Performance Evaluation of an Energy-Efficient MAC Scheduler by using a Test Bed Approach

Daniele Alessandrelli, Luca Mainetti, Luigi Patrono,
Giovanni Pellerano, Matteo Petracca, and Maria Laura Stefanizzi

***Abstract*: A Wireless Sensor Network consists of a large number of sensor nodes that are usually battery powered and deployed in large areas in which changing or recharging batteries may be impractical or completely unfeasible. Therefore, energy efficiency represents one of the main design objectives for these networks. Since most of the energy is consumed by the radio communication, the development of Medium Access Control protocols able to minimize the radio energy consumption is a very attractive research area. This paper presents an energy efficient communication protocol and its implementation in the Contiki Operating System. The performances and the portability of the proposed solution are thoroughly evaluated by means of both simulations, carried out using the Contiki simulation tools (i.e., Cooja and MPSim), and test beds based on two different platforms. Obtained results show that the proposed scheme significantly reduces the sensor nodes power consumption compared to the IEEE 802.15.4 standard solution already implemented in Contiki.**

***Index terms*: Wireless Sensor Networks, MAC protocol, Contiki, Test bed, Performance evaluation.**

I. INTRODUCTION

The next-generation of Internet aims at integrating heterogeneous wired and wireless communication technologies in order to enable the Internet of Things (IoT) [1] concept. According to the IoT vision, the everyday objects that surround us will become proactive actors of the global Internet, with the capability of generating and consuming information to be used to perform advanced applications. Wireless Sensor Network (WSN) is one of the most important technology to be considered for a full accomplishment of the IoT paradigm. In general terms, a WSN is composed of a large number of spatially distributed sensor nodes that cooperate among them in order to monitor, collect, process and share data acquired from the environment in which they are deployed. The recent progresses in wireless technology and embedded systems have enabled the development of low-cost, low-power, multifunctional sensor nodes characterized by ad hoc communication. This main feature makes WSNs suitable for a wide range of potential applications including environmental monitoring, home and building automation, healthcare, and robotic exploration.

However, the realization of such applications requires the use of efficient power management techniques. Indeed, sensor nodes are usually battery powered and deployed in large areas in which changing or recharging batteries is impractical or completely unfeasible. Therefore, energy consumption is a primary issue to be considered, and the use of effective solutions for increasing the lifetime of WSN nodes is fundamental in real applications.

The main activities performed by a WSN node when deployed in a real scenario consist of sensing, data processing and communication. Among them, the last mentioned must be considered the most remarkable in terms of energy consumption, therefore several communication protocol enhancements aiming at minimizing the energy consumption have been proposed in literature. However, when considering radio communication, the most prevalent source of energy waste is the idle listening [2] (i.e. listening to the channel in absence of communications). Consequently, the design of a Medium Access Control (MAC) protocol able to carefully coordinate the nodes receiving time can significantly improve the network lifetime.

In literature, most of the proposed MAC solutions are validated only by means of mathematical or simulation methods, which often show significant limits. The use of a test bed approach, on the contrary, overcomes these theoretical lacks, and is able to guarantee that a proposed protocol solution achieve the desired results on real devices. Moreover, this approach does not demonstrate only the efficiency but also proves the feasibility of the proposed solution. In fact, a protocol able to ensure excellent performance in a simulated environment may present implementing problems on hardware platforms or may reveal highly degraded performance in a real environment. These differences in overall performance are due to the difficulty of simulating the actual behavior of a WSN, which often depends on events that are not controllable in a mathematical or simulation model (e.g., software installed in sensor devices, atmospheric and environment conditions, etc.).

Manuscript received December 15, 2012; revised February 19, 2013.

The material in this paper was presented in part at the 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2012), Split, Croatia, Sept, 2012.

L. Mainetti, L. Patrono and M.L. Stefanizzi are with the Department of Innovation Engineering at University of Salento, Lecce, Italy (email: {luigi.patrono, laura.stefanizzi }@unisalento.it).

D. Alessandrelli is with the Real-Time Systems Laboratory at Scuola Superiore Sant'Anna, Pisa, Italy (email: d.alessandrelli@sssup).

G. Pellerano and M. Petracca are with the National Laboratory of Photonic Networks at National Inter-University Consortium for Telecommunications, Pisa, Italy (email: {giovanni.pellerano, matteo.petracca}@cnit.it).

Furthermore, to the best of authors' knowledge, almost all of the MAC protocols for WSNs presented in literature are validated by using only one kind of sensor nodes, despite the most of sensor nodes now available on the market. Since such nodes are characterized by very different hardware features, especially in terms of computation capabilities, communication range and power consumption, a protocol optimized for a specific device may not work properly on a different hardware platform and its porting may require a significant efforts. Therefore, code portability represents a very important design objective for WSNs, and the only way to ensure the effective portability of the developed solutions is to validate them by using multiple platforms with different hardware features. In addition, a comparative study performed by using multiple devices may be useful to provide important feedback for the design of new sensor boards.

In this work, an energy efficient MAC duty cycling mechanism based on an asynchronous scheduler is presented and validated by using both simulation and test bed approaches. The proposed scheduler is an enhancement of the solution reported in [3]. Specifically, this new work addresses the clock drift problem and takes into account technical requirements related to two different boards. The resulting solution is able to reduce energy consumption by preventing unnecessary awakenings of nodes. Each node communicates to its neighbors the time interval during which it will transmit, so that they can set their wakeup time accordingly. This way, every node has the complete list of the transmission times of its neighbors and knows in advance when it must be awake for receiving data, and when it can switch to sleep mode when no transmissions are scheduled. To solve the clock drift problem, a node updates its neighbors list every time it receives a new data packet. The proposed solution is also able to face topology changes. Indeed, every node updates its neighbor list as soon as it is not able to listen a transmission of a neighbor or it detects the presence of a new node.

In order to analyze the effectiveness of the proposed protocol, a performance comparison by using two different kinds of sensor node running the Contiki Operating System [4] was carried out. As a propaedeutic step, an intensive optimization campaign, using the Contiki simulation environment was conducted. Such a study took into account different parameters and network topologies. Finally, the network power consumption of the proposed MAC scheduler was compared with the current IEEE802.15.4 [5] standard solution by means of two test beds based on two different sensor boards: MB851 [6], and Seed-eye [7].

The rest of the paper is organized as follows. Section II summarizes the state-of-the-art for energy consumption minimization in WSNs and the use of test bed approaches. The proposed scheduler is described in Section III. The test bed environment is presented in Section IV, while in Section V numerical results are discussed. Conclusions are drawn in Section VI.

II. Overview of Energy-Aware MAC Protocols and Validation Techniques

This section summarizes the most important MAC protocols based on the duty cycle mechanism presented in literature and the main techniques used to validate them. All protocols fit into three main categories: *preamble-sampling*, *scheduling*, and *hybrid* approaches.

Preamble-sampling MAC protocols, such as B-MAC [8], WiseMAC [9], and X-MAC [10], exploit the Low-Power Listening (LPL) [11] technique for sampling packet preambles. LPL minimizes the duty cycle when there are no packets to be exchanged, but during transmissions it needs a preamble longer than the wakeup interval to guarantee the detection of the channel activity at the receiver side. Even if B-MAC uses unsynchronized duty cycling for reducing idle listening, it uses a preamble longer than a sleep period, thus allowing other nodes in the communication range to overhear the channel and consume more energy. Furthermore, WiseMAC reduces the length of the wakeup preamble by exploiting knowledge of the sampling schedule of direct neighbors. X-MAC solves the overhearing problem (i.e. listening of messages addressed to another node) by using a strobed preamble that consists of a sequence of short preambles prior to data transmission.

