A novel MAC scheduler to minimize the energy consumption in a Wireless Sensor Network

Luca Anchora a, Antonio Capone b, Vincenzo Mighali b, Luigi Patrono b,⇑, Francesco Simone b

a Department of Information Engineering (DEI), University of Padova, Via Gradenigo 6/B, 35131 Padova, Italy
b Department of Innovation Engineering, University of Salento, Via per Monteroni, 73100 Lecce, Italy

ARTICLE INFO
Article history:
Received 31 January 2013
Received in revised form 7 October 2013
Accepted 5 December 2013
Available online xxxx

Keywords:
Energy saving
Duty cycle
MAC scheduler
Performance evaluation
Wireless Sensor Network

ABSTRACT

The rising success of the Internet of Things has led the Wireless Sensor Networks to play an important role in many fields, ranging from military to civilian applications. However, since sensor nodes are battery powered, communication protocols and applications for these networks must be carefully designed in order to limit the power consumption. In this work, a new MAC protocol able to significantly reduce the power consumption and compatible with the IEEE 802.15.4 standard, is designed and validated. The defined protocol is based on an efficient setting of the node’s duty cycle as a function of the transmission times of the neighbor nodes. In a duty cycle period, each node wakes up once to transmit and \( N \) times to receive, where \( N \) is the number of neighbors, while it remains in sleep mode for the rest of the time. The defined protocol has been validated through both an analytical and a simulative approach. By using the first approach, the proposed solution is compared with another energy-efficient protocol, namely AS-MAC; then, the differences between the simulated scenario and the analytical one are analyzed. By using the second approach (through Omnet++ simulator), we carried out a performance comparison between our protocol and the current MAC protocol compliant with the ZigBee standard. All the results have shown the effectiveness of the proposed solution, which has proved to be flexible and efficient, since it is able to provide high energy savings at different date rate, without a negative impact on the packets delivery.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The capability to sense key parameters from an environment is becoming more and more important in many application scenarios such as military operations, surveillance, building automation, healthcare, and logistics. This trend aims at creating smart environments able to capture, in a pervasive way, all useful information from the real world, contributing to assert the concept of the Internet of Things (IoT). The use of the wireless technology can facilitate this evolution process also leading to a growth of the Internet itself, which is no longer seen only as a tool for linking people to services, but as a means to allow the implementation of the new Machine-to-Machine (M2M) paradigm. Among all wireless technologies, WSNs are the ideal choice since their ability to self-configure and self-organize allows to overcome the typical weaknesses of infrastructured networks, in order to carry out pervasive environments such as smart city, smart farm, and smart hospital [1]. A WSN consists of a number of nodes distributed in an area of interest and equipped with some sensors (e.g., temperature, pressure, humidity). These nodes communicate each other through a radio transceiver that allows them to send the captured data to a special entity called sink. One of the main characteristics of WSNs is the use of multi-hop communication, which allows nodes farthest from the sink to reach the sink itself, but at the...
same time it requires that the network is never partitioned, i.e. a path to the sink should always be present. With regard to this aspect, it is important to note that the nodes of a WSN are battery-powered and, in most cases, such a power cannot be regenerated or replaced, therefore its exhaustion corresponds to a definitive shutdown of the node. In this situation, not only data collected by that node are lost, but also routes eventually handled by that node. Of course, a large number of node losses, due to battery depletion, could lead to network partitioning. That said, the energy consumption is, therefore, a key aspect in WSNs and it is crucial that sensor nodes optimize power consumption to extend the network lifetime in a consistent way with the real use cases.

The main operations that a WSN node can perform are data sensing, data processing and data communication. This last operation is certainly the most stressful operation from the point of view of power consumption, because it is associated with phenomena such as collision, overhearing (i.e. listening of messages addressed to another node), over-emitting (i.e. transmission of data to a node that cannot receive them) and idle listening (i.e. listening to the channel in absence of communications). For these reasons, many works in literature are focused on energy saving exploiting enhancements at various layers of the protocol stack of sensor nodes [2–4], with particular interest on the MAC layer. In fact, since the MAC layer is responsible for managing channel access control mechanisms, it provides directly access to the functionality of the radio transmitter and allows to modify its state without any modifications to the standard. The opportunity, for example, to exploit the existing MAC control packet to interact with the radio transmitter enables new protocols to optimize the power consumption while maintaining compatibility with the IEEE 802.15.4 standard. Then, in this field, a very interesting approach is based on an appropriate tuning of the node duty cycle, through which nodes switch between ON and OFF state according to a predefined scheduling. Such an approach must deal with synchronization problems among nodes, high overhead, hidden node especially in the initial setup phase, etc.

In this work, an asynchronous scheduler is defined that significantly reduces the power consumption of WSN nodes using an approach based on duty-cycle. Thanks to the defined solution, each node is able to know in advance the time instants in which its neighbors transmit. By exploiting this information, the local scheduler of each node regulates the activation and deactivation of the radio transmitter so that this transmitter is active only when nodes actually send or receive data. This approach consistently reduces power consumption connected with the phenomenon of idle listening and thus it causes an extension of network lifetime. To make the proposed solution as consistent as possible to the real nodes behavior, also network topology changes have been suitably managed, since the information stored in each node is updated as soon as the fall of a neighbor node or the presence of a new one are detected. In addition, the desynchronization problem caused by clock drift is adequately addressed. The effectiveness of the proposed solution has been evaluated through both an analytical and a simulative approach. In particular, from the mathematical point of view, we compared our solution with another energy-efficient protocol, namely AS-MAC. The results of this comparison showed that our approach has a better flexibility in addressing different application requirements. Then, in order to evaluate the accuracy of the mathematical model, we analyzed the differences between the simulated scenario and the analytical one. For the simulative approach, we used OMNET++ simulator [5], which is considered one of the best choice for evaluating protocols for WSNs in a simulated environment [6]. By using this tool, we carried out a performance comparison between our protocol and the current MAC protocol compliant with the ZigBee standard. Simulation results are very encouraging since the proposed solution allows to achieve a significant energy savings in a chain topology and, therefore, a considerable extension of network lifetime. Finally, the drawbacks of the defined approach in terms of packet delivery ratio are very negligible especially considering that WSNs are usually exploited in low traffic scenarios.

The paper is organized as follows. Section 2 summarizes the state of the art related to the minimization of power consumption in a WSN. The defined scheduler is described in Section 3. A detailed description of the mathematical model consistent with the proposed protocol is reported in Section 4. In Section 5, a simple mathematical model for AS-MAC protocol is presented. The simulation model used in Omnet++ simulator is given in Section 6, whereas in Section 7 the results are discussed. Conclusions are drawn in Section 8.

2. Related works

Among solutions designed to minimize the energy consumption in a WSN, this section tries to summarize the main enhancements focused at MAC layer and that exploit the duty cycle mechanism. In [7], authors present an extensive and exhaustive taxonomy concerning energy conservation in WSNs. In this work, approaches based on duty cycling are mainly focused on networking subsystem and ground on the fact that the radio device could be turned off when node does not have to communicate. Based on this observation two different and complementary approaches are described, namely topology control and power management.

