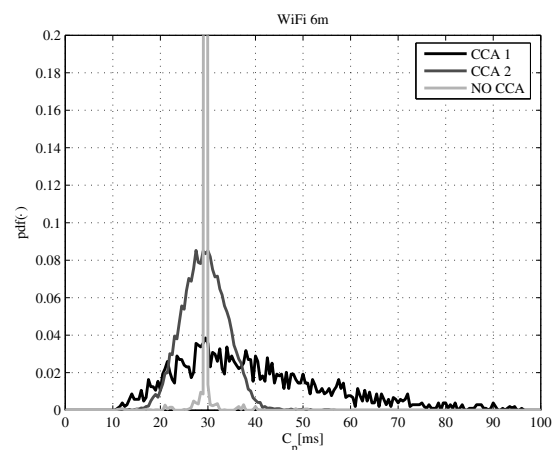


Figure 6. Probability density function for Polling cycle (C_p) with WiFi interference.

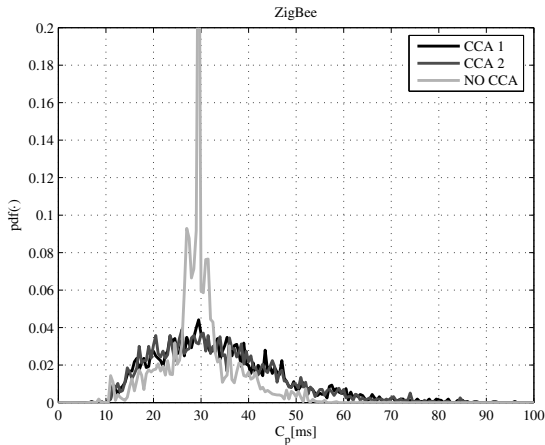


Figure 7. Probability density function for Polling cycle (C_p) with ZigBee interference.

Conversely, in the case of CCA 1 the channel is often sensed busy and the CSMA/CA protocol tries to wait to by-pass the interference. This pdf reveals that many attempts to gain the channel have been tried, causing a great dispersion of the pdf that now spans from 10 to 95 ms and is much flatter than the one obtained in the absence of interference.

The results summarized in Fig.7 reveal that in this case also with NO CCA the polling cycle is no longer constant, probably due to the large amount of failed pollings. As expected, both modes 1 and 2 lead to the same results, because with a IEEE 802.15.4 interferer above the CCA threshold the channel is sensed busy by both methods. With a ZigBee interferer the choice of NO CCA is no longer a efficient solution, even if gives a better behavior.

5.3 Alarm Latency

In this section the alarm latency pdf analysis is presented, highlighting the CCA mode influence on it. The reported plots are normalized histogram, typically made of 300 samples, *i.e.* the number of alarms generated, in mean, in a total of 10000 polling cycles of 30 ms. The statistical behavior suggested in the following figures, even if based on a small amount of data, remains the same even with a larger and more complete set of samples, as reported in [5]. In Fig. 8 the alarm latency pdf in the absence of interference is shown. For CCA mode 1 and 2 the shape is asymmetrical, exponential like: latency is upper bounded to 40 ms. For NO CCA latency is smaller, more concentrated below 10 ms with a spike at 3 ms. As for polling cycle, Bluetooth does not influence alarm latency, leading to pdfs similar to those just described.

The effect of WiFi interference (Fig. 9), as for polling cycle, is greatly mitigated by the use of CCA 2, or, even better, with NO CCA. If mode 1 is used, alarm latency is spread toward greater values, even above 100 ms.

The effect of ZigBee interferer is deleterious also on alarm latency. Both with CCA 1 and 2 the latency is enlarged toward 120 ms, and also with a NO CCA mode, although more concentrated below 40 ms, the latency still reaches 120 ms.

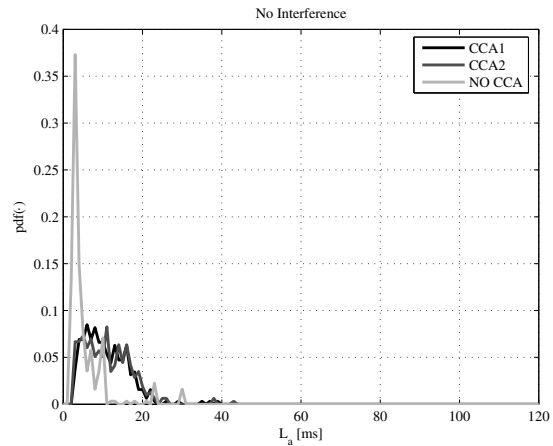


Figure 8. Probability density function for Alarm latency (L_a) with no interference.