S-MAC [12] and T-MAC [13] are scheduling MAC protocols. They synchronize the wakeup schedules of sensor nodes in a neighborhood by exchanging synchronization packets. Such messages exchange results in high overhead and significant energy consumption even when there are no useful data to send. S-MAC is the first duty cycling MAC protocol designed for WSNs. It is characterized by a distributed coordination scheme able to synchronize node sleep schedules in a multi-hop network. T-MAC improves S-MAC with an adaptive timer for the reduction of the wakeup duration, and with the introduction of future requests to send policy. A more recent work, focusing on the scheduling approach, is the PW-MAC [14] protocol. It improves both S-MAC and T-MAC exploiting a scheduler based on a pseudo random algorithm, which allows the sender to predict the next wakeup time of the receiver node.

Hybrid approaches, such as SCP-MAC [15] and AS-MAC [16], combine preamble sampling with scheduling techniques. SCP-MAC minimizes the length of the preamble by exploiting the synchronization of the wakeup time of neighboring nodes. It reduces the energy consumption but it is not able to avoid the overhearing problem. AS-MAC uses LPL to minimize the periodic wakeup time, and it also asynchronously coordinates the wakeup times of neighboring nodes to reduce overhearing, contention, and delay. One of the main disadvantages of this protocol is the inefficiency in broadcast transmissions since AS-MAC has to transmit every packet once for each neighbor.

Many works describe new protocols validated by mathematical or simulation approach. These techniques often show an incomplete analysis because they do not take into account the physical requirements of real boards. In order to overcome this problem, a test bed approach is needed. In [12], the authors developed S-MAC on two different platforms (i.e., the Rene Mote and the Mica Mote) and demonstrated its effectiveness by a real test bed. The motes were running TinyOS [17], an efficient event-driven operating system for tiny sensor nodes. The goal of the experimental campaign was to reveal main trade-offs among energy, latency, and throughput. For this purpose, they used two different network topologies: a two-hop network with two sources and two sinks, and a ten-hop linear network with one source and one sink. In [15], a performance comparison, in terms of average power consumption, between SCP-MAC and LPL protocols is reported. In particular, the comparison was performed considering 10 nodes forming a single hop mesh. All nodes were in the same communication range with the possibility to communicate with each other. Each node periodically generated a 40 byte long data message (not including preamble) to be broadcasted at several transmission rates in order to study MAC performance. In [16], authors implement the AS-MAC protocol in TinyOS on MicaZ [18] motes and compare its performance with an existing implementation of SCP-MAC. They considered three evaluation metrics (i.e., energy consumption, packet loss, and delay), and two different network topologies (i.e., single-hop and multi-hop topology). Specifically, for the single-hop experiment, they use a star topology consisting of one receiver and up to five senders. For the multi-hop experiments they set up a chain network with six nodes, with the sink at the end of the chain. Five TelosB motes were used in [10] to compare the X-MAC and the LPL protocols. Also in this work, authors used a star and a chain topology to analyze duty cycle and latency of the considered solutions.

According to state-of-the-art solutions previously described, the proposed MAC scheduler is based on a duty cycling mechanism to avoid idle listening problem while asynchronously coordinating the wakeup times of neighboring nodes. The proposed solution reduces both the number of synchronization packets typically used in scheduler-based solutions, and the overhearing problem that affects preamble-sampling MAC protocols. Indeed, the nodes store the scheduled wakeup times of their neighbors, therefore, they do not need to add long preambles when sending data packets. The performance and the portability of the proposed solution are evaluated by means of both simulations and test beds based on two different hardware platforms. In each test bed, both single-hop and multi-hop scenarios are considered.

III. Proposed MAC Scheduler

The basic idea of the proposed scheduler is that nodes wake up periodically to transmit packets, and at scheduled time instants to receive messages from their neighbors. To solve the hidden node problem, a node validates the awakening times announced by its neighbors and alerts them of possible collisions. When the radio is on, a node either transmits data and receives acknowledgment messages (ACKs), or receives data from other nodes and sends ACKs. Before describing the scheduler, some parameter definitions are introduced:

- T_0 is the time interval (in seconds) between two subsequent transmissions. It is the same for every node and it is preconfigured.
- *WakeTime* is the time interval (in seconds) in which a node can transmit the local buffered data or receive data from its neighbors.
- *Announce Packet* (Pkt_{ANN}) is a signaling packet used by each node to advertise its presence; it contains the time interval between its transmission time and the next chosen awakening time to be used for transmission.
- *Alert Packet* (Pkt_{ALERT}) is a signaling packet used by a node to alert a neighbor about a possible collision.
- *Full Packet* (Pkt_{FULL}) is a signaling packet used by a node to inform its neighbors that it is out of the network.
- *Wake-up Table* (W_{TBL}) is a table used by each node to store information about the transmission times of its neighbors. Each table entry is associated with exactly one neighbor and contains the following information: (a) the ID of the neighbor, (b) the offset of the awakening time, and (c) the number of cycles of length T_0 during which no data have been received from the corresponding node.

In the following, the start-up phase and the periodic listening and sleep phase are described.

A. Network Start-up

During the network initialization phase, all nodes stay awake for the period of time necessary to set the network parameters and they exchange information about their transmission time by sending Pkt_{ANN}s. On the reception of such a message from an unknown neighbor, a node updates its W_{TBL} by inserting a new entry. However, before being stored, the information on the transmission time of the neighbor must be validated: the node verifies that the time chosen by the new neighbor does not overlap with the transmission intervals of the other neighboring nodes already stored into its W_{TBL}. Let us observe that the offset stored into the new entry is obtained by subtracting an appropriate time interval. This time interval, takes into account the processing, transmission and propagation time of the packet. The entries are stored in the table in an ascending order based on the offset. Otherwise, if the transmission interval chosen by the new node overlaps with any of the transmission intervals already in W_{TBL}, the node sends a Pkt_{ALERT} to the new node, specifying the overlapping interval. In such a case, the new neighbor stores the received information into its W_{TBL} and chooses a new transmission time. The Pkt_{ALERT} packet has been introduced to greatly reduce the hidden node problem that is one of the main problems that afflict ad hoc networks. By this way, in fact, collisions among nodes two hops away are avoided. If the new node cannot find a valid transmission time, i.e., the network is full, it communicates the information by broadcasting a Pkt_{FULL} and turns off the radio. On the reception of such a message, nodes delete the corresponding entry from their W_{TBL}. After listening to the channel for a period of time equal to $2*T_0$ to detect the announcements of its neighbors, a node chooses its transmission time and sends its Pkt_{ANN}.

Note that the time is divided into fixed slot time with duration T_0. Therefore, a node must choose its transmission time within this period. In particular, each node chooses its own transmission time as a random value in a proper interval, also taking into account the choices done by its neighbors. This time differentiation permits to reduce the channel access contention. If the W_{TBL} is empty, then this value is randomly selected in the interval

$$\left[0, T_0 - (WakeTime + 2*TurnAroundTime)\right] \qquad (1)$$

where *WakeTime* is the transmission time and *TurnAroundTime* is time required by the radio needs to change its state. If the W_{TBL} is not empty, then every node tries to set its transmission time to a value different from those of its neighbors in order to avoid collisions due to simultaneous transmissions. This value is chosen so that the time interval reserved for the transmission (*WakeTime*) does not overlap with the transmission time of any neighbor. The node chooses the two consecutive entries in the table, *i-th* and *(i+1)-th*, whose offsets difference is maximum and checks if this difference is greater than:

$$2*WakeTime + 4*TurnAroundTime \qquad (2)$$

Note that the node also checks the time intervals:

$$[0, offset[0]] \quad \text{and} \quad [offset[n], T_0 - D] \qquad (3)$$

where *offset[0]* and *offset[n]* are the offsets associated to the first and last entry respectively, while *D* = *WakeTime* + *2*TurnAroundTime*. If this is the case, then the transmission time is chosen within the interval:

$$[offset[i] + D, offset[i+1] - D] \qquad (4)$$

where *offset[i]* and *offset[i+1]* are the offsets associated to entry *i* and *i+1* respectively.