Topology control is a power saving technique for a densely populated wireless networks that reduces the number of nodes participating in forwarding and routing packets generated by the other nodes without diminishing network connectivity and coverage. This technique is particularly useful in Wireless Sensor Networks because they are typically deployed densely, thus offering much redundancy in network coverage and connectivity. Through this approach, unnecessary nodes for network connectivity could turn asleep, saving energy. Generally, this kind of approach increases lifetime by a factor of 2 or 3 compared to networks in which all nodes are always on. On the contrary, power management technique schedules radio switching off when no activity is detected on the channel, so that nodes alternate sleep and wakeup time intervals.
Regarding power management-based protocols, it is possible to define a particular classification different from [7] and more useful for the work presented in this paper. In detail, power management-based protocols fit into three main categories: *preamble-sampling, scheduling* and *hybrid* approaches. A diagram summarizing the main protocols in the literature that belong to these categories is shown in Fig. 1.

Preamble-sampling MAC protocols \[8,9\] exploit the technique of Low Power Listening (LPL) \[10\] in order to sample the preamble of the packets. When there are no packets to be exchanged, LPL minimizes the duty cycle, but it needs a preamble longer than the wake up period to assure that the receiver can detect the channel activity. Even if BMAC \[8\] uses unsynchronized duty cycling, a preamble longer than a sleep period is exploited to wake up receivers. For this reason, this protocol introduces latency at each hop, so suffering from excessive energy consumption because of non-intended receivers. Through WiseMAC \[9\], that combines non-persistent CSMA with preamble sampling to mitigate idle listening, node schedulers are independent and information about the next awake period is piggybacked in the data acknowledgment frame. This value is used to dynamically determine the next awake time, allowing node to use short preamble. X-MAC \[11\] is a particular preamble-sampling protocol that sends a strobed preamble, i.e. a sequence of short preambles each of which is followed by a pause. A preamble packet contains the address of the target node, so that the destination node can recognize its own IP address as soon as it receives the preamble, and can immediately reply with an ACK (early acknowledgment) during the short free interval. When the sender receives an ACK packet, it stops sending preamble. A recent enhancement of X-MAC is XY-MAC \[12\], which proposes a technique, called early termination, to sharpen the size of X-MAC pauses. With sharpened pauses, the sampling period at receivers can be effectively reduced to acquire a high level of energy efficiency, especially under low traffic loads.

MAC protocols based on scheduling approach \[13–16\] manage duty cycling by periodic synchronization messages and packet transmissions. Such control messages can cause high overhead and consume significant energy even when there is no traffic. According to the first duty cycle MAC protocol for WSNs, S-MAC \[13\], all nodes in a neighborhood wake up simultaneously and listen to the channel. They remain awake during the entire awake period even if they are neither sending nor receiving data, so leading to high latency and low throughput. T-MAC \[14\] tries to reduce the long wake up time of S-MAC by shortening the awake period if the channel is idle. Anyway, the wake up time is much longer than the LPL. DMAC \[15\] uses a data gathering tree structure to achieve both energy efficiency and low packet delivery latency, and it assumes that nodes fit the duty cycles according to the traffic load in the network. RMAC \[16\] exploits cross-layer routing information in order to avoid latency. In particular, a setup control frame is used to schedule the upcoming data packet delivery along a specific route so that an upstream node can send the data packet to intermediate relay nodes that can immediately forward it to the downstream node.

Recent works on scheduling-based protocols are \[17–19\]. In particular, \[17,18\] present a TDMA-based approach. In \[17\], authors propose a MAC protocol, called LASA, in which time slot is not static but has a dimension that varies dynamically depending on the traffic load in the network. Exploiting this feature, LASA protocol eliminates idle times in each slot, so removing the main cause of energy consumption. In order to achieve this result, network is first organized into clusters and in each cluster a special node, called cluster head, is elected. This smart node is responsible for assigning time slots to all nodes in its own cluster. In \[18\], authors propose eL-MAC protocol, that reduces idle listening, overhearing and hidden terminal problems, and in addition decreases the power consumption by allowing to select different duty cycle depending on the current data rate. The scenario, considered in \[18\], involves the use of beacon messages sent by each node at the beginning of its own time slot. Each node uses the beacon message to notify the destination node about the presence of data to be received, allowing nodes to switch to the sleep mode if there are no transmissions. Finally, in \[19\], a cross-layer approach is represented by CL-MAC, that allows to obtain significant energy savings by using network-layer information. In particular, this protocol seeks to maintain the nodes in a sleep state as long as possible, so that only the nodes in the routing path of the current transmission wake up to receive and forward data.

Fig. 1. MAC protocols based on duty cycle approach.
To achieve this result, CL-MAC protocol slightly modifies RTS and CTS packets, adding to them the information whether a node is included or not in the routing path of the current transmission. A separate mention should be given to [20], in which authors propose a duty cycle learning algorithm (DCLA) that adapts the duty cycle during run time, without the need of human intervention, in order to minimize power consumption. It runs on coordinator devices and collects network statistics during each active duration to estimate the incoming traffic.

Hybrid solutions [21,22], match some features of preamble sampling with scheduling techniques. In [21], SCP-MAC reduces the length of the preamble and minimizes the cyclic wake up time of an LPL system by exploiting the synchronization of the wake up time of neighboring nodes. It decreases energy consumption, especially at very low duty cycles, but it is not able to avoid the overhearing problem. AS-MAC [22] is used to coordinate asynchronously the wake up times of neighboring nodes to reduce overhearing, contention and delay. Furthermore, it exploits LPL to minimize the periodic wake up time. Since nodes store the wake up schedules of their neighbors, they know when these turn active. One of the main disadvantages of the asynchronous wake up interval is the inefficiency in broadcasting since AS-MAC has to transmit every packet once per each neighbor.

The previous description highlights how the different protocols defined in the literature still suffer from various problems. Regarding preamble sampling approaches, even though they give the sender guarantee that the receiver stays in the awake state for the reception, an excessive energy consumption occurs both at sender and non-intended receivers. Scheduling-based protocols, instead, exhibit various drawbacks depending on the synchronization mechanism used. The synchronous scheduling, for example, leads to an unnecessary consumption of energy at non-intended receivers, whereas the dependence on particular network topologies (e.g. tree structure, clustering, etc.) or on network layer information leads to excessively constrained and therefore less flexible protocols. Finally, also hybrid-based protocols reveal some problems. In particular, SCP-MAC protocol does not prevent energy loss because of overhearing and, due to its synchronization procedure, it results in increased contention and delay. Whereas, AS-MAC protocol, as stated before, is inefficient in broadcasting. It is important to note that the current MAC layer for WSNs is compliant with IEEE 802.15.4 standard [23], which refers to both MAC and PHY layers. On the top of such standard, the most widely used technology for WSNs is Zigbee [24], since it is able to pander to the needs of low power, low cost and low maintenance of this kind of networks. As stated in [25], CSMA/CA schema of IEEE 802.15.4 produces a high number of retransmissions causing a large nodes energy consumption, so the use of innovative schedulers for a better management of the sleep periods could lead to save energy. The Zigbee specification allows the introduction in the Zigbee protocol stack of the Low Power Routing feature enabling a multi-hop mesh networking without requiring router nodes free from energy constraints.

