Comparative assessments of IEEE 802.15.4/ZigBee and 6LoWPAN for low-power industrial WSNs in realistic scenarios

Emanuele Toscano, Lucia Lo Bello
Department of Electrical, Electronic, and Computer Engineering
University of Catania
Viale A. Doria 6 - 95125, Catania (Italy)
emanuele.toscano@dieei.unict.it, lucia.lobello@dieei.unict.it

Abstract

This paper addresses the low-power mechanisms provided by the IEEE 802.15.4/ZigBee and 6LoWPAN protocols, providing comparative assessments through experimental measurements performed on a real testbed. For a fair performance comparison, a special effort has been made to both tune the parameters of each protocol so as to make it able to properly operate in low-power mode and make the measurement scenarios equivalent in terms of traffic and energy efficiency. After addressing this tuning phase, the paper compares the protocols performance obtained on the same network, under the same workload and while working with the same duty cycle. The comparison focuses on the impact of the low-power mechanisms on the network performance. The experimental assessments highlight the strengths and weaknesses of both protocols when working in low-power mode.

1. Introduction

The IEEE 802.15.4/ZigBee and 6LoWPAN protocols are widely used for low-power Wireless Sensor Networks and the interest towards their adoption in factory automation for monitoring tasks such as machine assessment for failure prevention or plant energy consumption metering is steadily increasing. In this context, this paper aims to compare the performance of the low-power operating modes of IEEE 802.15.4/ZigBee and 6LoWPAN for industrial WSNs. To this aim, we first describe the two protocols with a focus on the low-power mechanisms they implement and on their capabilities. Then, we compare the protocols through experimental measurements on a real testbed, focusing on the impact of the low-power mechanisms on the network performance. Since these low-power mechanisms work by putting nodes to sleep so as to decrease their duty cycle, the protocol comparison is performed making them work on the same HW/SW platform, under the same workload and using the same duty cycle. To achieve the latter condition, a preliminary tuning of the operating parameters of the two protocols is needed. Such a tuning is not straightforward, as the low-power modes implemented by the two protocols, and thus the relevant parameters, are significantly different. In fact, although both protocols are based on the IEEE 802.15.4 PHY and MAC, they follow different approaches for energy saving and multihop communication. On one hand, the IEEE 802.15.4/ZigBee protocol defines the beacon-enabled mode in a way that allows nodes to stay in low-power sleep states for most of the time, while multihop transmission with sleeping nodes is achieved through the use of the tree routing protocol through the cluster-tree hierarchy [1]. The 6LoWPAN protocol, on the other hand, is based on IPv6 and operates in a fully asynchronous way. It adopts a mesh topology and uses a routing algorithm which does not take care of sleeping nodes, thus requiring approaches such as low-power listening [2] for energy saving purposes. In the paper, we address these differences and present a methodology to correctly perform the parameter tuning for each protocol. Then, we compare the experimental results obtained by the two protocols in two realistic scenarios under the same operating conditions.

The paper is organized as follows. Section 2 outlines related work while Section 3 describes the two protocols with a focus on their low-power capabilities. Section 4 discusses the testbed setup and the methodology for tuning the network configuration parameters, while Section 5 shows the experimental results, with a special focus on the trade-off between energy efficiency and network performance. Finally, Section 6 provides our conclusions.

2. Related work

Several performance studies of the IEEE 802.15.4/ZigBee protocol exist in the literature, carried out by experimental measurements, simulations and analytical models. A detailed analytical evaluation of slotted IEEE 802.15.4 performance in a star topology network is provided in [3], which considers both saturated and unsaturated periodic traffic scenarios. Both an analytical model and a simulative analysis are used in [4] to assess the performance of low-power IEEE 802.15.4/ZigBee networks. An experimental analysis of star and tree ZigBee network based on commercially available hardware and software is provided in [5] with the aim of determining the reach and limitations of the technology. All these works, however, only provide some parameter studies, without any comparison with other protocols. This paper, on the contrary, aims at comparing the performance of low-power operating modes

in IEEE 802.15.4/ZigBee and 6LoWPAN.

The 6LoWPAN protocol is attracting the attention of the market and research thanks to the new possibilities offered by the integration of WSN and Internet technologies. In [6], the main choices made in the design of the 6LoWPAN protocol are discussed, together with the open problems and challenges for further development. In [7] a method to use 6LoWPAN within IPv4 networks is provided, so that also IPv4 can take advantage of WSN integration. The adaptation of the SPEED geographic routing algorithm in a 6LoWPAN scenario is presented and its experimental validation in a real testbed is discussed in [8] so as to enable soft real-time transmissions over 6LoWPAN. In [9] a comprehensive description of the 6LoWPAN communication approach is given, and a protocol evaluation based on experimental measurements is also provided. However, no comparison with other WSN technologies is provided. A comparison of the capabilities provided by some competing technologies for home automation is provided in [10]. Both ZigBee and 6LoWPAN are taken into account in whis work that, however, focuses on home automation and only provides a qualitative comparison, without any quantitative evaluation of the protocols performance. An experimental comparison between low-power protocols for WSN is provided in [11], where the 6LoWPAN protocol is compared with the Collection Tree Protocol and the Multihop Link Quality Indicator Protocol. However, the ZigBee protocol, which is actually the most widely used WSN protocol at the moment, is not taken into account. Moreover, the different tradeoff between energy efficiency and performance for the different protocols is not analyzed. This paper, on the other hand, aims at analyzing such tradeoff by studying the performance of the IEEE 802.15.4/ZigBee and 6LoWPAN protocols when low-power configurations are used. Here energy-efficiency is measured by means of nodes duty cycle. In fact, in sensor nodes the power consumptions of transmission, reception, and idle states are similar,while the low-power sleep state is some order of magnitude smaller. It is easy to derive an approximated estimation of the nodes' energy consumption from their duty cycle, but this is behind the scope of our paper, since it is widely dependent on the hardware platform.

3. The protocols under test

This section provides a brief overview of the two protocols compared in this paper, focusing on the physical and medium access, and on their different energy-saving techniques.

3.1. IEEE 802.15.4/ZigBee

The IEEE 802.15.4 standard [12] defines the physical and MAC layer of low-power Wireless Personal Area Network (WPAN). This is a protocol for low-rate and low-power communications, particularly suitable for small embedded devices such as sensor nodes. The protocol allows for varying nodes' duty cycles from 100% to a minimum of about 0.1%.

The IEEE 802.15.4 MAC protocol features two operating modes: a nonbeacon-enabled mode, in which nodes access the channel using a classical (unslotted) CSMA/CA mechanism and a beacon-enabled mode in which a slotted CSMA/CA mechanism is used that supports duty cycling to save nodes energy. This mode is based on a superframe structure, which is delimited by beacon frames, that are sent by a coordinator in order to synchronize the associated nodes.