In order to maximize the probability that all its neighbors receive the message, a node sends the Pkt_{ANN} three times. After this announcement phase, if it has not received any Pkt_{ALERT}, the node inserts a new entry in the W_{TBL} containing its own address and the scheduled transmission time. To better clarify the node behavior in this phase, a simplified flow chart is shown in Fig.1.

Analyzing more in detail how the start up phase works, it can be stated that the radio transceiver switches among the following macro-states: IDLE (inactive), CCA (channel contention), TX, RX and OFF. The first four states are those visited also by MAC layer. Fig. 2 shows the state diagram of the PHY layer and highlights the main actions that trigger a state change. Let us observe that, during the initialization phase, the PHY and MAC layers of the node are in IDLE state for most of the time, and they switch to RX state only when a packet is detected on the channel. Moreover, since nodes do not still have a dedicated transmission time, they have to start a contention (i.e., both MAC and PHY in CCA state) with the other neighbors for the channel access in order to send their advertisement packets (i.e., Pkt_{ALERT}, Pkt_{FULL}, Pkt_{ANN}).

B. Steady State

After the warm-up period, a node performs the periodic listening and sleep phase. This phases manages two kind of periodic events, namely the transmission and the reception of a packet, and one aperiodic event, i.e., the arrival of a new node in the network.

The activity of each node is ruled by its W_{TBL}. For every periodic event the node switches on its radio, and handles the correspondent event before switching off again the transceiver.

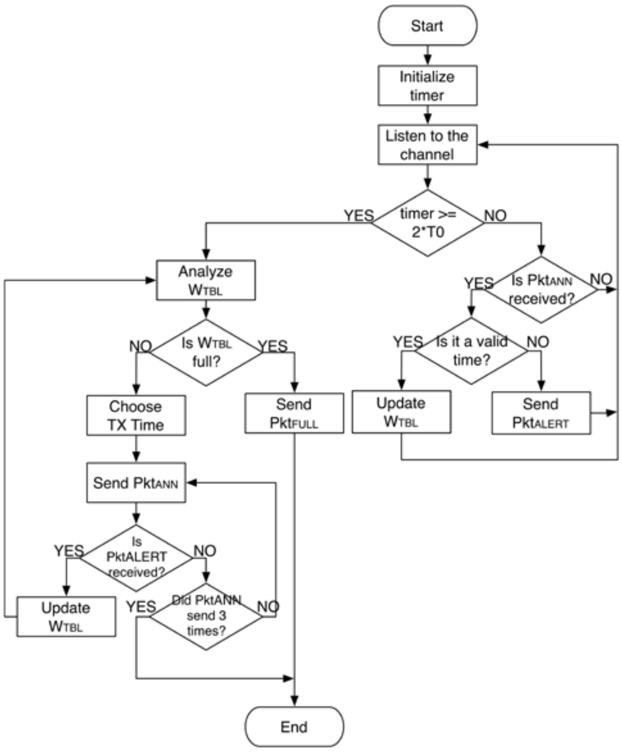


Fig.1. Flow diagram of the network start up phase.

The two periodic events are: (i) the transmission of a buffered packet (every T_0 seconds), and (ii) the reception of a packet from a neighbor that has a scheduled transmission. The node sets a timer for the next scheduled event in its W_{TBL}. When the timer expires, if the event scheduled is a data transmission (the entry contains its own address) the node wakes up, checks the presence of packets to be transmitted in its queue, and starts the transmission if it is required. When the transmission ends, the node waits for an ACK from the intended receiver. If no ACK is received, the message must be sent again. At the end of its transmission interval, the node schedules the next event and switches off its radio. When the scheduled event is a data reception (the entry contains the address of a neighbor), the node switches on its radio and starts listening the channel. If a packet is received, it sends an ACK, and at the end of the receiving interval, it switches off the radio. If nothing is received the node updates the packet-missed counter for the

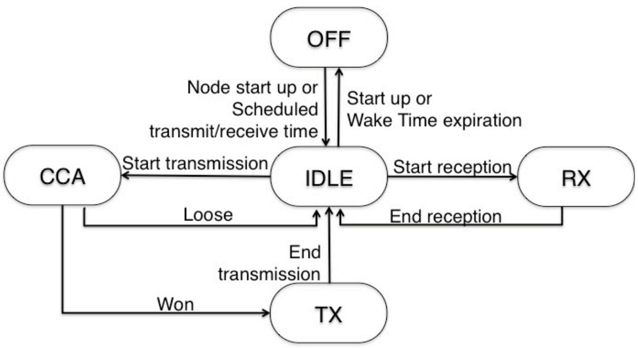


Fig.2. PHY layer state diagram.

corresponding W_{TBL}. After a fixed number *m*, of consecutive missed receptions, the entry is removed to avoid useless awakenings (i.e., the neighbor corresponding to that entry is supposed to have left the network). A simplified flow chart of the node behavior in this phase is shown in Fig. 3.

When a new node joins the network, it first listens the channel for an interval of time equal to $2*T_0$ with the aim of detecting the transmissions of its current neighbors. For each packet received from an unknown node it adds an entry in its W_{TBL}. Afterward, it announces its presence by sending a Pkt_{ANN} to each neighbor. To allow a new node enough time to send its Pkt_{ANN}s, nodes already members of the network always delay the transmission of their data packets of a period of time δ. This time interval is used by the new node to transmit its Pkt_{ANN}s.

As shown in Fig.2, also during the steady state phase the radio transceiver switches among the five states of CCA, IDLE, TX, and RX and OFF. This last one is the state leading to the actual power saving results. Let us observe that in this phase each node transmits its data during its transmission interval and, therefore, channel collisions are avoided. However, the CCA state is provided to guarantee the compatibility with the IEEE 802.15.4 standard.

Finally, an important issue to consider is the clock drift of nodes' timer. It is managed updating W_{TBL}, and considering as new WakeTime the time instant at which a packet is received. The update is performed only if the new WakeTime is a valid value, i.e., there are no overlaps between transmission slots. Otherwise, a Pkt_{ALERT} is sent to the node with the wrong transmission slot to notify that a new association procedure is necessary.

IV. Test Bed Environment

The effectiveness of the MAC scheduler described in Section III is evaluated by using two test beds carried out by two different research groups: the IDentification Automation Lab (IDA Lab) of Salento University in Lecce, and the Real-Time Systems Laboratory of Scuola Superiore Sant'Anna in Pisa. In each test bed, a different hardware platform is used. This choice permits to analyze the portability of the MAC scheduler implementation as a function of the hardware characteristics (e.g., clock speed, memory) and architectures (e.g., ARM, MIPS).

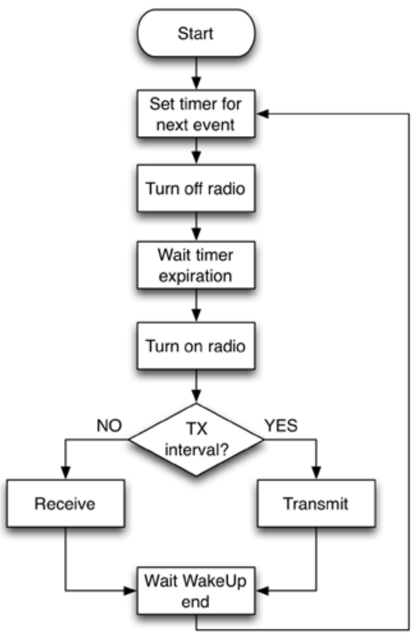


Fig.3. Flow diagram of the steady state phase.