In this context, the coordination of the wake up times of neighboring nodes and the opportunity to enable networks to self-configure are relevant topics to be addressed, as proposed by the solution hereafter introduced.

3. The asynchronous MAC scheduler: AS2-MAC

3.1. Overall description of the proposed solution

The main idea of the defined scheduler, called AS2-MAC, is the concept of smart awake. In any duty cycle period, a node wakes up to both send and receive, but awakenings for reception are scheduled at the transmission times of the neighboring nodes. In particular, thanks to the information gathered during the initialization phase, each node enables its radio transceiver only if it has to transmit or receive data from its neighbors. In this way, the scheduler can reduce unnecessary awakenings, to which both idle listening and overhearing phenomena are related. Also IEEE 802.15.4 was recently amended by the Task Group 4e in order to add functionality for the reduction of power consumption [26]. For consistency with the aforementioned amended standard, we can say that our solution exploits the so called coordinated sampled listening (CSL): a low-energy mode of media access control specifying how receiving devices periodically monitor the channel(s) for incoming transmissions at low duty cycles.

In order to simplify the communication among nodes and to reduce the interference due to simultaneous transmissions, data transmission and data reception phases work separately. So, when the radio is ON, a node either transmits data and receives acknowledgment messages (ACKs), or receives data from other nodes and sends ACKs.

For the sake of clearness, before describing the new scheduler in detail, some parameters used in the discussion are introduced below:

- $T_0$, shown in Fig. 2, is the time interval (in seconds) between two subsequent transmissions. It is the same for every node and its value is preconfigured.
- $Wake Time$, shown in Fig. 2, is the time interval (in seconds) during which a node can transmit the local buffered data or receive data from its neighbors. Recalling the IEEE 802.15.4e standard, this parameter matches the so called coordinated sampled listening period, that is, the time interval in which receiving devices monitor the channel(s) for incoming transmissions.
- $Sleep Time$, shown in Fig. 2, is the time interval (in seconds) during which a node can turn off the radio.
- $Announce Packet (Pkt_{ANN})$ is a signaling packet used by a node joining the network. It is exploited to announce the presence of the node and its next awakening time for transmission.
- $Wake-up Table (WTBL)$ is a data structure in which a node stores the awakening time for transmission of its neighbors. In particular, there is a table entry for each neighbor containing the following information: (a) the neighbor ID, (b) the neighbor offset, $O_n$, defined as the time interval elapsed since the awakening of the first neighbor.
neighbor stored in the table, and (c) the number of cycles of length $T_0$ during which no data have been received from the corresponding node.

3.2. Initialization phase

In the initial network setup, all nodes exchange information about their transmission time by sending a PktANN. On the reception of such a message from an unknown neighbor, the AS2-MAC updates its $W_{TBL}$ by storing a new entry. Before being stored, the information on the transmission time of the neighbor is converted into offset, in order to indicate the time interval (in seconds) that elapses from the beginning of the duty cycle period to the instant of neighbor transmission. The entries in the table are in increasing order of the offset.

Each node chooses its own transmission time as a random value in a proper interval, also taking into account the choice done by its neighbors. This separation in time among transmissions of neighboring nodes leads to a reduced channel access contention. More in detail, if the $W_{TBL}$ is empty, then the transmission time is randomly selected in the interval

$$[T_C, T_C + T_0 - (WakeTime + 2 \cdot TurnAroundTime)]$$

where $T_C$ is the current time, $WakeTime$ is the time window dedicated to data transmission, and $TurnAroundTime$ is the amount of time the radio needs for changing its state. If the $W_{TBL}$ is not empty, then the node tries to set its own transmission time to a value different from those of its neighbors in order to avoid collisions due to simultaneous transmissions. The value is chosen so that the $WakeTime$ of the node does not overlap with that of any other neighbor. In particular, the node checks if there are two consecutive entries in the table, namely $i$ and $i + 1$, whose offsets difference is greater than

$$2 \cdot WakeTime + 4 \cdot TurnAroundTime$$

If so, the transmission time is chosen within the interval

$$[offset[i] + D, offset[i + 1] - D]$$

where $D = WakeTime + 2 \cdot TurnAroundTime$, while $offset[i]$ and $offset[i + 1]$ are the offsets associated to entries $i$ and $i + 1$ respectively. If there is only one entry in the $W_{TBL}$, the transmission time is chosen within the interval

$$[offset[0], T_0 - D]$$

3.3. Steady state

After the warm-up phase, during which all nodes determine their transmission time and build their $W_{TBL}$, the network starts working regularly. The activity of each node is managed by MAC scheduler module using information included in the $W_{TBL}$. The most important events in this phase are: the periodic wake up for a transmission, the scheduled wake up for the reception of packets from a neighbor, the entry of a new node in the network.

In Fig. 3, the sequence diagram that indicates the interactions among the scheduler, the MAC layer and the physical layer during the data transmission is shown. When the scheduled time for the transmission occurs, the proposed scheduler checks if the MAC is IDLE. This operation is necessary to avoid inconsistent states that the MAC is not able to manage. As soon as the MAC is IDLE, it notifies its status to the scheduler, which enables MAC for data transmission and turns on the radio transceiver. Afterwards, the MAC protocol works in the standard way, whereas the scheduler waits the end of the transmission time interval. When the wake time expires, the scheduler disables MAC for data transmission and turns off the radio transceiver. A similar sequence of activities occurs when the node wakes up to receive data from a neighbor in accordance with its $W_{TBL}$, but with both the MAC layer and the radio transceiver enabled in reception rather than in transmission.

Analyzing more in detail how the protocol works, it can be stated that the macro-states among which the radio transceiver switches are substantially five: IDLE (listing the channel), CCA (channel contention), TX, RX and OFF. The first four states are those visited also by MAC layer during a duty cycle period, whereas the last one is the state leading to the actual power saving results. Fig. 4 shows the state diagram of the PHY layer and highlights the main actions that trigger a state change.

As already mentioned, these states are visited in order to manage two type of events: (i) the transmission of a buffered packet (every $T_0$ seconds) and (ii) the reception of packets from a known neighbor. Assuming that at some point PHY layer is in OFF state, when the transmission timer expires, the node schedules the next transmission after $T_0$ seconds, wakes up (MAC and PHY in IDLE state), checks the presence of packets to be transmitted in its queues and, if any, starts the contention for the channel access (MAC and PHY in CCA state). If the contention is won, then the transmission is initiated (MAC and PHY in TX state). As the packet transmission ends, the node switches to the IDLE state and waits for an ACK from the intended receiver. If the acknowledgment is not received before the expiration of a specific timer managed by MAC layer, then the message must be sent again. On the reception of the ACK, the node switches to the RX state and receives the packet; then it goes back to IDLE state and checks if there are other packets to be sent. If so, the procedure is repeated, otherwise the node keeps the IDLE state until the next transmission or the end of
the Wake Time. When the Wake Time expires the node switches its radio transceiver OFF.