The superframe structure, which is shown in Fig. 1, is divided into two parts: an active and inactive part. The

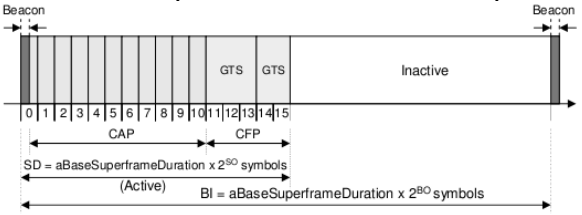

Figure 1. IEEE 802.15.4 Superframe.

active part is divided into 16 slots of equal duration, at the boundary of which nodes can access the channel to transmit their data. This is divided in turn into two subparts: the Contention Access Period (CAP), in which medium access is regulated by the slotted CSMA / CA, and the Contention Free Period (CFP), in which up to seven Guaranteed Time Slots (GTS) can be allocated to access the channel without any contention. In the inactive part, on the other hand, there is no communication and therefore nodes can turn their radios off and go into power saving mode (sleep). The timing of the superframe is governed by two parameters:

- The *Beacon Order (BO)*, which defines the time interval between two consecutive beacons, and consequently the superframe duration. This interval is called Beacon Interval (BI) and is defined as $BI = aBaseSuperframeDuration \cdot 2^{BO}$, where $aBaseSuperframeDuration$ is a constant defined by the standard and is equal to 960 symbols, with a symbol time of 16 microseconds.
- The *Superframe Order (SO)*, which defines the duration of the active period of the superframe. This time interval is called superframe duration (SD) and is defined as $SD = aBaseSuperframeDuration \cdot 2^{SO}$.

According to the IEEE 802.15.4 standard, we have $0 \leq SO \leq BO \leq 14$.

As nodes sleep in the inactive period of the superframe, the duty cycle (DC) of nodes only depends on the superframe structure, and can be specified as

$$DC = \frac{SD}{BI} = 2^{(IO)}, \quad (1)$$

where $IO = SO - BO$ is called the Inactivity Order.

When IEEE 802.15.4 devices use their low-power capabilities in the beacon-enabled mode, two different data transfer modes are used to achieve communication between a coordinator and its devices. In the case a packet has to be sent from the device to the coordinator, *direct transfer* is used. In this mode, the device waits for the beacon frame sent by the coordinator to synchronize with the superframe. Then, it sends the data in one of the superframe slots. Upon the reception of the packet, the coordinator sends an acknowledgment to complete the transmission. This last point, however, is optional. In the case a packet goes from the coordinator to one of its associated devices, the *indirect*

mode is used. Here, the coordinator indicates in the beacon frames that it has pending data for the device. Upon the reception of the beacon, the device sends a data request frame. After receiving the data request, the coordinator sends both an acknowledgment for the data request frame and the data packet. Finally, the network device sends an acknowledgment to complete the transmission.

Since IEEE 802.15.4 defines only the PHY and MAC layers, the data forwarding protocol is defined by the ZigBee specification [1], as well as the higher layers of the protocol stack. The ZigBee protocol supports three different topologies, i.e., star, mesh, and cluster-tree. In the cluster-tree topology, the network is organized in clusters, each managed by a coordinator, while coordinators are hierarchically connected to form a tree, rooted at the PAN coordinator. Two different forwarding protocols are defined by the ZigBee specification: an ad-hoc routing protocol based on Ad hoc On-Demand Distance Vector (AODV) [13] and a tree routing protocol that forwards the packets following the coordinators' hierarchy. Star and cluster-tree topologies that use tree routing can take advantage of the low-power feature of the beacon-enabled IEEE 802.15.4 mode. However, the star topology can only be used for small networks, whereas the cluster-tree topology is suitable for larger sensor networks. For this reason, this paper focuses on cluster-tree topologies with tree routing.

ZigBee networks using the beacon-enabled mode allow for deterministic nodes duty cycles, since these depends only on the superframe parameters, according to eq. (1). These parameters can be set at design time so as to meet the network requirements in terms of energy consumption. However, the waiting time for the next active superframe and the increased complexity and overhead of indirect communication can affect the performance of time-sensitive data transmissions, especially in the case of multihop cluster-tree networks.

3.2. 6LoWPAN

The 6LoWPAN protocol specification [14] enables IPv6 communications over low-power wireless devices. In particular, the 6LoWPAN RFC standard defines the network layer of IP-based WPANs which use the IEEE 802.15.4 physical and medium access layers. The main reason for developing 6LoWPAN was that the common IPv6 standard is too bulky for small embedded devices such as sensor nodes. The standard IP protocol could not fit WSNs for several reasons. For example, the standard IPv6 header is 40 octets, which is comparable to (or even bigger than) the sensor data that is transmitted over a WSN. An additional overhead of 20 or 8 octets is introduced by the TCP or UDP protocol, respectively. Considering that the maximum frame size at the MAC layer is 102 octets, or even smaller in the case link-layer security is enabled, very few bytes would be available for data transmissions. Moreover, the IPv6 maximum transfer unit is 1280 octets, which is much bigger than the maximum IEEE 802.15.4 frame. Therefore some fragmentation is needed. Other problems raised by the support of IP over IEEE 802.15.4 MAC and PHY are the addressing schema and the routing mechanism. In fact, IP and IEEE 802.15.4 have different addressing formats and the routing protocols of IP-based networks are not able to deal with sleeping nodes. Therefore, 6LoWPAN introduces an adaptation layer between the MAC and the network layer, which compresses the header to a few bytes while keeping the main IPv6 functionalities. The 6LoWPAN adaptation layer guarantees interoperability with legacy IP networks, but introduces mechanisms to support fragmentation and reassembly, to compress the headers, and to perform address mapping between IPv6 and IEEE 802.15.4 addresses. The details about these mechanisms, however, are outside the scope of this paper.

The 6LoWPAN specification does not provide any specific mechanism to achieve low-power communication, but it relies on lower layer approaches. The simplest solution to achieve energy efficiency in 6LoWPAN networks might be the exploitation of low-power capabilities of the beacon-enabled IEEE 802.15.4 protocol. However, even though the MAC layer could theoretically work in both beacon-enabled and nonbeacon-enabled mode, all the current 6LoWPAN implementations only support the nonbeacon-enabled mode. Here, beacons are used for link-layer device discovery to facilitate device discovery and association.

The main reason for not implementing the beacon-enabled communication in 6LoWPAN is that IP protocols assume that the link is always-on. This is for the sake of simplicity, so that IP protocols do not need to schedule datagram transmissions [9]. Therefore, 6LoWPAN must use duty-cycling techniques which give the illusion to the upper layers that the receiver is always on, although it is actually off most of the time. Most of the 6LoWPAN implementations use Low Power Listening (LPL) [2], a technique which allows the nodes to access the channel in a completely distributed and asynchronous way.

The operation behind LPL is the following. An idle node periodically wakes up, turns the radio on and checks for activity. If no activity is sensed, the node turns off the receiver and goes back to sleep, otherwise, the node stays awake until the packet is received or, in the case of a false positive, until a timer expires. A graphical representation of LPL operations is shown in Figure 2. As the false positives decrease the energy-efficiency of the protocol, it is important to have an accurate channel assessment. Nonetheless, this technique has been recently adapted to work in noisy environments [15] as well.