In the following, the two selected hardware platforms are presented, and, afterward, the MAC scheduler implementation in Contiki OS and the test beds settings are discussed.

A. Hardware platforms

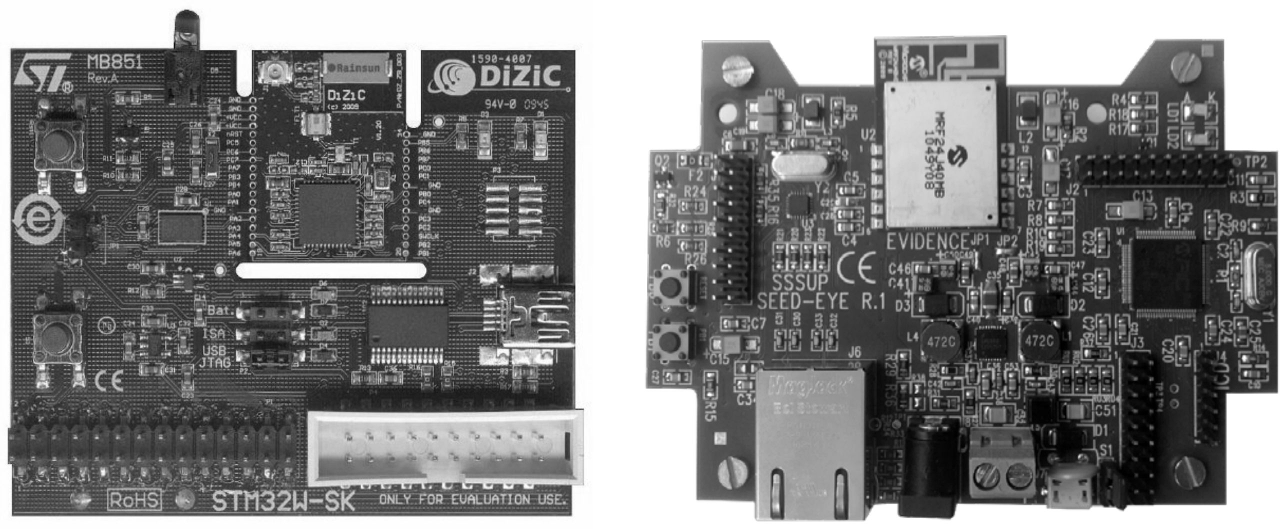
The first board selected for evaluating the scheduler performance is the MB851 (Fig. 4.a) developed by ST Microelectronics. It is equipped with a 32-bit ARM® Cortex™-M3 microcontroller operating at a clock frequency up to 24 MHz and embedding 16 Kbytes of RAM and 256 Kbytes of eFlash as ROM. Moreover, the board integrates a 2.4 GHz wireless transceiver compliant with the IEEE 802.15.4 standard and providing in hardware some MAC features. The mounted microcontroller is highly optimized to guarantee high performance at very low power consumption.

The second board selected for experimental purposes is the Seed-eye [7] (Fig. 4.b) developed by Scuola Superiore Sant'Anna and Evidence S.r.l.. It mounts a 32-bit Microchip™ PIC32MX795F512L microcontroller based on the MIPS architecture. The maximum clock speed is equal to 80 MHz with 128 Kbyte of RAM and 512 Kbyte of ROM installed on board. Wireless communication capabilities are provided by the Microchip™ MRF24J40MB transceiver, which is fully compliant with the IEEE 802.15.4 standard. The transceiver operates in the 2.4 GHz frequency band with a maximum transmission power of +20 dBm.

B. MAC scheduler implementation

The proposed MAC scheduler has been implemented as additional module of the Contiki OS communication stack. Contiki is a popular open-source operating system targeted to small microcontroller architectures developed by the Swedish Institute of Computer Science. The Contiki communication stack is organized in several layers in which both protocol solutions and radio transceiver features can be easily configured.

The lowest layer of the stack is the *NETSTACK_CONF_FRAMER*. It is in charge of the data packet format conversion before the transmission over the physical channel. The upper layer is the



(a) (b)

Fig.4. The evaluation boards used for test beds: (a) MB851 board at Salento University, (b) Seed-eye board at Scuola Superiore Sant'Anna

NETSTACK_CONF_RADIO. It directly manages the wireless transceivers features through the appropriate device driver. These two first levels can be considered the PHY layer of the ISO/OSI model. The third layer of the Contiki stack is the *NETSTACK_CONF_RDC*, which cannot be directly mapped to the ISO/OSI model. It is just below the MAC layer, identified as *NETSTACK_CONF_MAC*, and it is in charge of managing the radio duty cycling to provide energy saving capabilities. The last layer of the stack is the *NETSTACK_CONF_NETWORK* providing the functionality of the network layer of the ISO/OSI model.

Considering the above described communication stack architecture, the MAC scheduler has been developed as additional module of the NETSTACK_CONF_RDC layer. From an implementation point of view, the new RDC module exposes Contiki-based Application Programming Interfaces (APIs) to the upper layer while using the ones provided by the radio drivers. In order to develop a software solution to be used on several architectures, the scheduler timing has been implemented as completely independent from the system clock. In the implementation phase, a very big issue encountered was on the transceiver timing in switching from the active to the sleep state. Although in a protocol design the time necessary to switch among states is not considered, the problem cannot be neglected during real implementations. Indeed, the switch among internal states of the device needs to satisfy stringent delay constraints. If they are not respected in the driver implementation the radio transceiver may experience an internal block, thus making impossible further wireless communications.

C. Test bed settings and data collection scenario

As previously introduced, the performance of the MAC scheduler was evaluated by means of two real test beds. As a preparatory step to a real system deployment, several simulation campaigns were run to evaluate the impact of both T_0 parameter and packet rate. In all simulations both a single-hop and a multi-hop scenario were considered. In particular, for the single-hop analysis a star topology consisting of one receiver and four senders was set. In this first scenario, all nodes were in the same communication range with the sink in the center. Instead, for the multi-hop experiments, a chain network of five nodes was used. In both network topologies, node 1 was the sink and in all performed simulation campaigns each node sent 100 packets towards the sink at a Constant Packet Rate (CPR). To analyze the protocol behavior with different levels of network load, three different data rates were simulated: 1 packet every 5 seconds (high load), 1 packet every 30 seconds (medium load), and 1 packet per minute (a typical data rate used in sensor networks [19]). For each of selected packet rate, three values of T_0 were considered. In all simulations, a static routing protocol was used and, therefore, the performance of the proposed solution are not affected by routing traffic overhead. The main simulation parameters are reported in Table I, while the results of the performed analysis are discussed in the next section.