The data reception is, instead, managed by checking the WTBL: when a transmission event from a neighbor occurs, the node switches its radio ON and starts listening to the channel (MAC and PHY in IDLE state). If the neighbor actually transmits data, the nodes goes in RX state (MAC and PHY in RX state) and, when the reception is completed, it goes in IDLE state and then sends an ACK (switching the MAC and PHY to TX state). Finally, the node goes back to the IDLE state waiting for further packets from its neighbor. This procedure is repeated until the expiration of Wake Time, after which the node switches the radio component OFF. If nothing is received from that neighbor, the node updates the counter
in the corresponding $W_{TBL}$ entry in order to keep track of the number of times that no packets have been received from that sender. After a predefined number $m, m > 0$, of consecutive failed receptions, the entry is removed so as to avoid useless awakenings. In this way, if a node fails then all its neighbors will remove the corresponding entry from their tables after a limited amount of time. Because of this mechanism, a node may delete all the entries of its $W_{TBL}$. If this condition happens, the neighbor discovery process is started again to get an up to date view of the neighborhood.

Analyzing the discovery process in detail, when a new node joins the network, it first listens to the channel for an amount of time slightly larger than $T_0$, 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 $W_{TBL}$. After this listening phase, the node announces its presence by sending a $PktANN$, whose payload contains the transmission time selected according to the procedure described in Section 3.2. Since the new node does not still have a dedicated transmission time, it has to start a contention with the other neighbors for the channel access in order to broadcast the announce packet. Finally, it is important to note that an inaccurate synchronization may occur among the various node timers, potentially leading to idle listening and over-emitting phenomena, in addition to possible data losses. In order to address this issue, a receiver anticipates the wake up for reception by a quantity of $I_G = 2 \cdot C_{DRIFT}$ with respect to the scheduled time, where $C_{DRIFT}$ is the maximum clock drift that may occur (Fig. 5).

Anyway, in order to avoid the occurrence of the above mentioned phenomena due to a poorly managed desynchronization, a node:

- after a predefined number $p, p > 0$, of consecutive transmissions without ACK, executes a procedure similar to the network joining exploiting the $W_{TBL}$ already built or, if the table is empty, doing again the neighbor discovery and the announcement procedure;
- performs only the neighbor discovery process if its $W_{TBL}$ is empty but it is still receiving some ACKs.

4. AS2-MAC: mathematical analysis

In this section, a mathematical model of the system is defined in order to obtain a quantitative evaluation of the main performance. Some assumptions are necessary to simplify the analytical formulation and the treatment; nonetheless, it is still important to determine a possible limit on the performances.

For the sake of clarity, we introduce the following notation:

- $T_0$ = duty cycle period;
- $W_t$ = duration of Wake Time;
- $TAT$ = Turn Around Time, the amount of time needed for switching the transceiver;
- $R$ = application packet rate (expressed in packets per second - pkts/s);
- $\alpha$ = packet error rate (PER);
- $B$ = bandwidth;
- $L_{pkt}$ = packet length at PHY layer (expressed in bits);
- $L_{ack}$ = ACK length (expressed in bits);
- $AckWait$ = duration of wait timer for ACK;
- $P_{air}$ = $R \cdot T_0$. It is the traffic generated by the application layer during a duty cycle period.

The first consideration refers to the maximum ($N_{max}$) number of neighbors that a node can have (and thus to the average network density). Since the transmission times of the nodes in the same neighborhood must differ for at least $W_t + 2 \cdot TAT$ seconds, then we have that

$$N_{max} = \frac{T_0}{W_t + 2 \cdot TAT}$$

In the worst case, the number of nodes that can be fit is $N_{min} = \lfloor N_{max}/2 \rfloor$. This corresponds to the case when the transmission times of the nodes differs for a value equal
to \((W_t + 2 \cdot TAT) - \epsilon\), with \(\epsilon\) arbitrarily small and positive. Therefore, if \(N < R_{\text{max}}\), then \(T_0\) is large enough to fit all the neighbors; on the contrary, if \(N > R_{\text{max}}\) then certainly some nodes will not be able to select their transmission time. Regarding the case \(N_{\text{min}} < N < R_{\text{max}}\) it depends on how the transmission time is chosen by each node, i.e. on the average offset between consecutive transmission times.

For the analysis of maximum packet rate, Packet Delivery Ratio (PDR) and Power Consumption, the network is modeled as a network of queues in sequence, as illustrated in Fig. 6. Each queue is a D/D/1/K + 1 system, where \(K\) is the maximum buffer dimension, which is supposed to be the same for each node. The network topology chosen for the mathematical analysis is the chain, but this analysis can be naturally extended to more complex topology (e.g. the grid). We assume that in our chain topology each node can transmit only to its next neighbor along the path towards the sink, as shown in Fig. 7.

Call \(\mu_{\text{plk}} = \frac{W_t}{\mu}\) the transmission time for each packet (all the packets are supposed to have the same length) and \(\mu_{\text{ack}} = \frac{1}{\mu}\) the transmission time for an ACK. Denote with \(V\) the maximum amount of packets that a node can transmit during a wake interval, i.e.

\[
V = \frac{W_t}{\mu_{\text{plk}}} + \frac{\text{AckWait}}{\mu_{\text{ack}}} + P_1 \cdot \mu_{\text{ack}} + 2 \cdot TAT \tag{6}
\]

where \(P_1\) is the probability of correct reception of a packet from a node to its next hop. In this expression, the random backoff time that the node waits before starting a transmission, according to the CSMA-CA protocol, was neglected. For this reason, the maximum number of packets that can be actually sent is slightly smaller.

Since a node can only be active for transmission during \(W_t\) seconds over \(T_0\), \(V\) represents as well the maximum quantity that a node can send during the whole \(T_0\) interval.

4.1 Maximum packet rate

In this section, we analyze the maximum data rate \(R_{\text{max}}\) that can be afforded by the whole network without causing an infinite growth of the transmission buffer. In each duty cycle period \(T_0\), a node receives \(P_{\text{arr}}\) packets from its own application layer and, except for the node 1, a certain number of packets from the preceding node. Recalling that \(\alpha\) is the PER, since a node retransmits a packet up to three times, the probability that a packet arrives at the next hop is:

\[
P_1 = (1 - \alpha) + \alpha \cdot (1 - \alpha) + \alpha^2 \cdot (1 - \alpha) \tag{7}
\]

More in detail, indicating with \(P_{\text{out}}\) the amount of packets transmitted from node \(j\) in a period, then the total number of packets received from node \(i\) in a period is

\[
P_{\text{in}}^i = \begin{cases} P_{\text{arr}} & i = 1 \\ P_{\text{arr}} + P_{\text{out}}^{i-1} & i = 2, 3, \ldots, N \end{cases} \tag{8}
\]