In order to avoid packet losses while the node is sleeping and, therefore, to make the link appear as always-on, the

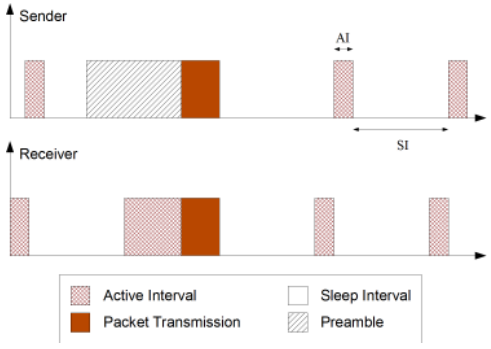

Figure 2. Low Power Listening.

preamble of each packet must last as long as the interval between two consecutive LPL samples.

In the end, although 6LoWPAN is based on IEEE 802.15.4 PHY/MAC, the actual mechanism to achieve low-power communication is quite different from that adopted in IEEE 802.15.4/ZigBee networks. While in ZigBee networks the nodes have deterministic duty cycles, but asynchronous communications may arrive in the inactive part of the superframe and therefore experience large delays, in 6LoWPAN networks the duty cycle of nodes depends on the amount of traffic, but the delay introduced by the low-power mechanism is almost constant.

4. Testbed setup

The aim of our analysis is to compare the performance of the IEEE 802.15.4/ZigBee and 6LoWPAN protocols when operating in low-power mode, i.e., with small duty cycles. For a fair comparison between the protocols, we use two open source implementations which run on same hardware platform and under the same operating system. We chose TelosB as the common hardware platform and TinyOS v. 2.1.1 as the common operating system. For the IEEE 802.15.4/ZigBee protocol, the TKN154 [16] MAC/PHY implementation was used. On top of that, a simple network layer featuring the tree routing protocol has been implemented. For the 6LoWPAN protocol, the BLIP [17] implementation included in TinyOS version 2.1.1 has been used.

The network realized using the two protocols in turn is made up of four TelosB nodes deployed in a linear topology, as shown in Figure 3. The sender and the receiver

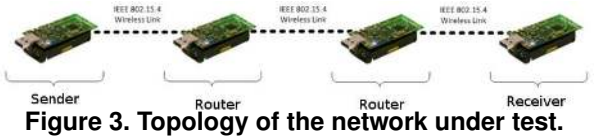

Figure 3. Topology of the network under test.

are connected to two USB ports of the same PC, so that it is possible to use a common time reference to calculate one-way end-to-end packet delays. For this purpose, every time either the sender produces a packet or the receiver receives a packet, a unique packet identifier is communicated to the PC, which adds a timestamp. At the end of each experiment, the packet identifier is used to find the reception time associated to every transmitted packet.

Since the two protocols have different characteristics, an effort is needed to make the measurement scenarios equivalent in terms of both traffic and energy efficiency. First of all, we have to carefully tune the parameters of each protocol so as to make it able to properly operate in low-power mode. For IEEE802.15.4/ZigBee, this means that any potential for overlapping transmissions (i.e. for overlapping superframes) has to be eliminated by a proper selection of the BO and SO parameters. For 6LoPAN, proper operation under low-power mode implies a suitable selection of the Sleep Interval. After performing this tuning phase for both protocols, we will be able to compare them on the same network, under the same workload while working with the same duty cycle.

The details of the network configuration parameters tuning are given in subsections 4.1 and 4.2 for the IEEE 802.15.4/ZigBee and the 6LoWPAN protocol, respectively.

4.1. Tuning the IEEE 802.15.4/ZigBee network

A preliminary assessment of the TKN154 implementation has been carried out to tune the IEEE 802.15.4 parameters used in our testbed. The reason is that cluster-tree topologies comprise multiple coordinators, each with its own superframe delimited by beacon frames. To achieve reliable communication, the active superframes of all the coordinators (and therefore their beacon transmissions) must be scheduled in a conflictless manner, i.e., without any overlapping between the active superframes of two coordinators. This means that the feasible values for the BO and SO parameters in a network depend on the number of coordinators in the network. Although recent research has shown that multichannel superframe scheduling [18] can be successfully applied to IEEE802.15.4 cluster-tree WSNs with a significant increase in the schedulability space, this paper sticks to the classical single-channel approach, in which all the coordinators use the same radio channel. In this case, in a cluster-tree network, the sum of all the duty cycles must be lower than one, i.e.,

$$\sum_i DC_i = \sum_i \frac{SD_i}{BI_i} \leq 1. \qquad (2)$$

To assess which BO and SO parameters can be used without collisions between overlapping suoerframes, we carried out some measurements in a simple network, comprising the PAN Coordinator and another coordinator, and analyzed the distribution of the experimental beacon intervals. This test has been repeated for BO values from 1 to 5. The value of 0 has not been tested, because it is not possible to have a beacon interval equal to the superframe duration in a cluster-tree network with multiple coordinators. Statistics about the actual beacon intervals are taken from 3000 observations for each BO value, and are presented both in a compact tabular form (in Table 1) and in terms of the probability density function (PDF) estimated from the experimental data (in Figure 4). In the

Table 1. Statistics about the experimental beacon intervals.

BO	mean BI	Std. dev.	min BI	max BI
1	29.3 ms	2.3 ms	15 ms	41 ms
2	58.6 ms	2.0 ms	35 ms	81 ms
3	117.2 ms	2.3 ms	90 ms	142 ms
4	234.4 ms	2.1 ms	222 ms	246 ms
5	468.7 ms	1.8 ms	456 ms	478 ms

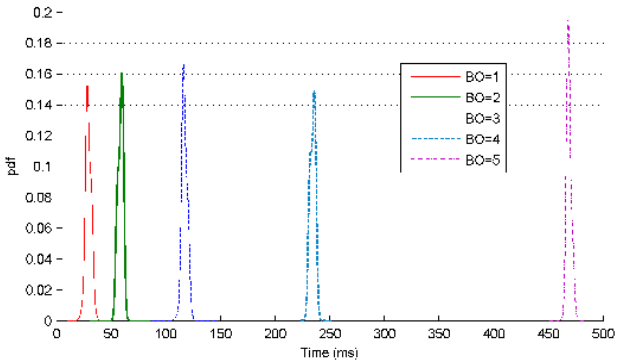

Figure 4. Probability density function of the actual beacon intervals.

figure we see that, apart from an obvious shift of the mean value, the PDF of the actual beacon intervals has almost the same shape, regardless of the BO value. In other words, our experimental results suggest that the actual beacon intervals follow the same distribution, albeit with some minor differences, such as a slightly narrower and taller PDF when using a BO value of 5. The same trend can be also seen in Table 1, where the standard deviation in correspondence to a BO of 5 is slightly smaller than the others. Moreover, looking at Figure 4, it is possible to see that the PDF width is always wider than $15 - 20ms$. This explains why this HW/SW platform could not support the smallest beacon order defined by the IEEE 802.15.4 standard, which provides a beacon interval of $15.36ms$. For this reason, in our tests we avoided the use of the smallest superframe order, which leads to a superframe duration of $15.36ms$. From the table we also note that the mean BI is always shorter than the theoretical value, given by $BI = aBaseSuperframeDuration \cdot 2^{BO} \cdot T_{symbol} = 960 \cdot 2^{BO} \cdot 16\mu s$. As explained in [19], this is due to hardware limitations of the TelosB platform, which provides an actual T_{symbol} of $15.259\mu s$ which is smaller than the theoretical $16\mu s$ value. However this behaviour does not compromise the validity of our results, since all our measurements are made using the same hardware platform.