The same network topologies and transmission data rates considered in the simulation campaigns were also used in both test beds, while only the most suitable values of T_0 were considered. The test bed at the Salento University (called MB851_TEST in the rest of the paper) was carried out in an outdoor environment inside the campus. As shown in Fig. 5.a, a soccer field, without buildings in the surrounding area was used. The five MB851 devices used in the experiments were positioned at a height of 1.5 m, so as to limit the multipath problem due to the ground. Moreover, in order to obtain good communication links only between neighboring nodes the transmission power of each node was reduced to the value of -8 dBm and the nodes were placed about 10 meters apart. The test bed at the Scuola Superiore Sant'Anna (called Seed_TEST in the rest of the paper) was carried out in a parking space close to the main building of the Real-Time Systems Laboratory. In such a deployment, the outdoor condition is

TABLE I
SIMULATION PARAMETERS

Parameter	**Value**
Network Topology	Star, Chain
Number of nodes	5
Number of packets	100
WakeTime	160 ms
(Rate, T_0)	(1 packet every 5 seconds, 5) (1 packet every 5 seconds, 10) (1 packet every 5 seconds, 15) (1 packet every 30 seconds, 30) (1 packet every 30 seconds, 60) (1 packet every 30 seconds, 90) (1 packet every 60 seconds, 60) (1 packet every 60 seconds, 120) (1 packet every 60 seconds, 180)

kept fixed, however, the surrounding area is quite different, with a building and several cars close to the deployment area. A picture of the second test bed is reported in Fig. 5.b. In such an experiment, the same setting parameters used in the first one were considered, while the Seed-eye board was used. In order to collect significant information during the data collection campaigns, a custom data logging application was developed. The application, installed on the sink node, was able to send all received packets to a laptop working as a storage device. The data exchange between sink node and laptop was performed by a serial communication. By embedding in the transmitted packets the amount of time in which a node uses the radio, a measure of the power used to both listen the channel and send data was performed.

Finally, we observe that simulations and test beds were carried out by using the independent replications method and all results are characterized by a 95% confidence interval with a 5% maximum relative error.

V. RESULTS

In this section, main results obtained by both simulations and experimental campaigns are reported. The performance of the MAC scheduler in a real environment are presented separately for both test beds before discussing device limitations in implementing energy saving policies.

A. Simulation results

As previously described, several simulation campaigns performed by using the Contiki’s simulation tools (i.e., the Cooja network simulator and the MSPsim device emulator) were carried out to evaluate the behavior of the proposed scheduler in a timing-accurate and controlled environment. This environment combines a cycle-accurate simulation of the Tmote Sky platform with a bit-level accurate simulation of its CC2420 radio transceiver.

The main purpose of such simulations was to evaluate the scheduler benefits in terms of energy consumption, while considering its impact on communication delay. A performance comparison between the proposed MAC scheduler and the MAC solution implemented by the IEEE 802.15.4 standard was carried out. The latter was chosen as reference due to the lack of a different scheduling-based protocol already implemented in Contiki. Moreover, similar analysis can be found in literature [21]. More in detail, we compared the performance of our solution with the always-on MAC protocol in Contiki, based on the NULLRDC driver. As expected, the power consumption for a node using NULLRDC is high. This configuration is very similar to the CSMA-CA solution specified in the IEEE 802.15.4 standard.

The power consumption of each node was measured using Contiki Powertrace tool [20]. This metric was evaluated for both star and chain network topologies.

Table II shows the power consumption of each node in a network of five nodes organized in a chain topology. The results clearly show that the proposed scheduler reaches a

(a) (b)

Fig.5. Test bed settings: transmission power -8dBm, distance between nodes 10 m at (a) University of Salento, (b) Scuola Superiore Sant’Anna

TABLE II
CHAIN TOPOLOGY ENERGY CONSUMPTION IN mW

Node ID	Proposed Scheduler									IEEE 802.15.4
	DR=5s			DR=30s			DR=60s			
	T_0=5s	T_0=10s	T_0=15s	T_0=30s	T_0=60s	T_0=90s	T_0=60s	T_0=120s	T_0=180s	
2	7.30	4.48	3.58	2.64	2.18	2.02	2.17	1.94	1.86	61.20
3	7.29	4.49	3.58	2.64	2.18	2.02	2.17	1.94	1.86	61.20
4	7.29	4.51	3.57	2.64	2.18	2.02	2.17	1.94	1.86	61.20
5	5.42	3.57	2.95	2.33	2.02	1.91	2.02	1.86	1.81	61.20

substantial energy consumption reduction with respect to the IEEE 802.15.4 protocol. This main overall result is true for each node and configuration, since idle listening and collisions are avoided. More in detail, considering the condition of high traffic load (Fig. 6.a), for a value of T_0 equal to 5 seconds the minimum power gain is about 85%, while for T_0 equal to 15 seconds it reaches about 95% of the IEEE 802.15.4 solution. From the simulations, it can be noted that the power consumption is a decreasing function of T_0, which is the parameter that regulate the node activity. A small value of this time means a high reactivity of the node, which is required to be awake every time a transmission is going to occur. On the contrary, a high value indicates a small reactivity of each node in the network and thus a reduced number of awakenings for its neighbors. Moreover, since the idle power consumption is the dominating factor of the system power consumption, there is not a significant difference among nodes closer to the sink, which forward messages generated by others too, and nodes further away. However, the farthest node shows lower power consumption due to the different number of neighbors, which determines the number of awakening in the W_{TBL}. The last node in the chain has only one neighbor, and so it is awake for less time. Similar considerations concern also results related to the other two different packet rates considered, shown in Figg. 6.b and 6.c. In these cases, the reduction of the energy consumption of each node as a function of T_0 is lower, and therefore it appears less noticeable from a graphical point of view. This is because the energy spent by each node is always lower than the 90% with respect to the IEEE 802.15.4 protocol.

On the contrary, when the network is configured in a star topology, with all the nodes able to communicate to each other, they exhibit the same energy consumption. This result is true for each data rate. Note that, also in this case, data in Table III clearly show that the minimum power gain is around 85% and the power consumption is a decreasing function of T_0. As expected, the nodes energy consumption, when using the IEEE802.15.4 protocol, is the same for both network topologies.

Duty cycling MAC protocols impose a trade-off between transmission delay and energy saving. Lower values of energy consumption require higher transmission delays, since nodes must be in a sleep state for a longer amount of time. To quantify the latency of the proposed scheduler only the chain network previously described was adopted. In order to estimate the Round Trip Time (RTT), we measured the time needed to forward a packet from the sink (i.e., node 1) to the last node of the chain and in the reverse direction. In Fig. 7, results of the 8-hops latency test are shown. As expected, the delay introduced at each hop by the proposed scheduler is very close to half of the wakeup interval. The variations are due to the random choices of the wakeup time offsets imposed during the initialization phase. The delay is accumulated at each hop. This effect can be negative in the case of delay-sensitive applications, while it is negligible for delay-tolerant applications. Higher transmission delays are compensated by high improvements in power consumption.

Finally, it is important to observe that the chosen parameters have allowed a successfully transfer of all packets to the sink in all performed simulation campaigns.