Note that, since, as already mentioned, a node can retransmit a packet up to three times due to channel errors (according to standard 802.15.4), the average number of transmissions for each packet is:

\[
M = 1 \cdot (1 - \alpha) + 2 \cdot \alpha \cdot (1 - \alpha) + 3 \cdot \alpha^2 \cdot (1 - \alpha) \tag{9}
\]

To ensure the stability of the system, i.e. to prevent the infinite growth of the transmission buffer, the following condition must be true:

\[
M \cdot P_{\text{in}}^i \leq V, \quad \forall i \tag{10}
\]

Let \(R_{\text{max}}\) be the maximum packet rate for the \(i\)th node so that the system is stable. Related to node 1 we have that:

\[
M \cdot P_{\text{in}}^1 = V \Rightarrow M \cdot R \cdot T_0 = V \Rightarrow R = \frac{V}{T_0 \cdot M} \Rightarrow R_{\text{max}} = R \tag{11}
\]

Assuming that the limitation on the packet rate for the stability of the node 1 is valid, for node 2 we have that:

\[
P_{\text{in}}^2 = P_{\text{arr}} + P_1 \cdot P_{\text{out}}^1 \quad P_{\text{arr}} = R \cdot T_0 \quad P_{\text{out}}^1 = P_{\text{in}}^1 \quad R = R \cdot T_0 \tag{12}
\]

\[
M \cdot P_{\text{in}}^2 = V \Rightarrow M \cdot R \cdot T_0 \cdot (1 + P_1) \leq V \Rightarrow R \leq \frac{1}{M \cdot (1 + P_1) \cdot T_0} \Rightarrow R_{\text{max}} = \frac{1}{(1 + P_1) \cdot M} \tag{13}
\]

Generalizing, for the \(i\)th node, it follows that:

\[
P_{\text{in}}^i = P_{\text{arr}} + P_1 \cdot P_{\text{out}}^{i-1} = \sum_{j=0}^{i-1} P_1^j \cdot R \cdot T_0 = \frac{1 - P_1^i}{1 - P_1} \cdot R \cdot T_0 \tag{14}
\]
Finally, the mean value of PDR, denoted with \( PDR_{av} \), is:

\[
PDR_{av} = \frac{1}{N} \sum_{i=1}^{N} PDR_i
\]

(21)

Now, the minimum size of the transmit buffer able to prevent packet loss due to buffer overflow is calculated. For this purpose, it is necessary to determine the number of packets that a node receives before waking up for transmission. The node 1 receives at most \((T_0 - W_1) \cdot R\) packets, whereas the node \(i\) receives a number of packets equal to:

\[
(T_0 - W_i) \cdot R + \sum_{k=1}^{i-1} R \cdot T_0 \cdot P_{1}^{i-k}
\]

(22)

Therefore, in order to ensure that no packets are lost due to buffer overflow, the queue size for the \(i\)th node, \(K_i\), should be at least:

\[
K_{min} = (T_0 - W_i) \cdot R + \sum_{k=1}^{i-1} R \cdot T_0 \cdot P_{1}^{i-k}
\]

(23)

If \(K_i < K_{min}\), then \(K_i\) packets will be lost due to buffer overflow. Assuming that all nodes have the same size of the transmission queue, then:

\[
K_{min} = \max(K_{1}^{min}, K_{2}^{min}, \ldots, K_{N}^{min}) = K_N^{min}
\]

(24)

Through this mathematical analysis, it is possible to correctly sizing the transmission buffer in the simulation scenario.

### 4.3. Power consumption

In order to calculate the power consumption, it is necessary to determine the amount of time that a node spends in reception, transmission and idle states during a duty cycle period. Then, each time interval must be multiplied by the corresponding power consumption value (per unit of time) reported in the datasheet of the specific node model. Adding the obtained values and dividing the result by the duration of the duty cycle period \(T_0\), the instantaneous power consumption of each node is found.

More specifically, let \(PW_{TX}\), \(PW_{RX}\) and \(PW_{IDLE}\) be the power consumption (per unit of time) in transmission, reception and idle state respectively. In the defined chain topology, a node wakes up once to transmit and, at most, twice to receive in each period. In particular, a node wakes up in reception for each of its neighbors, that is, both for its successor and its predecessor. Therefore, node 1 and node \(N\) wake up in reception only once because they have only one neighbor (the sink does not fall in this consideration), whereas remaining nodes wake up twice in reception because they have two neighbors. The amount of time for both the transmission and the reception phase is equal to \(W_i\). Let assume that the system is stable and that the queue has the minimum size to avoid the buffer overflow. Recalling that \(M\) is the average number of transmissions for each packet, the average number of transmission attempts in a period by a node is equal to:

\[
N_{TXi} = \left( \frac{R \cdot T_0 + \sum_{k=1}^{i-1} TX_{k-i}}{T_0} \right) \cdot M, \quad i = 1, \ldots, N
\]

(25)
where the first term represents the packets received from the application layer, whereas the second ones those received from preceding nodes. Indicating as usual with $\mu_{\text{pkt}}$ the time required to send a packet, in each transmission period a node remains in transmission state for an amount of seconds equal to:

$$TX\_TIME_i = N\_TX_i \cdot \mu_{\text{pkt}}$$  \hspace{1cm} (26)

With regard to the reception phase, a node activates the radio in RX mode even if the packet is discarded due to a channel error or due to wrong destination, then all potential transmission attempts of neighboring nodes must be considered to calculate the power consumption during the reception. Therefore, the number of receptions carried out by the $i$th node, on average, is:

$$IDLE\_TIME_i = \begin{cases} 
2 \cdot W_i - (TX\_TIME_i + RX\_TIME_i + ACK\_TIME_{\text{RX}}^i + ACK\_TIME_{\text{TX}}^i) & i = 1, N \\
3 \cdot W_i - (TX\_TIME_i + RX\_TIME_i + ACK\_TIME_{\text{RX}}^i + ACK\_TIME_{\text{TX}}^i) & i = 2, \ldots, N - 1
\end{cases}$$  \hspace{1cm} (33)

Therefore, the average power consumption for the nodes in the network is given by:

$$PW_{\text{TOT}}^i = \frac{(TX\_TIME_i + ACK\_TIME_{\text{TX}}^i) \cdot PW_{\text{TX}} + (RX\_TIME_i + ACK\_TIME_{\text{RX}}^i) \cdot PW_{\text{RX}} + IDLE\_TIME_i \cdot PW_{\text{IDLE}}}{T_0}$$  \hspace{1cm} (34)

$$PW_{\text{TOT}}^i = \frac{1}{N} \sum_{i=1}^{N} PW_{\text{TOT}}^i$$  \hspace{1cm} (35)

5. AS-MAC: mathematical analysis

Although it is out of the aim of this work, the mathematical analysis of the AS-MAC is functional to the evaluation of our protocol. Indeed, by comparing the two analytical models it is possible to appreciate the advantages of the proposed solution. Therefore, in this section, a simplified model of AS-MAC is presented. The in depth description of AS-MAC is given in [22].