In the following, the PDF of the actual beacon intervals will be used to calculate the probability of an overlap between the active parts of subsequent superframes sent by either two adjacent coordinators (e.g., one coordinator and its child in the cluster-tree topology) or by the same coordinator.

4.1.1. Probability of overlapping superframes. We recall that the IEEE 802.15.4 standard implements superframe scheduling by using the *StartTime* parameter in the MLME-START primitive, which specifies the time offset between the parent and the child superframes. Each superframe starts as soon as the beacon is sent by the coordinator, therefore if the beacon frame is sent at time t, the active part of the superframe finishes at time $t + SD$, and the superframe of the child coordinator starts at time $t + StartTime$. To achieve reliable communication, i.e., to avoid overlapping, $StartTime$ must be greater than SD. Note that, here, the time of the next beacon frame t repeats with a random interval, with the PDF shown in Figure 4. Therefore, also the actual time at which the active part of the superframe ends and the time at which the next superframe from an adjacent coordinator starts are random variables. We refer to these random variables as T_{SD} and $T_{StartTime}$, respectively. To obtain the probability that two active superframes (the one of a generic coordinator and that of the child coordinator) overlap, it is necessary to calculate the probability

$$\begin{aligned} P(\text{overlap}) = & \quad P(T_{StartTime} \leq T_{SD}) = \\ = & \quad P(T_{StartTime} - T_{SD} \leq 0). \end{aligned} \tag{3}$$

To obtain analytical results in a closed form, we approximate the beacon interval PDFs in Figure 4 to normal distributions $\mathcal{N}(\mu, \sigma^2)$, with mean μ and standard deviation σ^2 defined as the relevant entries in Table 1. In other words, we suppose that T_{BI} follows the distribution

$$f_{BI}(t) = \mathcal{N}(\mu_{BI}, \sigma^2_{BI}), \tag{4}$$

where the μ_{BI} and σBI parameters are those shown in Table 1. This is a good enough approximation, since in all our experiments the Mean Square Error between the experimental PDFs in Figure 4 and their normal approximation is about 10^{-4}.

If we call T_{BI} the random variable which describes the beacon interarrival time, the time at which the active part of the superframe ends can be modeled as $T_{BI} + SD$, i.e., the same PDF as the beacon intervals, but shifted by the superframe duration. Similarly, the time at which the superframe from the child coordinator starts is $T_{BI} + StartTime$. It follows that the random variables T_{SD} and $T_{StartTime}$ have the following PDFs:

$$f_{SD}(t) = \mathcal{N}(\mu_{BI} + SD, \sigma^2_{BI}), \tag{5}$$

$$f_{StartTime}(t) = \mathcal{N}(\mu_{BI} + StartTime, \sigma^2_{BI}). \tag{6}$$

In order to estimate the probability in eq. (3) it is possible to derive the PDF of the random variable $T_{diff} = T_{StartTime} - T_{SD}$, which results in

$$f_{diff}(t) = \mathcal{N}(StartTime - SD, 2\sigma^2_{BI}). \tag{7}$$

The probability in eq. (3) can be therefore obtained by considering the Cumulative Distribution Function (CDF) of T_{diff}, which is $F_{diff}(t) = \int_{-\infty}^{+\infty} f_{diff}(x)dx$. In particular, we obtain the probability by estimating the CDF at point $t = 0$. From the PDF $f_{diff}(t)$ in (7), we obtain

$$P(\text{overlap}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{SD - StartTime}{2\sigma_{BI}}\right)\right]. \tag{8}$$

The values of such probability are plotted in Figure 5 as a function of $\Delta T = StartTime - SD$. Here we can see that, if $StartTime$ is equal to SD, the overlapping probability is always 0.5, regardless of the beacon interval value. Then the probability decreases with a rate that grows as the standard deviation of the BI distribution decreases.

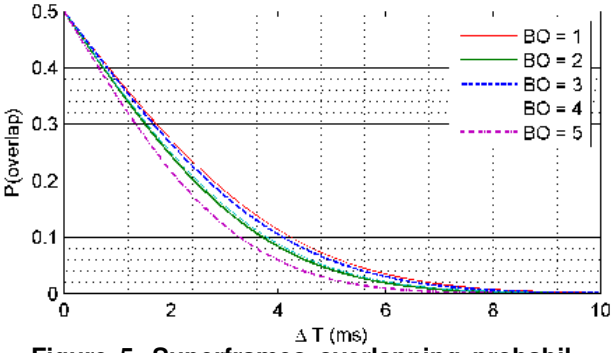

Figure 5. Superframes overlapping probability.

4.1.2. Probability of collapsing superframes. A less likely but detrimental overlapping may occur between two subsequent superframes sent from the same coordinator. This event happens when the the next superframe starts before the finishing time of the current superframe. We refer to this problem as superframe collapsing. The probability of such event can be written as

$$P(\text{collapse}) = P(T_{BI} < SD). \tag{9}$$

Approximating again the random beacon interval to a normal probability distribution, we can calculate such a probability as

$$P(\text{collapse}) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{SD - \mu_{BI}}{\sigma_{BI}\sqrt{2}}\right)\right]. \tag{10}$$

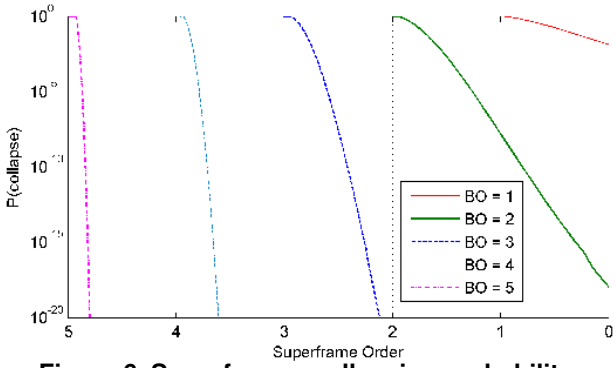

Figure 6. Superframes collapsing probability.

The results obtained from the experimental beacon intervals distributions given in Figure 4 are provided in Figure 6 as a function of the superframe order. It can be seen how the beacon order is related to the superframe order with respect to the collapsing probability. In particular, we see that with the considered platform $BO = 1$ is not a safe value, since the collapsing probability is non-negligible, even with the smallest superframe order. To stay on the safe side, a minimum beacon order of two should be used, which results in a collapsing probability of about 10^{-8} when the superframe order is 1. Increasing the beacon order further, we obtain smaller and smaller collapsing probabilities.