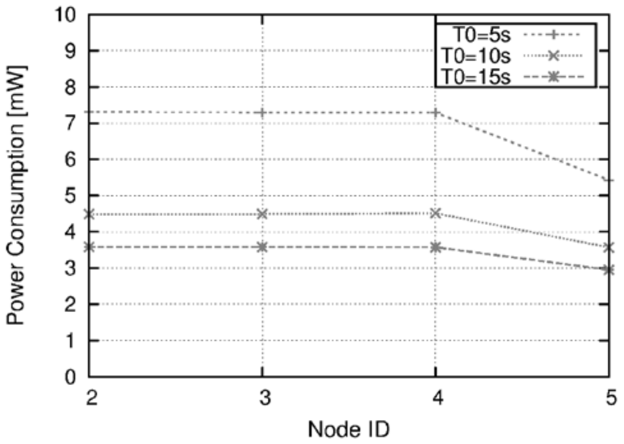
B. Test beds results

The results of the performance analysis of the MAC scheduler in a real environment are reported and discussed in this section. Results of the two test beds are first analyzed separately, and then, some considerations coming from a comparison analysis between the two boards are outlined and

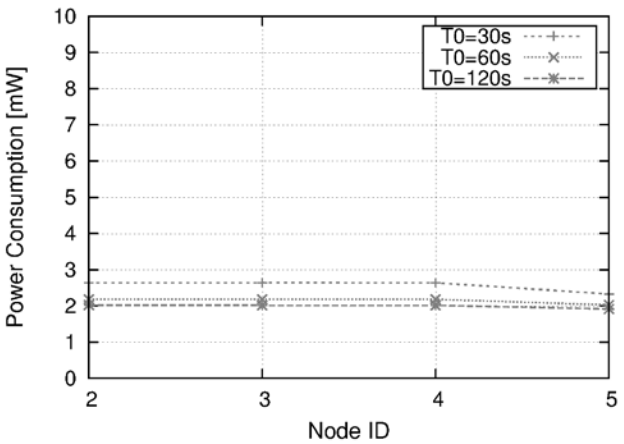
TABLE III
STAR TOPOLOGY ENERGY CONSUMPTION IN mW

Node ID	Proposed Scheduler									IEEE 802.15.4
	DR=5s			DR=30s			DR=60s			
	T_0=5s	T_0=10s	T_0=15s	T_0=30s	T_0=60s	T_0=90s	T_0=60s	T_0=120s	T_0=180s	
2	11.0	6.36	4.80	3.25	2.48	2.22	2.17	2.09	1.86	61.20
3	11.0	6.35	4.80	3.25	2.48	2.22	2.17	2.09	1.86	61.20
4	11.0	6.36	4.81	3.25	2.48	2.22	2.17	2.09	1.86	61.20
5	11.0	6.36	4.80	3.25	2.48	2.22	2.02	2.09	1.81	61.20

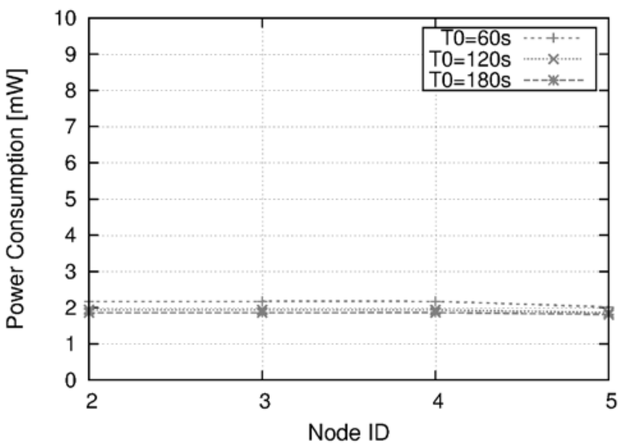
discussed.



(a)



(b)



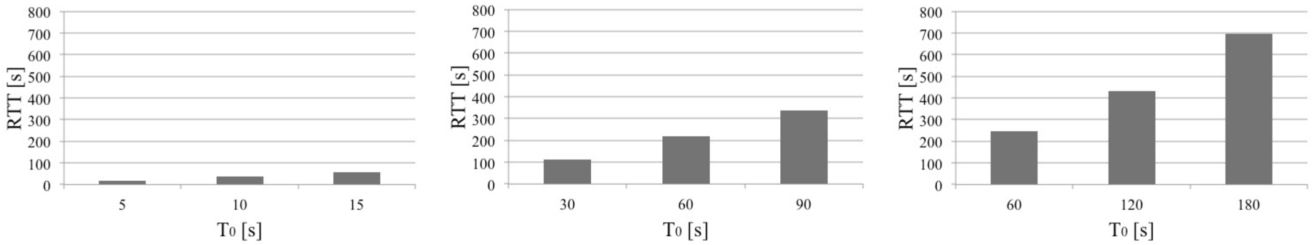
(c)

Fig.6. Chain topology energy consumption for the three different data rates considered: (a) 1 packet every 5 seconds, (b) 1 packet every 30 seconds, (c) 1 packet every 60 seconds.

In the MB851_TEST both chain and star network topologies were analyzed using five MB851 nodes sending data at the three data rates, considered in the simulation campaigns. For each data rate the middle value of T_0 was used, so as to balance power consumption and network latency. In this analysis, the energy estimation features implemented in Contiki [22] were used to experimentally quantify the energy-efficiency of the proposed scheduler. The performance results obtained by considering both topologies are reported in Table IV and in Table V. All the measured energy consumption values are expressed in mW, while the three used data rates are labeled as DR with the indication of the elapsed time between two consecutive packets. It must be stressed that all reported energy consumption values are evaluated by considering the activation periods of both radio transceiver and device microcontroller at each scheduled transmission. Indeed, in the MB851 board all components are switched in the sleep mode during idle periods.

As mentioned in the simulation results, in a chain network topology each node sends data packets towards the sink by forwarding them to nodes closer to the sink. In terms of energy consumption this means that nodes in the middle of the chain should consume more energy because they are in charge of forwarding packets from farther nodes. However, since the transmission power consumption is not a dominating factor, the described behavior can be noticed only as general trend in Table IV, where, for each data rate, the energy consumed by nodes in the middle of the chain only slightly increases for nodes closer to the sink. Moreover, Fig. 8.a highlights the reduced energy consumption of the node 5, already discussed in the simulation results. Finally, the curves clearly show that the energy consumption of each node decreases when the network load decreases. In order to better appreciate the effectiveness of the proposed scheduler, a performance comparison with the IEEE 802.15.4 MAC protocol working in CSMA-CA mode was carried out. In this case, the energy consumption of each node does not depend by its position in the chain or by the load of the network, and the energy saved by the proposed scheduler is around 81%, when considering a communication with the highest used data rate.

On the contrary, when the network is configured in the star topology with the sink in the center, each node performs a one-hop transmission, and all nodes in the network consume the same energy, as reported in Table V. Fig. 8.b shows as this behavior can be noticed for all nodes by considering each data rate. Considering the energy consumption behavior as a function of the data rate, it is possible to note how lower values can be experienced at lower data rates. Results summarized in Table V confirm that the proposed scheduler outperforms the IEEE 802.15.4 MAC protocol also using the star topology. As expected, the energy consumption values of all nodes, using IEEE 802.15.4 protocol, are the same for both topologies.



(a) (b) (c)

Fig.7. Round Trip Time varying the Data Rate: (a) 1 packet every 5 seconds, (b) 1 packet every 30 seconds, (c) 1 packet every 60 seconds.

Ultimately, let us observe that, due to channel conditions, the percentage of successfully arrived packets, when the proposed scheduler is used, is about 97% in the chain topology and 100% in the star topology.

In the Seed_TEST, the performance of the proposed MAC scheduler was analyzed by using the Seed-eye board. In such a case, the same settings for network topologies and data rates, used in the MB851_TEST, were considered. Let us observe that, unlike the MB851 board, the Seed-eye microcontroller cannot be switched in sleep mode, thus showing higher energy consumption than the first board. Therefore, to better analyze the effects of the proposed MAC scheduler, only the radio energy consumption was reported in Table VI and Table VII. It is important to observe that, for a fair comparison between the two boards, an energy consumption value equal to 280.50 mW, must be added to the values reported in both tables. This value represents the energy consumed by the Seed-eye microcontroller.

Considering the chain topology results, the general trend observed in Table IV is confirmed. Remarking the results obtained in the MB851_TEST, the radio energy consumption decreases when lower values of data rates are used. These described behaviors can be observed graphically in Fig. 9.a. By comparing experimental results obtained using the proposed scheduler, with those achieved by using the IEEE 802.15.4 standard, it is possible to observe that an energy consumption reduction of close to 56% can be reached.