It is important to point out that, even though similar, AS-MAC has some fundamental differences from AS2-MAC. First of all, while in the former a node wakes up periodically to receive packets from the neighbors and periodically only to transmit, in the latter the situation is the opposite. Second, every time a node wakes up, all the nodes that have a packet for it start a contention phase.
and this may result in collisions and deferred transmissions. Third, when a sender wins the contention for the channel access then it can transmit only one packet to the receiver.

To simplify the analysis and make the results comparable with those obtained in Section 4, the following assumptions are done:

- \( I_{\text{wakeup}}(i) = I_{\text{hello}}(i) = T_0, \ i = 1, 2, \ldots, N \); i.e., every time a node wakes up for reception it first transmits a HELLO packet;
- \( \alpha = 0 \), since in the original paper the authors do not describe the behavior of a node in case of packet loss.

Therefore, packets can get lost only because of queue overflow discards.

Where possible, the notation used is the same of Section 4; all the other parameters are taken from [22]. The chain network topology is considered in this case as well, thus the queuing model in Fig. 6 is applied hereafter. This has a fundamental implication: each node but the edge ones can receive packets only from the previous and send only to the following one, thus no collisions can happen for the channel access.

5.1. Maximum packet rate

Each node has a transmission opportunity every \( T_0 \) seconds and can transmit only one packet toward its (unique) destination. Moreover, each node has to transmit not only the packets generated by itself but also those coming from the previous ones in the chain, with a burden increasing while getting closer to the sink. The whole network can be seen as a single system with an incoming rate of \( N \cdot R \) packets per second (this is true because each node generates packets independently from the others). Considering that each node can have an output rate of 1 packet every \( T_0 \) seconds, then in order to guarantee the stability of the system it must hold

\[
NR \leq \frac{1}{T_0}.
\]

Therefore, the maximum packet rate is

\[
R_{\text{MAX}} = \frac{1}{NT_0} \tag{36}
\]

which is less than the value obtained in (16) with \( \alpha = 0 \) \( (R_{\text{MAX}} = \frac{C}{NT_0}) \).

5.2. Packet delivery ratio

If \( R \leq R_{\text{MAX}} \) then the PDR is the same as derived for AS2-MAC, since in the chain topology no packet collision can happen.

5.3. Power consumption

The energy consumption formula for each node exploited in [22] is

\[
E_{\text{TOT}} = E_{\text{tx}} + E_{\text{rx}} + E_4 + E_{\text{pl}} + E_{\text{tx}}
\]

where \( E_{\text{tx}} \) is the total energy spent by a node for transmitting packets; \( E_{\text{rx}} \) is the total energy spent by a node for receiving packets; \( E_4 \) is the total time spent by a node in the sleep state; \( E_{\text{pl}} \) is the total energy spent during the LPL phase; and \( E_{\text{tx}} \) is the total energy spent by the node for listening to the channel.

To better understand how to calculate those quantities, it may help to sum up the steps followed by senders and receivers with reference to the particular scenario considered here.

Every \( T_0 \) a node wakes up to receive possible packets and:

1. performs an LPL for \( T_{\text{pl}} \) seconds;
2. sends a HELLO packet;
3. waits for the reception of a packet for a maximum of \( T_{\text{pl}} \) seconds and, if its neighbor has something to transmit before that time, it starts the reception phase;
4. returns to the sleep state.

When a node has something to transmit and the wake-up time of the intended receiver is approaching, then it

1. wakes up \( t_{c1} \) seconds in advance;
2. waits for an additional \( t_{c2} \) seconds at latest to receive the HELLO from the neighbor, otherwise it goes back to the sleep state;
3. receives the HELLO message;
4. starts the back-off timer and waits while listening to the channel;
5. transmits the packet and then returns to the sleep state.

With reference to the contention started by a sender before the transmission, the backoff time is evenly distributed in an interval \([0, CW]\) and thus the average duration of that interval is

\[
T_{\text{CW}} = \frac{CW}{2}.
\]

Consider that node 1 generates and transmits a packet every \( 1/R \) seconds; node 2 generates one packet in \( 1/R \) seconds, receives one from node 1 and transmits both. In general, every \( 1/R \) seconds node \( i \) generates 1 packet, receives \( i-1 \) packets from its predecessor and transmits all the \( i \) packets to its successor in the path to the sink. Define \( K \) the number of \( T_0 \) length slots between the generation of two successive packets, it results

\[
K = \left\lfloor \frac{1}{RT_0} \right\rfloor
\]

Without loss of generality of the analysis, it can be assumed that \( 1/R \) is a multiple of \( T_0 \). By applying the aforementioned steps for transmission and reception to (37), the energy consumed by each node in this interval of time can be calculated as follow.

\[
E_{\text{tot}} = K T_{\text{pl}} P_{\text{pl}} + K T_{\text{pl}} P_{\text{tx}} + (i-1) [ T_{\text{pl}} P_{\text{tx}} + T_{\text{pl}} P_{\text{rx}} ] + \sum_{k=1}^{i} [(k-i) T_{\text{pl}} P_{\text{tx}} + T_{\text{pl}} P_{\text{tx}} + T_{\text{pl}} P_{\text{rx}} ]
\]

\[
+ \sum_{k=1}^{i} [ T_{\text{pl}} P_{\text{tx}} + T_{\text{pl}} P_{\text{rx}} ] + i T_{\text{pl}} P_{\text{tx}}
\]

\[
+ (k - i) T_{\text{pl}} P_{\text{tx}} + T_{\text{pl}} P_{\text{tx}}
\]

\[
+ i T_{\text{pl}} P_{\text{tx}} + i T_{\text{pl}} P_{\text{rx}}
\]

\[
+ i T_{\text{pl}} P_{\text{tx}} + i T_{\text{pl}} P_{\text{rx}}
\]

(38)
\[ T_s = KTO - KT_{pl} - KT_{hello} - (i - 1)(Trx + T_{CW}) - (k - i)TO + \]
\[ -i(Tc1 + Tc2) - iT_{hello} - iT_{CW} - iT_{tx} \]

(39)

where \( T_{pl} \) is the duration of an LPL interval and \( P_{pl} \) is the power consumed by the radio in such phase; \( T_{hello} \) is the time needed to transmit an HELLO packet according to its length and to the available bandwidth; \( P_{tx} \) is the power consumption in transmission; \( T_{rx} \) is the time needed to receive a packet depending on its length and on the available bandwidth. Actually, \( T_{rx} = T_{tx} \) but in the formulas they have been distinguished for separating the transmission and reception steps; \( P_{rx} \) is the power consumed by the radio for receiving a packet; \( P_{s} \) is the power consumed by the radio for channel listening; \( P_{i} \) is the power consumed in the sleep phase.

From (38), \( PW_{TOT} \) and \( PW_{av} \) can be derived as follow

\[ PW_{TOT} = \sum_{i=1}^{N} \frac{E_{tot}^i}{KT_0} \]

(40)

\[ PW_{av} = \frac{1}{N} PW_{TOT} \]

(41)

6. Simulation model

In this section, the simulation model configured in Omnet++ simulator [5] is presented. The main objective of the simulations is to show how the designed scheduler allows to achieve a considerable energy saving without excessively compromise the efficient data delivery with respect to the current MAC protocol compliant with IEEE 802.15.4. Therefore, power consumption and packet delivery ratio are selected as metrics for performance evaluation. These metrics have been analyzed with the change in important parameters such as the duty cycle period duration, \( T_0 \), and the packet rate.