4.1.3. Concluding remarks on tuning the IEEE 802.15.4 network. In the four nodes topology shown in Figure 3, only the transmitter node is configured as an end-device, while the receiver and the two routers are configured as coordinators. Although a beacon order of 2 provides a small enough superframe collapsing probability, we decided not to use this BO in our network. The reason is that, due to eq. (2), to support three coordinators with a beacon order of 2, a superframe order of 0 is required. However, our experimental results in Sect.4.1 show that superframe durations as small as $15.36ms$ cannot be supported.

As a result, in our experiments we use a minimum beacon order of 3, with a minimum superframe order of 1. Moreover, since the beacon interval is always longer than the duration of three superframes (the beacon interval is at least four times the superframe duration), we used the spare time in the schedule to introduce a guard time ΔT between the end of a superframe and the beginning of the next one. In this way, we obtained a negligible superframe overlapping probability.

4.2. Tuning the 6LoWPAN network

The 6LoWPAN protocol does not require any synchronization between the network devices, therefore it is not subject to the superframe overlapping and superframe collapsing problems analyzed in Section 4.1. However, a different issue must be faced in the design of low-power WSNs using 6LoWPAN communication, that is, the configuration of sleep-related parameters so as to meet the requirements in terms of nodes and network lifetime. This issue is relevant to our analysis, since to have a fair comparison between low-power protocols, we must analyze the network performance with comparable energy savings features. For this reason, it is necessary to tune the LPL parameters so as to achieve a desired duty cycle for a node.

There are two LPL parameters which define the duty cycle of an idle node, the Sleep Interval (SI) and the Activity Interval (AI). As shown in Figure 2, an LPL cycle is made up of the AI followed by the SI. In the LPL implementation provided by TinyOS, the SI is determined by a timer and therefore is a configurable parameter, while the AI is implemented by consecutive CCA requests and therefore is not as easy to change as the SI. As soon as a packet is sensed, during the AI, the receiver is kept on so as to receive the whole packet. Therefore, all the traffic in the network affects the nodes duty cycle.

Considering a 6LoWPAN node which periodically sends sensor data at a fixed rate, the duty cycle can be controlled by tuning the duration of the SI and the rate at which packets are transmitted. Under low workload, which is actually the target of our analysis since our aim is to assess the protocols in low-power mode, the average interval between data packets is much greater than the LPL cycle. Under this condition, the duty cycle of an LPL network can be calculated as

$$DC = \frac{AI + T_p \cdot N_{PPC}}{AI + SI} \tag{11}$$

where AI and SI are the duration of the activity interval and the sleep interval, respectively, T_p is the activity time due to a packet transmission, and N_{PPC} is the average number of packets per LPL cycle. Note that under low workload we have $N_{ppc} << 1$.

In order to find the correct T_p values, we inspected the TinyOS source code [20]. Specifically, looking at the `DefaultLPL.nc` file, it is possible to see that, for a transmitter node, the LPL cycle does not restart just after the end of the packet transmission, but after a delay D_{tx} which is constantly set to $20ms$. As a result, T_p can be expressed as

$$T_p = T_{preamble} + T_{pkt} + D_{tx}, \tag{12}$$

where $T_{preamble}$ is the duration of the preamble, which is equal to SI, T_{pkt} is the packet transmission time. The transmission duration is calculated as $\frac{L_{header}+L_{payload}}{R}$, where R is the data rate, which is fixed to $250kbps$, while L_{header} and $L_{payload}$ are the size (in bits) of the header and the payload, respectively. In all our experiments, the value of L_{header} and $L_{payload}$ is fixed to 128 and 320, respectively.

Further inspection on the TinyOS source code is necessary to find the value of the AI parameter in eq. (11). Looking at the `DefaultLPL.h` and `PowerCycleP.nc` files, we found that $AI = 8 \times T_{symbol} \times$ `MAX_LPL_CCA_CHECKS`, where `MAX_LPL_CCA_CHECKS` is a constant value defined equal to 400. As a result, we have an AI value of $51.2ms$.

Finally, the N_{PPC} parameter in eq. (11) can be obtained as

$$N_{PPC} = \frac{AI + SI}{T_{send}}, \tag{13}$$

where T_{send} is the packet send period.

Now it is possible to substitute the T_p and N_{PPC} parameters in eq. (11) with the expressions in eq. (12) and (13), respectively, to express the duty cycle of a 6LoWPAN node as a function of the network parameters. We obtain

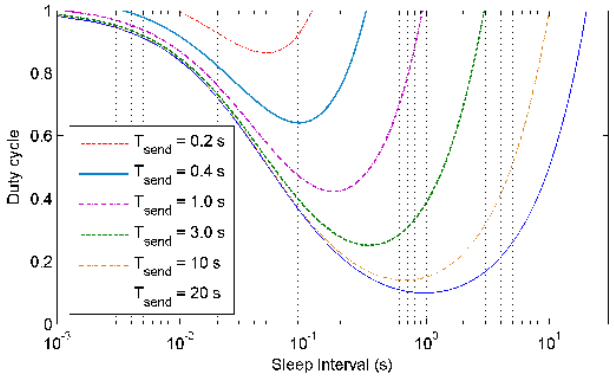

Figure 7. Duty cycle of a 6LoWPAN sensor node.

$$DC = \frac{AI}{SI + AI} + \frac{SI + T_{pkt} + D_{tx}}{T_{send}}. \qquad (14)$$

Figure 7 shows how the duty cycle can be changed by changing the sleep interval, under a given packet send period of the sensor node. Here it is possible to see that the duty cycle has a global minimum. Therefore, considering a node which has to transmit its sensor data with a given packet send period, it is possible to optimize the energy consumption by finding the minimum of eq. (14) with respect to SI. In particular, the optimal sleep interval SI^* that minimizes the duty cycle can be found by calculating the first derivative of eq. (14) with respect to SI and taking its non-negative zero. For $T_{send} \geq AI$, we obtain

$$SI^* = \sqrt{AI \cdot T_{send}} - AI. \qquad (15)$$

However from Figure 7 we also see that optimizing the sleep interval has an impact on the duty cycle which becomes smaller as soon as the send period decreases. This means that only a marginal improvement can be achieved by using LPL when the packet transmission rate is high, and it is possible to achieve high energy efficiency (i.e. low duty cycles) only under low workload.

5. Performance assessment

The comparative assessment is done in two different test scenarios, in which nodes have the same configuration, but the send period of the transmitting node (and therefore the workload) is different. The first scenario emulates a low-power sensing application where the monitored phenomenon has slow dynamics, where the sensor node sends data packets with a period of $20s$. The second scenario emulates a WSN application with faster dynamics, where the send period is $400ms$. It must be noted that the difference on the send period also changes the duty cycles in the case of 6LoWPAN.