Looking at results of the Seed_TEST, when nodes are organized in a star topology, as shown in Table VII and Fig. 9.b, the energy consumption trend outlined in the MB851_TEST is again confirmed. In the star topology, the proposed scheduler reaches a reduction in consumed energy close to 95% with respect to the IEEE 802.15.4.

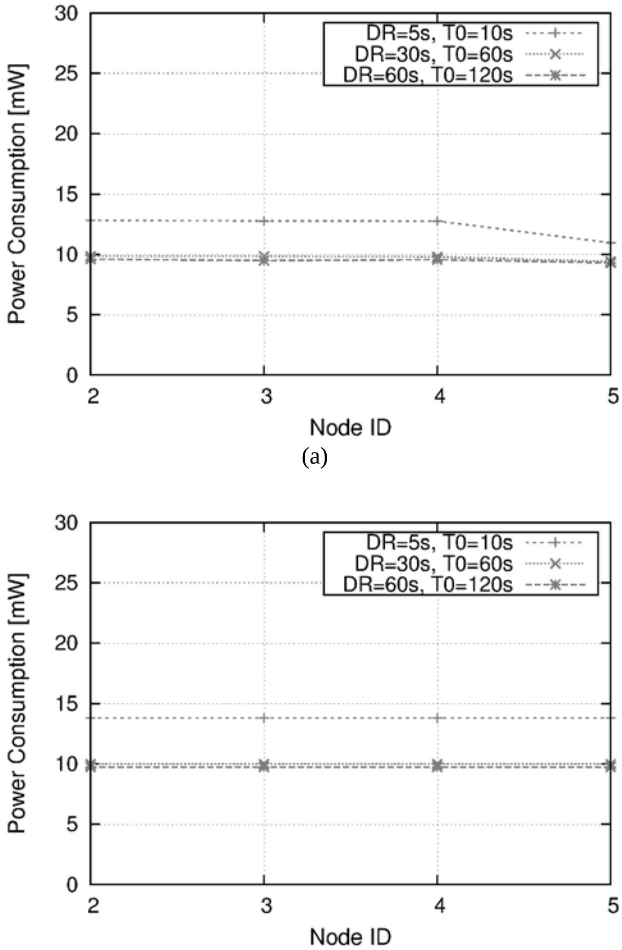
It is important to observe that, the scheduler has shown a

TABLE IV
MB851_TEST: CHAIN TOPOLOGY ENERGY CONSUMPTION IN mW

Node ID	Proposed Scheduler			IEEE802.15.4
	DR=5s T_0=10 s	DR=30s T_0=60 s	DR=60s T_0=120 s	
2	12.83	9.85	9.61	66.45
3	12.79	9.84	9.50	66.44
4	12.76	9.80	9.57	66.45
5	10.96	9.52	9.42	66.45

TABLE V
MB851_TEST: STAR TOPOLOGY ENERGY CONSUMPTION IN mW

Node ID	Proposed Scheduler			IEEE802.15.4
	DR=5s T_0=10 s	DR=30s T_0=60 s	DR=60s T_0=120 s	
2	13.78	10.00	9.75	66.44
3	13.78	10.00	9.75	66.44
4	13.78	10.00	9.75	66.45
5	13.78	10.00	9.75	66.44



(a)

(b)

Fig.8. MB851_TEST: energy consumption in the (a) chain topology, (b) star topology.

TABLE VI
SEED_TEST: CHAIN TOPOLOGY ENERGY CONSUMPTION IN mW

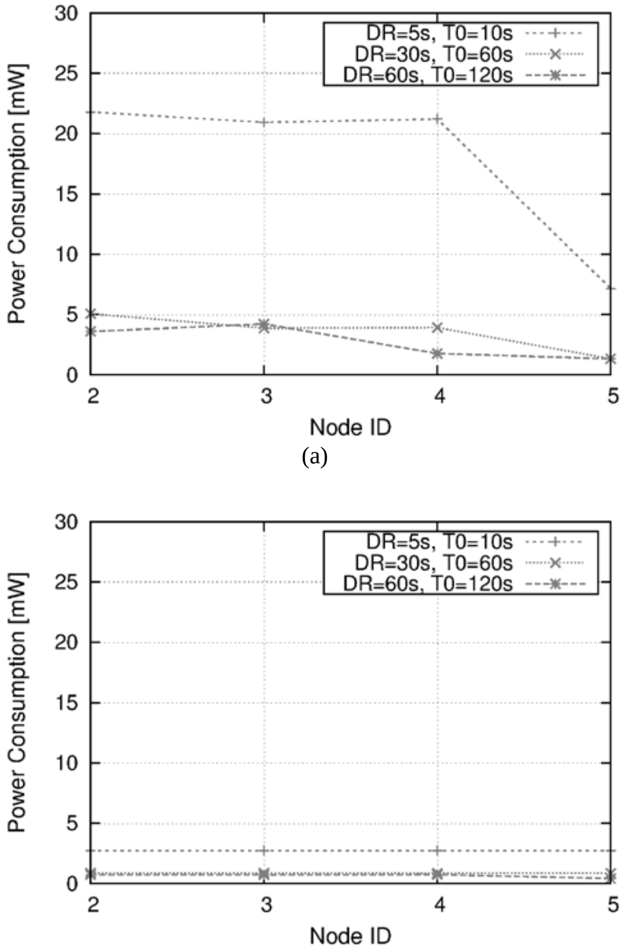
Node ID	Proposed Scheduler			IEEE802.15.4
	DR=5s T_0=10 s	DR=30s T_0=60 s	DR=60s T_0=120 s	
2	21.79	5.07	3.60	62.71
3	20.94	3.88	4.22	62.71
4	21.21	3.93	1.77	62.71
5	7.17	1.33	1.34	62.71

TABLE VII
SEED_TEST: STAR TOPOLOGY ENERGY CONSUMPTION IN mW

Node ID	Scheduler			IEEE802.15.4
	DR=5s T_0=10 s	DR=30s T_0=60 s	DR=60s T_0=120 s	
2	2.75	0.89	0.74	62.71
3	2.75	0.89	0.72	62.71
4	2.75	0.86	0.75	62.71
5	2.75	0.86	0.42	62.71

successfully transfer of all packets (i.e., 100%) to the sink in the star topology, while only the 90% of the total packets sent have reached the sink in the chain network topology. The high packet loss rate experienced in this case is due to the presence of obstacles capable of attenuate the electromagnetic waves in the considered environment.

As previously mentioned, the energy consumption values of the Seed-eye board do not take into account the microcontroller power consumption due to its inability in switching in a sleep mode. Note that adding the energy consumed by the microcontroller, the total energy consumed by each node became substantial. As example, considering node 5, chain topology, and lowest data rate, the total energy consumed by the Seed-eye is equal to 281.84 mW against the 9.42 mW required by the MB851 board. Albeit the use of an energy efficient MAC scheduler can provide great benefits in extending WSN nodes lifetime, its use only makes sense if the used hardware devices embed microcontrollers and sensors capable of supporting low power consumption policies.



(a)
(b)

Fig. 9. Seed_TEST: energy consumption in the (a) chain topology, (b) star topology

VI. CONCLUSIONS AND FUTURE WORK

The paper mainly presents an energy efficient MAC protocol targeted to IoT devices, its implementation in Contiki OS, and its validation in two real test beds. The proposed protocol is an enhanced version of an early implementation already presented in [3], and it specifically addresses the clock drift problem, a non negligible issue when using such protocols in real environments.