The network layer has been implemented according to the ZigBee specifications, whereas the MAC and PHY layers are compliant to the IEEE 802.15.4 specifications [23]. The proposed schema has been introduced into the MAC layer as a manager for the IEEE 802.15.4 MAC protocol.

One of the key mechanisms of the AS2-MAC is the Announcement Packet, a new network command that enables self-configuring network. The Announcement Packet carries the information concerning the transmission time of the node. It has been defined as a ZigBee MAC Command Frame, exploiting one of the reserved values. In this way, an open feature of the standard has been exploited with no need to introduce a new packet format. This packet has also been used to greatly reduce the hidden node problem, that is one of the main problems afflicting ad hoc networks based on the simple CSMA-CA protocol. The Announcement Packet is one hop forwarded by the neighbors of the source node, so that also nodes in two hops are aware of the sender transmission time, avoiding possible collisions at a common neighbor.

The network topology is a multi-hop chain network and the area in which nodes are distributed is set at 250 m.

Table 1

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network topology</td>
<td>Chain network</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>6 (5 nodes and 1 sink)</td>
</tr>
<tr>
<td>Sensors per node</td>
<td>1</td>
</tr>
<tr>
<td>Frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>250 kbps</td>
</tr>
<tr>
<td>Transmission power</td>
<td>0 dBm</td>
</tr>
<tr>
<td>Packet Rate</td>
<td>0.1, 0.2, 0.5 pkts/s</td>
</tr>
<tr>
<td>Packet error rate</td>
<td>0.1%</td>
</tr>
<tr>
<td>( T_0 )</td>
<td>10, 20, 40 s</td>
</tr>
<tr>
<td>Wake time</td>
<td>1 s</td>
</tr>
<tr>
<td>Payload size</td>
<td>60 bytes</td>
</tr>
<tr>
<td>Packet size (at PHY layer)</td>
<td>91 bytes</td>
</tr>
<tr>
<td>ACK size</td>
<td>11 bytes</td>
</tr>
<tr>
<td>ACK waiting timer</td>
<td>8.64e–4 s</td>
</tr>
<tr>
<td>Buffer size</td>
<td>3.12 Kbyte</td>
</tr>
<tr>
<td>TurnAround Time</td>
<td>1.92e–4 s</td>
</tr>
<tr>
<td>Modeled system on chip</td>
<td>ST/Ember EM250</td>
</tr>
</tbody>
</table>

Each node is provided with an on-board sensor that generates data packets with a fixed size of 60 bytes at the application layer. A Constant Packet Rate (CPR) traffic has been chosen, since WSNs are often used in applications aimed at periodic environmental monitoring. More in detail, we have chosen particularly high packet rates, which are usually used, for example, in surveillance applications or to monitor critical conditions in some environments (e.g., gas leak, maintenance of a constant temperature for the preservation of sensitive drugs, etc.). The Wake Time, i.e. the time interval during which a node transmits or receives data packets, has been set at 1 s. The frequency band used is 2.4 GHz and antennas are omnidirectional. Packets are discarded by the receiver with a packet error rate equal to 0.1%.

All the main simulation parameters are reported in Table 1. Among these parameters, the possible values of the packet rate and the size of the transmission buffer are chosen according to the mathematical analysis described in Section 4. In particular, all the values of the packet rate are set to ensure the stability of the system, while the buffer size was set approximately equal to the \( K_{min} \) value that is obtained when \( T_0 \) is equal to 20 s and the packet rate is equal to 0.5 pkts/s. In this way, the buffer overflow is avoided for all parameter combinations with the exception of that combination where \( T_0 \) is equal to 40 s and the packet rate is equal to 0.5 pkts/s.

All the simulation results are characterized by a 95% confidence interval with a 5% maximum relative error.

7. Results and discussions

7.1. Comparison with AS-MAC

The first step of validation phase concerned the comparison between our solution and the AS-MAC protocol, mainly by comparing the respective mathematical models.

The first major difference between the two protocols is represented by the different scheduling of the nodes wake-ups. In the case of AS-MAC, a node knows when each of its neighbors wakes up in receiving mode and thus it knows...
when to wake up for transmitting to the desired neighbor. In the case of AS2-MAC, a node knows when each of its neighbors wakes up in transmission mode and then it knows when to wake up for receiving data. This different behavior has two very obvious advantages for AS2-MAC. First of all, while with AS-MAC nodes that want to transmit to a same receiver must enter into contention for the channel access when the destination node wakes up for receiving, with AS2-MAC nodes never come in contention because they transmit in different time instants. This behavior ensures less packet delay and a greater PDR for AS2-MAC, thanks also to the absence of any collisions that may occur due to simultaneous transmissions. Of course, these differences are less evident in the chain topology (used for the development of mathematical models described in the previous sections), because, in this topology, nodes cannot collide with each other during transmission. However, analyzing the formulas (34) and (38), relating to the power consumption of the two protocols, it is evident that, also in the chain topology, AS-MAC protocol presents additional power consumption due to the channel listening during the backoff procedure; this additional power consumption is not present in the AS2-MAC protocol. Indeed, although there are no collisions in the chain topology, nodes compliant with the AS-MAC protocol perform anyway the backoff procedure required by the protocol.

Moreover, as mentioned in the Related Works section, when a node compliant with AS-MAC protocol has to transmit a broadcast packet, it is forced to retransmit the packet to each neighbor, eventually several times if collisions occur. Otherwise, with AS2-MAC protocol, the packet broadcasting is natively supported: when a node needs to transmit a broadcast packet, all its neighbors are already awake and ready to receive it.

Analyzing now the mathematical models of the two protocols, another substantial difference is evident: on equal duty cycle period, AS2-MAC protocol is capable of supporting much higher packet rate. This behavior is evident in the graph of Fig. 8, where the trends of the maximum packet rate, \( R_{\text{max}} \), as a function of the duty cycle period, \( T_0 \), are shown. The curves are the graphic representation of the formulas (16) and (36), where \( N \) is equal to 5 nodes, whereas \( V \) is equal to 111 packets (the Wake Time, \( W_t \), of AS2-MAC is 0.5 s. All the other parameters needed for the calculation of \( V \) are reported in Table 1). This result proves that, in addition to the energy-efficient features (which are evaluated more in detail in the next Section), the proposed solution has also a good flexibility in addressing different application requirements. Indeed, once a duty cycle period is fixed, AS2-MAC protocol is able to efficiently manage a wide range of application data rates.