The network topology in both the scenarios is the same as shown in Figure 3. All the nodes in the IEEE 802.15.4/ZigBee network share the same superframe settings and therefore the same duty cycle. Four different superframe settings have been used for the IEEE 802.15.4/ZigBee network, with different duty cycles and therefore different energy saving capabilities. The first setting uses a BO of 3, which is the smallest beacon order that provides sufficient transmission reliability. With this BO we use a SO of 1, which again is the smallest SO that provides sufficient transmission reliability. We obtain a 0.25 duty cycle. To further reduce the duty cycle, we also tested superframe configurations using a beacon order of 5. In this way, we were able to achieve duty cycles down to 0.06. The details for these configurations are shown in Table 2.

Table 2. Superframe settings for the IEEE 802.15.4/ZigBee networks

Config. Name	BO	SO	DC
ZigBee-1	3	1	0.25
ZigBee-2	5	1	0.06
ZigBee-3	5	2	0.12
ZigBee-4	5	3	0.25

The 6LoWPAN settings (BI and SI) have been selected so as to provide the sender node with the same duty cycle as for the IEEE 802.15.4/ZigBee network. Since such a duty cycle also depends on the send period, we used different configurations in the two scenarios.

5.1. Scenario 1 - $T_{send} = 20s$

Solving equation (14) with SI as the unknown and the value specified in Sect. 4.2 for the other parameters, we obtained that a SI of 162 and $492ms$ allow for a duty cycle of 0.25 and 0.12, respectively, with a send period of $20s$ and the standard AI of $51.2ms$. However, using these LPL parameters it is not possible to achieve a duty cycle of 0.06 such as in ZigBee-2 scenario. For this reason, we tried to decrease the AI parameter by halving the value of the `MAX_LPL_CCA_CHECKS` constant. In this way we achieved an AI of $25.6ms$, that makes it possible to achieve a duty cycle of 0.06, by using a SI of $606ms$. The LPL settings used for this scenario are summarized in Table 3.

Table 3. LPL settings of the 6LoWPAN Network

Config. Name	AI (ms)	SI (ms)	DC
6LoWPAN-1	25.6	606	0.06
6LoWPAN-2	51.2	492	0.12
6LoWPAN-3	51.2	162	0.25

The experiment consisted in sending 1300 packets and logging the transmission and reception times of every packet. The log files then were used to calculate three performance metrics:

- The *end-to-end delay*, calculated as the difference between the timestamp of a reception event and the timestamp of the relevant transmission event.
- The *update time*, which is the time interval between two consecutive packet reception events.
- The *delivery ratio*, which is the percentage of successfully received packets over the total number of packets sent.

5.1.1. End-to-end delay. The Cumulative Distribution Functions (CDF) of the experimental end-to-end delays for all our measurement campaigns are plotted in Figure 8, while the mean delay and the standard deviation are shown in Table 4. Here it is possible to see that the end-to-end delay of the IEEE 802.15.4/ZigBee network is greatly influenced by the beacon interval. In fact, the ZigBee-1 configuration with a beacon order of 3 has noticeable smaller delays than the ZigBee-4 configuration that features the same duty cycle but with a beacon order of 5. This is an expected result, as in a multihop cluster-tree network each

coordinator has to wait for the next active superframe of the parent (or child) coordinator before it can forward the packet. However, it is also possible to observe some effect of the superframe duration on the delay. In fact, the ZigBee-4 configuration has smaller delays than ZigBee-2 and ZigBee-3, which have smaller beacon intervals. On the other hand, the ZigBee-2 and ZigBee-3 configurations have almost the same delays.

The end-to-end delay of the 6LoWPAN configurations are also greatly influenced by the duty cycle. Considering the network configurations with duty cycle 0.25 and 0.12, we see that 6LoWPAN has a smaller mean delay, but with a noticeably larger standard deviation than IEEE 802.15.4/ZigBee. In Figure 8 it is possible to see that, considering the ZigBee-1 and 6LoWPAN-3 scenarios featuring a duty cycle of 0.25, although in the latter protocol the mean delay is smaller, the largest delay of ZigBee-1 is about $200ms$, which is far smaller than that of 6LoWPAN-3. However, when considering the scenarios which have a duty cycle of 0.12 (ZigBee-3 and 6LoWPAN-2), the 6LoWPAN protocol provides smaller delays than IEEE 802.15.4/ZigBee most of the times. Finally, from Table 4 it is possible to observe that decreasing the length of the AI in the 6LoWPAN-1 configuration has a seriously negative impact on the network performance, as both the mean delay and the standard deviation increase of some order of magnitude. As a result, the ZigBee-2 configuration with the same 0.06 duty cycle clearly outperforms 6LoWPAN-1.

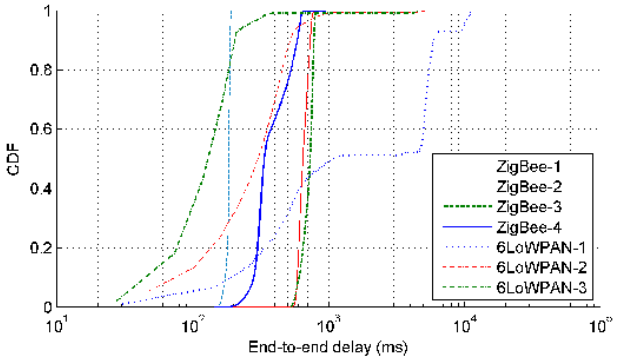

Figure 8. CDF of the end-to-end delay.

5.1.2. Update time and delivery ratio. Looking at the CDF of the update times, in Figure 9, we see that IEEE 802.15.4/ZigBee achieves smaller time intervals between consecutive packet receptions. Note that, due to the fact that TelosB nodes have a slightly shorter symbol time than the one foreseen by the standard, the actual send period is shorter than $20s$, and in particular it is about $19.07s$. Therefore, the smaller update times of the IEEE 802.15.4/ZigBee configurations are closer to those of the 6LoWPAN ones. Again, the 6LoWPAN-1 configuration shows far worse performance than all the other configurations. Finally we see that, in the ZigBee-1 configuration, there are some sporadic cases where the update time is twice the theoretical value. A closer look to Figure 9 reveals that this also happens for the other IEEE 802.15.4/ZigBee configuration, but with a smaller probability. This result is due to packet loss as, if a packet is lost, the update time is recorded at the next packet reception, and therefore the update time is doubled.

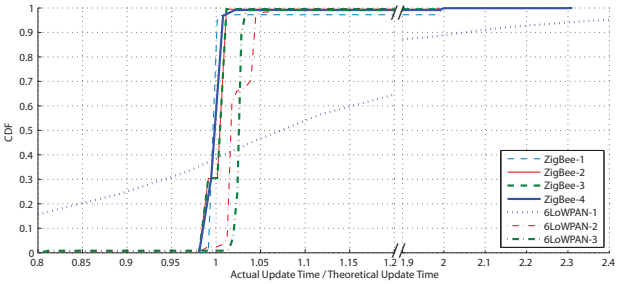

Figure 9. CDF of the update times.