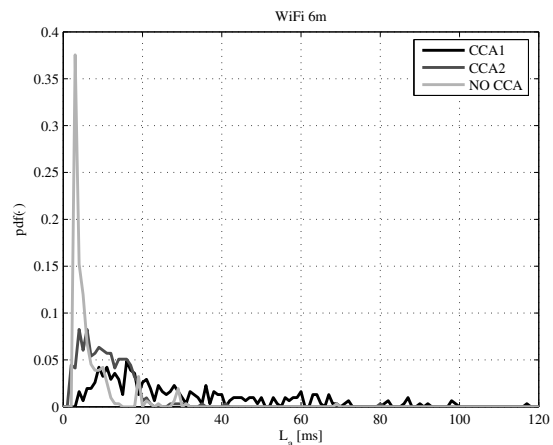


Figure 9. Probability density function for Alarm latency (L_a) with WiFi interference.

6 Conclusions

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

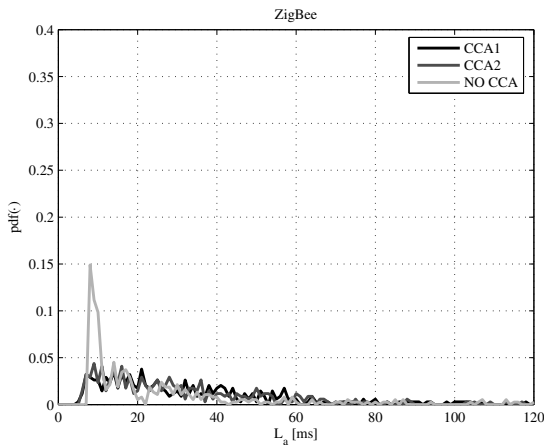


Figure 10. Probability density function for Alarm latency (L_a) with ZigBee interference.

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

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

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

References

- [1] A. Willig, K. Matheus, A. Wolisz: “Wireless Technology in Industrial Networks”, *Proceedings of the IEEE*, Vol. 93, June 2005.
- [2] “Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs),” *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–305, 2006.
- [3] “Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs),” *IEEE Std 802.15.1-2005 (Revision of IEEE Std 802.15.1-2002)*, pp. 1–580, 2005.
- [4] “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, pp. C1–1184, June 12 2007.
- [5] M. Bertocco, G. Gamba, A. Sona and S. Vitturi: “Performance Measurements of CSMA/CA-Based Wireless Sensor Networks for Industrial Applications”, *Proc. of IMTC 2007*, May 2007, Warsaw Poland.
- [6] A. Willig, M. Kubisch, C. Hoene and A. Wolisz: “Measurements of a wireless link in an industrial environment using an IEEE 802.11-compliant physical layer”, *IEEE Trans. on Ind. Electr.*, Vol. 49, N. 6, December 2002.
- [7] G. Bianchi: “Performance analysis of the IEEE 802.11 Distributed Coordination Function”, *IEEE Journal on Sel. Areas in Comm.*, Vol. 18, N. 3, 2000.
- [8] F. De Pellegrini, D. Miorandi, S. Vitturi, A. Zanella: “On the use of wireless networks at low level of factory automation systems”, *IEEE Transactions on Industrial Informatics*, Vol. 2, Issue 2, pp. 129-143, May 2006.
- [9] A. Flammini, D. Marioli, E. Sisinni, A. Taroni and M. Pezzotti: “A wireless thermocouples network for temperature control in plastic machinery”, *Proc. of WFCS 2006*, pp. 219-222, June 2006, Torino Italy.
- [10] A. Koubaa, M. Alves and E. Tovar: “A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks”, *Proc. of WFCS 2006*, pp. 183-192, June 2006, Torino Italy.
- [11] “Moteiv Tmote Sky Data Sheet”, *Moteiv*, 2006.
- [12] “MSP430x1xx Family Data Sheet”, *Texas Instruments*, 2006.
- [13] “CC2420 Data Sheet”, *Texas Instruments*, 2006.
- [14] [Online]. Available: www.dei.unipd.it/ricerca/gmee.
- [15] D. Gay, P. Levis, D. Culler: “Software Design Patterns for TinyOS”, www.tinyOS.net.
- [16] J. Polastre, J. Hui, P. Levis: “A Unifying link abstraction for wireless sensor networks”, *SenSys '05* (site www.polastre.com).