### 7.2. Simulation results

The second step of the validation phase provides the comparison among the results obtained with the simulations carried out through Omnet++ and the values provided by the mathematical model. In particular, Fig. 9 shows the trend of power consumption when a packet rate of 0.5 pkts/s is generated by nodes. As can be seen, the curves coming from simulations and those derived from the mathematical model are very close each other, and of course the mathematical model provides a lower power consumption due to some necessary simplifications. Furthermore, as expected, the power consumption of the nodes is drastically lowered by the increase of the duty cycle period, since nodes remain more time in the OFF state saving a greater amount of energy. Figs. 10–13 show, instead, the comparison between the mathematical model and the simulation model in relation to PDR. In order to
evaluate these graphs, it is important to note that the size of the transmission buffer, as before stated, guarantees the absence of buffer overflow for all combinations of parameters with the exception of the case where $T_0$ is equal to 40 s and the packet rate is equal to 0.5 pkts/s. Therefore, as expected, the analytic curve well approximates the simulation results when the operating conditions are far from the limit case (Figs. 10 and 11), whereas, when the combination of $T_0$ and packet rate approaches the critical threshold, the results of the simulated model significantly deviate from the analytical curve (Fig. 12). Exceeded this threshold, the mathematical model does not approximate simulation model since the percentage of lost packets due to buffer overflow is particularly high (Fig. 13).

The third step of the validation phase was aimed at comparing the AS2-MAC and the solution currently used by ZigBee standard, which does not provide any energy-efficient mechanism. In particular, Figs. 14 and 15 show the average power consumption with the change in distance between sender and receiver (in number of hops) for several values of $T_0$. The curves clearly show that, whatever the value of $T_0$, the proposed scheduler reaches a substantial reduction of the power spent by each node, since idle listening and collisions are avoided. More in detail, for $T_0 = 10$ s the minimum power gain is around 74%, while for $T_0 = 40$ s it reaches more than 90% with respect to the ZigBee solution. Furthermore, it is important to observe that, unlike one might expect, the node that consumes more power is not the one closest to the sink, but those within the chain. By using the proposed scheduler, indeed, the power consumption depends on the number of awakenings that a node performs during each duty cycle period, therefore nodes with more neighbors wake up more often, so depleting more energy. For this reason, as can be seen
from the graph, nodes at the edges of the chain, having only one neighbor, waste less power than the remaining nodes which have two neighbors. Finally, it is useful to highlight that the power consumption drops slightly when the packet rate increases. This behavior can be observed when comparing carefully Figs. 14 and 15, but it is even more evident in Fig. 16, which shows the power consumption for various values of the packet rate. This result is because the simulated node model has a power consumption in idle listening higher than the consumption in transmission, as can be seen from the data reported in Table 1. For this reason, if the packet rate is low, a node spends more time in idle listening during its wake interval, whereas, if the packet rate is higher, a node spends more time in transmission so consuming less power.

Figs. 17–19 describe the comparison between ZigBee and AS2-MAC with respect to the PDR. In particular, Fig. 17 shows the performance of the two protocols when nodes generate packets at a rate of 0.2 pkts/s. As expected, the performance of AS2-MAC are very close to those of ZigBee with this combination of parameters, since both protocols reach 99% of packets successfully delivered to the sink. To understand this result, it is necessary to point out, once again, that the size of the transmission buffer has been set to prevent buffer overflow for all values of $T_0$ and packet rate, with the exception of the combination in which $T_0$ is equal to 40 s and the packet rate equal to 0.5 pkts/s. Then, when the operating conditions of nodes are close to this limit, the PDR decreases, slowly when $T_0 = 20$ s and the packet rate is equal to 0.5 pkts/s (Fig. 18), quickly when $T_0 = 40$ s and the packet rate is equal to 0.5 pkts/s (Fig. 19).

8. Conclusions

A new MAC protocol able to significantly reduce the power consumption in a WSN and compatible with the IEEE 802.15.4 standard, is presented. The defined protocol is based on an efficient setting of the duty cycle since each node exploits information about the periodic transmissions of its neighbors to schedule its own wake up intervals and avoid useless awakenings. The solution is robust to network changes since the information stored by each node is updated every time a variation is detected. The defined protocol has been validated through the discrete event simulator Omnet++ and simulation results have highlighted an improvement in power consumption with respect to the current ZigBee MAC protocol, although at the cost of a small decrement in the PDR when the packet rate is relatively high. However, for most of the practical applications of WSNs, nodes generate particularly low data traffic. Furthermore, a mathematical model consistent with the proposed solution was defined in order to carry out a

---

performance comparison between the simulated scenario and the analytical one. The results have shown that the analytical model is well defined, since it is able to accurately predict the system behavior. The evaluation of the proposed protocol in a more complex and dynamic scenario is a natural evolution of this study.

References

[23] IEEE 802.15.4-2006. IEEE Standard for Information technology – Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs).
[26] 802.15.4e-2012 – IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC.

Luca Anchora received M.Sc. degree in Computer Engineering in 2008 from the University of Lecce, Italy, and Ph.D. degree in Computer Science and Engineering in 2012 from the IMT Institute for Advanced Studies of Lucca, Italy. He has been a visiting student at the Communication Systems Division, Department of Electrical Engineering, at the Linköping University, Sweden, where he conducted some research on spectrum sharing techniques in LTE cellular networks. He has been on leave at the SIGNET lab of the University of Padova, Italy, as a postdoc. He has published several scientific papers and served as a reviewer for several conferences and journals. His research interests are on networking, particularly on resource allocation in wireless networks and applications of game theory.

Antonio Capone received the Laurea Degree in Computer Engineering with honors at the University of Salento, Lecce, Italy, in 2010. His thesis concerned the definition and validation of a scheduler to minimize the energy consumption in a WSN based on ZigBee standard (IEEE 802.15.4). Since February 2011 he works as R&D Engineer for CISC Semiconductor GmbH in Klagenfurt, Austria.

Vincenzo Migliari received the Laurea Degree in Computer Engineering with honors at the University of Salento, Lecce, Italy, in 2012. Since January 2009 he collaborates with IDA Lab – Identification Automation Laboratory at the Department of Innovation Engineering, University of Salento. His activity is focused on the definition and implementation of new tracking system based on RFID technology and on the design and validation of innovative communication protocol aimed to reduce power consumption in Wireless Sensor Networks. He is also involved in the study of new solutions for building automation. He authored several papers on international journals and conferences.
Luigi Patrono received MS in Computer Engineering from University of Lecce, Italy, in 1999 and PhD in Innovative Materials and Technologies for Satellite Networks from ISUFI-University of Lecce, in 2003. He is Assistant Professor of Network Design at the University of Salento, Lecce. His research interests include RFID, EPC global, 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 (IEEE SoftCOM conference). He is author of about 60 scientific papers published on international journals and conferences and four chapters of international books.

Francesco Simone graduated cum laude in Computer Engineering at University of Salento, Italy, in April 2012. His thesis concerned the definition and validation of an energy-efficient MAC schema based on directional antennas for Wireless Sensor Networks. Since October 2012, he collaborates with IDA Lab - IDentification Automation Laboratory at the Department of Innovation Engineering, University of Salento. His activity is focused on “PIF SVILUPPO AGRICOLO”, a European Project on food traceability based on RFID and NFC technologies.