Looking at the delivery ratio, in Figure 10, we see that the IEEE 802.15.4/ZigBee network experiences some packet loss, which increase when the beacon order decreases. A possible explanation for this behavior is that this packet loss is due to some synchronization glitches due to the variability of the actual beacon intervals discussed in Section 4.1. In the figure we also see that 6LoWPAN with the default AI ($51.2ms$) has the best performance in terms of received packets (but the delivery ratio of the ZigBee configurations with BO=5 is only marginally worse). On the contrary, the 6LoWPAN-1 configuration with the smaller AI has bad performance in all the respects.

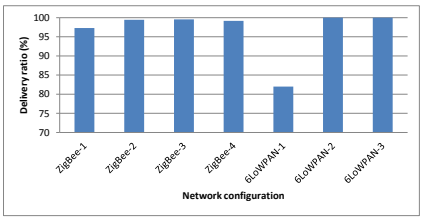

Figure 10. Delivery ratio.

5.2. Scenario 2 - $\mathbf{T_{send} = 400ms}$

The smaller send period of this scenario makes the 6LoWPAN-1 and 6LoWPAN-2 configurations unusable, because their sleep intervals are comparable to the send period and this would lead to network congestion. The 6LoWPAN-3 configuration is still usable, but the comparison with the other networks would be unfair, as the duty cycle of this configuration is 0.7 considering the $400ms$ send period. Looking at Figure 7 it is possible to see that duty cycles of 0.25 or smaller are not achievable at all with this send period and the default AI. On the other hand, we observed that decreasing the duration of the active intervals causes severe performance degradation. For this reason, we decided to compare the performance of the same IEEE 802.15.4/ZigBee configurations of Scenario 1 with the 6LoWPAN configuration that provides the smallest duty cycle for this send period, with the default AI. By using eq. (15), it is possible to see that the optimal SI is $92ms$, which provides a duty cycle of 0.64 with the default AI of $51.2ms$. We refer to this network configuration as 6LoWPAN-4.

The same performance figures seen for Scenario-1 were also calculated for Scenario-2. In the CDF of the end-to-end packet delays, shown in Figure 11, we see basically the same trends that were observed for Scenario-1. The delays of the IEEE 802.15.4/ZigBee configurations are al-

Configuration	DC	Update T. (mean)	Update T. (s.dev.)	Delay (mean)	Delay (s.dev.)
ZigBee-1 (BO=3,SO=1)	0.25	19603 ms	3121 ms	182.7 ms	6.8 ms
ZigBee-2 (BO=5,SO=1)	0.06	19178 ms	1405 ms	653.9 ms	52.7 ms
ZigBee-3 (BO=5,SO=2)	0.12	19163 ms	1303 ms	709.5 ms	65.3 ms
ZigBee-4 (BO=5,SO=3)	0.25	19242 ms	1811 ms	392.3 ms	117.1 ms
6LoWPAN-1 (AI=22.5,SI=606)	0.06	23805 ms	12240 ms	47035 ms	19260 ms
6LoWPAN-2 (AI=51.2,SI=492)	0.12	19532 ms	574.5 ms	334.5 ms	398.2 ms
6LoWPAN-3 (AI=51.2,SI=162)	0.25	19532 ms	551.2 ms	167.7 ms	389.5 ms

Table 4. Scenario-1 results.

most identical to those shown in Figure 8. This confirms that, under low workload, the performance are determined by the superframe settings, and in particular by the beacon order. In fact, the ZigBee-2, ZigBee-3, and ZigBee-4 configurations, which feature the same beacon order but different duty cycles, achieve the same delay performance. We can therefore conclude that a low-power WSN characterized by low workoad should keep the beacon intervals as small as possible, while taking care of the superframe overlapping probability and collapsing probability discussed in Section 4.1. On the other hand, the delay performance of the 6LoWPAN-4 configuration is better than that obtained in Scenario-1, although the workload of Scenario-2 is higher. This means that, under low workload, the 6LoWPAN performance are more affected by the duty cycle than by the actual workload. The main reason is that the SI which minimizes the duty cycle with the 400ms send period is smaller than the sleep interval value SI needed to achieve the desired duty cycles with the send period of Scenario-1. Looking again at Figure 7 it is possible to see that the optimal sleep interval increases its value (i.e., moves to the right) as soon as the send period decreases. In other words, there is a tradeoff between delay and energy consumption, since to decrease the duty cycle the sleep interval must be increased but, while doing so, also the delay increases. Compared to the IEEE 802.15.4/ZigBee configurations, 6LoWPAN/4 provides the smallest mean delays but the largest variance. In particular, in Table 5 we see that 6LoWPAN-4 has a 31% smaller mean delay than ZigBee-1 (which is the fastest ZigBee configuration we could use), but its standard deviation is at least one order of magnitude higher than any IEEE 802.15.4/ZigBee network we tried. As a result, the mean delay is smaller in the 6LoWPAN-4 configuration, while the largest delay is smaller in the IEEE 802.15.4/ZigBee configurations. Looking at Figure 11 we observe that ZigBee-1 configuration provides a smaller delay also when considering a statistical upper bound which covers more than 97% of the observations. The CDF of the update times is shown in Figure 12. Here it is possible to see that the ZigBee-1 scenario provides slightly better performance (i.e., actual update times closer to the theoretical ones) than 6LoWPAN-4. Both ZigBee-1 and 6LoWPAN-4, however, provide better results then the IEEE 802.15.4/ZigBee configurations with a BO of 5. In fact, the update times CDF of these configurations reveals that sometimes multiple packets are received within a very short time interval (less than 10ms). The reason for this behaviour is that in this scenario the send period is smaller than the beacon interval (which should theoretically be 491.5ms, but it is actually 468.7ms due

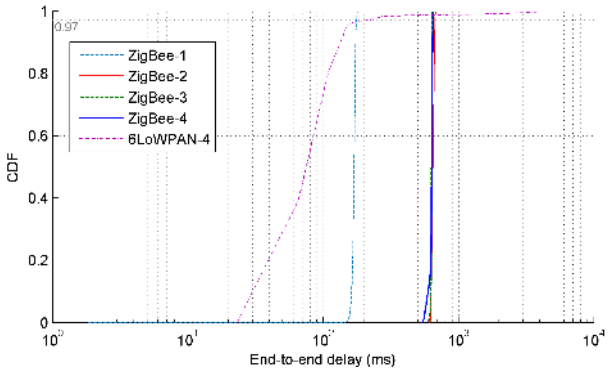

Figure 11. CDF of the end-to-end delay.