In the paper, the protocol implementation is detailed, before to present energy consumption performance by means of simulations and real test beds. The performance of the developed energy efficient MAC scheduler were evaluated in a real environment by deploying two test beds at the University of Salento in Lecce and Scuola Superiore Sant'Anna in Pisa, respectively. In both test beds several experimental campaigns were performed by varying the network topology and the data transmission rate. Moreover, two different boards, namely MB851 and Seed-eye, characterized by different hardware architectures, were used in order to validate the MAC protocol implementation. Overall performance results show the effectiveness of the proposed scheduler, as well as its benefits in saving nodes energy and extending network lifetime with respect to the standard IEEE 802.15.4 protocol. The evaluation of the proposed protocol considering a greater number of devices and different performance metrics, such as the RTT, will characterize future work.

REFERENCES

[1] L. Mainetti, L. Patrono, A. Vilei, "Evolution of wireless sensor networks towards the Internet of Things: A survey", 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Sept. 2011, pp. 1-6.

[2] J. Hill and D. Culler: "Mica: a wireless platform for deeply embedded networks", IEEE Micro, 22(6):12–24, November 2002.
[3] Alessandrelli D., Mainetti L., Patrono L., Pellerano G., Petracca M., Stefanizzi M.L., "Implementation and validation of an energy-efficient MAC scheduler for WSNs by a test bed approach", 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Sept. 2012.
[4] A. Dunkels, B. Gronvall, and T. Voigt: "Contiki – a lightweight and flexible operating system for tiny networked sensors", in First IEEE Workshop on Embedded Networked Sensors, Tampa, Florida, USA, November 2004.
[5] IEEE 802.15.4 Standard: http://www.ieee802.org/15/pub/TG4.html.
[6] MB851 user manual available at: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/USER_MANUAL/CD00262415.pdf
[7] Seed-eye datasheet available at: http://www.evidence.eu.com/products/seed-eye.html
[8] J. Polastre, J. Hill, and D. Culler: "Versatile low power media access for wireless sensor networks", in: Second ACM Conference on Embedded Networked Sensor Systems, pp. 95-107 (2004).
[9] C. Enz, A. El-Hoiydi, J.D. Decotignie, and V. Peiris: "WiseNET: An Ultralow-Power Wireless Sensor Network Solution", IEEE Comp., vol. 37, no. 8 (2004).
[10] M. Buettner, Ga. V. Yee, E.Anderson, and R. Han: "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", in: International Conference on Embedded Networked Sensor Systems, pp. 307–320, 2006.
[11] C.J. Merlin, and W.B. Heinzelman: "Duty Cycle Control for Low-Power-Listening MAC Protocols", IEEE Transactions on Mobile Computing, vol. 9, no.11 (2010)
[12] W. Ye, J. Heidemann, and D. Estrin: "Medium access control with coordinated adaptive sleeping for wireless sensor networks", IEEE/ACM Transactions on Networking (TON), Volume 12 Issue 3, June 2004.
[13] T. Van Dam and K. Lengendoen: "An adaptive energy efficient MAC protocol for wireless sensor networks", in: First ACM Conference on Embedded Neworked Sensor Systems, pp. 171-180 (2003).
[14] Lei Tang, Yanjun Sun, O. Gurewitz and D.B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," Proceedings of 30th IEEE Conference on Computer Communications (INFOCOM 2011), Shanghai, China, April 2011, pp. 1305-1313.
[15] W. Ye, F. Silva, and J. Heidemann: "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling", in: Fourth ACM SenSys Conference (2006)
[16] B. Jang, J.B Lim, and M.L. Sichitiu.: "AS-MAC: An asynchronous scheduled MAC protocol for wireless sensor networks", in: 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (2008).
[17] J. Hill, R. Szewczyk, A.Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in Proc. 9th Int. Conf. Architectural Support for Programming Languages and Operating Systems, Cambridge, MA, Nov. 2000, pp. 93–104.
[18] MicaZ datasheet available at: http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf
[19] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis. Collection tree protocol. In Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys), Berkeley, CA, USA, 2009.
[20] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes: "Powertrace: Network-level power profiling for low-power wireless networks", Technical Report T2011:05, Swedish Institute of Computer Science, March 2011.
[21] P. Suarez, C. Renmarker, A. Dunkels, T. Voigt, "Increasing ZigBee network lifetime with X-MAC", Proceedings of the workshop on Real-world wireless sensor networks 2008, pp. 26-30
[22] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He. "Software-based on- line energy estimation for sensor nodes", Proceedings of the Fourth IEEE Workshop on Embedded Networked Sensors (Emnets IV), Cork, Ireland, June 2007.

Daniele Alessandrelli received his M.Sc. in Computer Engineering from the University of Ancona, Italy, in 2009. Currently, he is a Ph.D. student at Scuola Superiore Sant'Anna, Pisa, Italy. His research interests are in Wireless Sensor Networks (WSNs) and the Internet of Things (IoT). Specifically his Ph.D. work focuses on middleware for WSNs and programming abstractions for the IoT. Daniele Alessandrelli is also the main author of Sniffer 15.4, an open-source IEEE 802.15.4 packet sniffer for Android devices.

Luca Mainetti is an associate professor of software engineering and computer graphics at the University of Salento. His research interests include web design methodologies, notations and tools, services oriented architectures and IoT applications, and collaborative computer graphics. He is a scientific coordinator of the GSA Lab - Graphics and Software Architectures Lab and IDA Lab - IDentification Automation Lab at the Department of Innovation Engineering, University of Salento. He is the Rector's delegate at the ICT.

Luigi Patrono received his MS in Computer Engineering from University of Lecce, Lecce, Italy, in 1999 and PhD in Innovative Materials and Technologies for Satellite Networks from ISUFI-University of Lecce, Lecce, Italy, in 2003. He is an Assistant Professor of Network Design at the University of Salento, Lecce, Italy. His research interests include RFID, EPCglobal, Internet of Things, Wireless Sensor Networks, and design and performance evaluation of protocols. He is Organizer Chair of the international Symposium on RFID Technologies and Internet of Things within the IEEE SoftCOM conference. He is author of about 60 scientific papers published on international journals and conferences and four chapters of books with international diffusion.

Giovanni Pellerano received his M.S. degree in Computer Engineering from the University of Pisa in October 2012. Currently he is collaborating with CNIT National Laboratory of Photonic Networks (LNRF) and TeCIP Institute under a research project on the Internet of Things topic and involving technologies such as IEEE802.15.4, 6LoWPAN, RPL and CoAP. Security researcher and good skilled computer programmer for passion, he is developer at Globaleaks, Tor2Web and SniffJoke.

Matteo Petracca received the M.S. degree in Telecommunication Engineering in 2003 and the Ph.D. degree in Information and System Engineering in 2007, both from the Politecnico di Torino, Turin, Italy. From March 2007 to November 2009 he was a post-doc researcher at the Politecnico di Torino working on multimedia processing and transmission over wired and wireless packet networks. In 2009 he joined the Scuola Superiore Sant'Anna in Pisa, Italy and in the 2010 the CNIT (National Inter-University Consortium for Telecommunications) as research fellow. Dr. Petracca has been actively involved in many Italian and European research projects. He is co-author of more than 30 scientific papers published in international journals, peer-reviewed conference proceedings and book chapters.

Maria Laura Stefanizzi graduated cum laude in Computer Engineering at University of Salento (Italy) in April 2012. Since January 2009 she collaborates with IDA Lab - IDentification Automation Laboratory at the Department of Innovation Engineering, University of Salento. Her activity is focused on the design and validation through test beds on real devices of innovative applications and protocols aimed to reduce power consumption in Wireless Sensor Networks. She is also involved in the study of new solutions for the integration of RFID and WSN technologies.