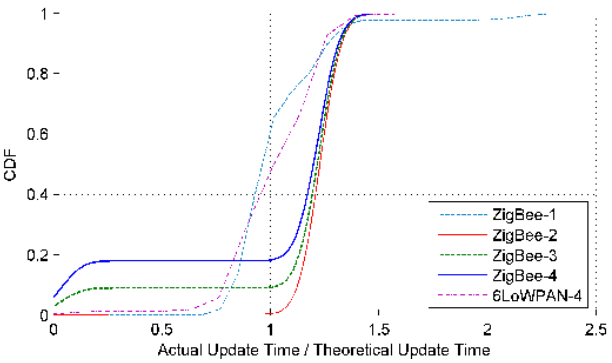

Figure 12. CDF of the update times.

to the TelosB hardware timers). Under these conditions, it happens that two data packets are produced within the same beacon interval. Even though these packets are generated at different times, they will be both transmitted at the next active superframe. To avoid this problem, we suggest to always use beacon intervals smaller than the send period.

The delivery ratio is given in Figure 13. The figure shows that, like in Scenario-1, the 6LoWPAN configuration with the default AI provides the highest communication reliability. Among the IEEE 802.15.4/ZigBee configuration, those with a BO value of 5 achieve slightly better performance than ZigBee-1, whose BO is 3. However, the percentage of received packets is quite high in all the configurations.

6. Conclusions

This paper addressed a comparative performance assessment of the IEEE 802.15.4/ZigBee and 6LoWPAN protocols for industrial WSNs. The paper offers several contributions. First, a theoretical analysis of the low-power

Configuration	DC	Update T. (mean)	Update T. (s.dev.)	Delay (mean)	Delay (s.dev.)
ZigBee-1 (BO=3,SO=1)	0.25	390.5 ms	95.8 ms	168.7 ms	4.6 ms
ZigBee-2 (BO=5,SO=1)	0.06	467.2 ms	26.1 ms	647.6 ms	13.2 ms
ZigBee-3 (BO=5,SO=2)	0.12	426.3 ms	130.4 ms	632.5 ms	5.6 ms
ZigBee-4 (BO=5,SO=3)	0.25	384.2 ms	173.9 ms	627.9 ms	20.9 ms
6LoWPAN-4 (AI=51.2,SI=92)	0.64	390.6 ms	160.8 ms	114.9 ms	294.9 ms

Table 5. Scenario-2 results.

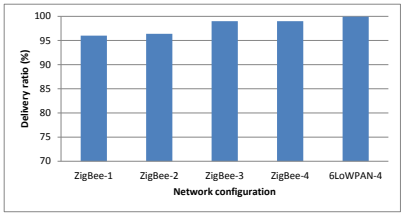

Figure 13. Delivery ratio.

characteristics of the two protocols, that is used as a basis for a methodology to tune the low-power relevant configuration parameters of the two protocols (i.e., the BO and SO for 802.15.4/ZigBee, the AI and SI for 6LowPAN) so as to achieve a desired duty cycle in a realistic testbed built with COTS hardware and open source software. Second, comparative experimental assessments on such a testbed, that highlight the strengths and weaknesses of both protocols when working in low-power mode. From these assessments useful hints for the network designer are derived, that allow him to select of the technology that best matches the performance goal to achieve (either a given duty cycle or a given maximum delay or packet loss). The paper shows that the IEEE 802.15.4 protocol allows a network designer to perform energy planning by setting the duty cycle of nodes, but the tuning has to be carefully made because of possible superframe overlapping problems. On the other hand, 6LoWPAN does not require any synchronization between nodes, but the nodes duty cycle depends on the overal network load, and this makes it more difficult to plan the duty cycle and therefore the energy consumption.

The results of the experimental assessments show that the IEEE 802.15.4/ZigBee network is able to support smaller duty cycles and provide smaller maximum end-to-end delays and update times slightly closer to the theoretical value than 6LowPAN. On the other hand, the 6LoWPAN network shows smaller mean end-to-end delays and higher reliability, i.e. a lower percentage of packet loss.

Future work will extend the comparison to different network topologies and different sets of parameters for both the protocols. Moreover, other low-power communication protocols will be also addressed.

References

[1] "ZigBee Specification", *ZigBee Document 053474r17 Version*, , pp. 1–576, 2008.

[2] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks", in *Proc. 2nd intl conf. on embedded networked sensor systems*, SenSys '04, 2004, p. 95–107, USA. ACM.

[3] S. Pollin et al., "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer", *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, 2008.

[4] Y. Huang, A. Pang, and H. Hung, "A comprehensive analysis of low-power operation for beacon-enabled IEEE 802.15.4 wireless networks", *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5601–5611, Nov. 2009.

[5] E. D. Pinedo-Frausto and J. A. Garcia-Macias, "An experimental analysis of Zigbee networks", in *IEEE Conf. on Local Computer Networks*, Oct. 2008, pp. 723–729.

[6] G. Mulligan, "The 6LoWPAN architecture", in *Proc. 4th workshop on Embedded networked sensors*, EmNets '07, 2007, p. 78–82, USA. ACM.

[7] C. Y. Yum et al., "Methods to use 6LoWPAN in IPv4 network", in *The 9th Intl Conf. on Advanced Communication Technology*, volume 2, 2007, pp. 969–972.

[8] S. Bocchino et al., "SPEED routing protocol in 6LoWPAN networks", in *IEEE Conf. on Emerging Technologies & Factory Automation (ETFA)*, Sept. 2011.

[9] J. W. Hui and D. E. Culler, "IPv6 in Low-Power Wireless Networks", *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1865–1878, Nov. 2010.

[10] M. Kovatsch, M. Weiss, and D. Guinard, "Embedding internet technology for home automation", in *IEEE Conf. on Emerging Tech. and Factory Automation (ETFA)*, 2010.

[11] S. Sasidharan, F. Pianegiani, and D. Macii, "A protocol performance comparison in modular WSNs for data center server monitoring", in *Intl Symp. on Industrial Embedded Systems (SIES)*, July 2010, pp. 213–216. IEEE.

[12] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)", 2006.

[13] C. E. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", http://www.ietf.org/rfc/rfc3561.txt.

[14] N. Kushalnagar, G. Montenegro, D. E. Culler, and J. W. Hui, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", http://tools.ietf.org/html/rfc4944.

[15] M. Sha, G. Hackmann, and C. Lu, "Energy-Efficient Low Power Listening for Wireless Sensor Networks in Noisy Environments", Technical Report 2011-61, 2011.

[16] J. Hauer, "TKN15.4: An IEEE 802.15.4 MAC Implementation for TinyOS 2", TKN Technical Report Series TKN-08-003, Telecommunication Networks Group, Technical University Berlin, Mar. 2009.

[17] J. Ko, S. Dawson-Haggerty, O. Gnawali, D. Culler, and A. Terzis, "Evaluating the Performance of RPL and 6LoWPAN in TinyOS", in *Proc. Workshop on Extending the Internet to Low power and Lossy Networks*, Apr. 2011.

[18] E. Toscano and L. Lo Bello, "Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks", IEEE Trans. Ind. Informat., available on line `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6009195`.

[19] A. Hernandez and P. Park, "IEEE 802.15.4 Implementation based on TKN15.4 using TinyOS", Technical report, KTH Electrical Engineering, Stockholm, Jan. 2011.

[20] "TinyOS Home Page", http://www.tinyos.net/.