Performance Evaluation of the Schedule Channel Polling MAC Protocol applied to Health Monitoring in the Context of IEEE 802.15.4

Luís M. Borges and Fernando J. Velez
Universidade da Beira Interior
Instituto de Telecomunicações, DEM
Covilhã, Portugal
lborges@lx.it.pt; fjv@ubi.pt

António S. Lebres
Universidade da Beira Interior
Unidade de Detecção Remota, DF
Covilhã, Portugal
lebres@ubi.pt

***Abstract*—The field of Wireless Body Area Networks (WBANs) has been growing considerably in the last years. A WBAN enables to monitor vital body signs through the placement of sensors on the human body and offers to patients more flexibility and mobility. In this paper, we examine the Quality of Service (QoS) performance metrics evaluation (single and multi-hop topology) of the Scheduled Channel Polling-MAC (SCP) protocol, whilst considering the IEEE 802.15.4 standard in the context of remote patient monitoring applications. We applied the MiXiM Framework (based on OMNeT++) in the network simulations. Our simulations results help to clarify some missing aspects in the original SCP protocol description. The IEEE 802.15.4 compliant Medium Access Control (MAC) is compared with the layer MAC layer in the absence of the IEEE 802.15.4. The results shows that considering low duty cycles, the IEEE 802.15.4 leads to lower energy consumptions in the absence and presence of piggyback. For multi-hop experiments in high duty cycles, the energy consumptions are lower with the AT86RF231 transceiver, comparing with the CC1100 and CC2420. Future work includes the adaptation of the SCP protocol to the IEEE 802.15.6 standard, which has been specially developed to optimize the WBAN performance at the physical (PHY) and MAC layers.**

Keywords-Wireless Sensor Networks, OMNeT++, MAC protocol, Modelling, Energy efficiency

I. INTRODUCTION

Wireless sensor networks (WSNs) answer to the need for nomadicity and mobility in the context of *in situ* event monitoring. The main purpose of this class of networks is to easily and quickly provide access to the information gathered from a set of sensor nodes scattered by a geographical region.

Energy efficiency is sought because it increases the lifetime of the network, where nodes are usually battery powered. These networks are usually data centric, where different protocols operate at different layers of the network protocol stack, with the goal of efficient management of transmission, reception, sleep and polling operation modes. Major sources of energy waste are idle listening, packet retransmissions (due to packets collisions), unnecessarily used transmission power, sub-optimal utilization of the available resource, overhearing and control overhead [1], [2], [3]. The radio duty cycle is the portion of time the node's radio is on (transmitting; receiving, or idle listening). Since the radio chip accounts for most of the energy consumed by a typical node, the duty cycle closely relates to the lifetime of the network. Adaptive duty cycling is a Medium Access Control (MAC) technique to reduce the duty cycle. Nodes follow a network-wide schedule with active and sleeping periods; during a sleeping period, the radios of all the nodes are off. Scheduled Channel Polling (SCP) [4] is an example of a duty-cycle based MAC protocol. In SCP, nodes contend for accessing the medium during active periods. It combines scheduling with channel polling in an optimal way, as does Low Power Listening (LPL) [5]. With periodic traffic, energy consumption can be minimized by using a scheduled listening algorithm, and the synchronization of the neighbour's channel polling time is maintained in a way similar to Sensor-MAC (S-MAC) [2], while adjusting the duty cycle to varying traffic conditions (see Fig. 1).

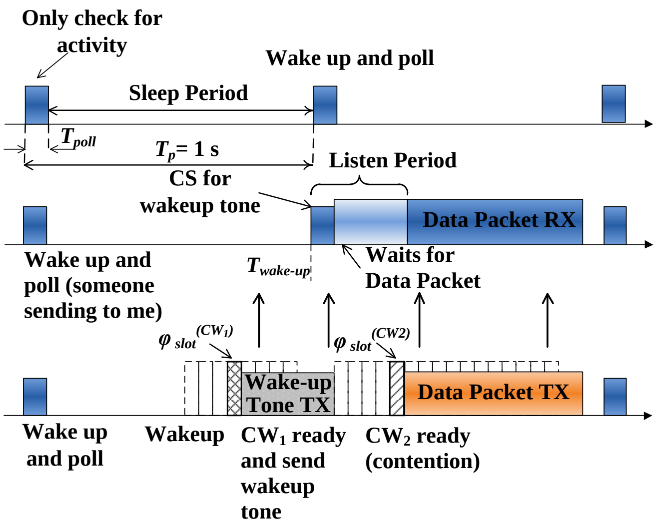

Figure 1. SCP mechanism and an ultra-low duty cycle example

The SCP protocol is a relatively recent protocol which employs energy efficient techniques to control the access to the medium. We decided to deeply study these techniques in order to investigate SCP based MAC protocols.

As stated by the authors from [7], the development of applications involving WSNs involve aspects of hardware capacity, deployment area, type of energy sources available, deployment cost, ambient conditions, the possibility of energy harvesting, etc. Therefore, it is of paramount importance to evaluate our application algorithm (e.g., MAC, routing) before deploying them into real world. There are some WSN evaluation tools and methodologies to test the performance of these algorithms and protocols, namely analytical modelling, simulation, emulation, testbeds and real deployment. However, in terms of cost-efficiency, the simulators, emulators and testbeds are the most effective tools we can use in our benefit to evaluate the performance of algorithms and protocols during the design, implementation and development process. Modelling the results may lead to inaccuracy when over-simplifications are considered. Real deployments are another way of testing and evaluating WSNs, but it is complex, costly and time-consuming, sometimes due to the deployment in irregular terrains or due to the need for high scale deployments.

There are several WSN MAC protocols with solutions for the energy wasting problem (e.g., S-MAC [2] and CrankShaft [8]), or suggestions to QoS performance improvement (latency decrease, packet delivery ratio increase, etc). Other ones offer improvements to previous published protocols. The majority of these authors analytically describe the protocol and directly implement it into hardware, whilst comparing the results with the ones obtained analytically.

The protocol specification includes a detailed description of the employed techniques, identifying the allowed ranges for each of the MAC protocol parameters. This way, if a researcher intends to replicate the protocol by means of simulation, the possibility of ambiguities will decrease considerably whilst minimizing the deviations in the performance results. This allows for the researcher to evaluate and perform comparisons with other MAC protocols with higher accuracy.

In [8], [9], [10], the performance of some of these protocols (e.g., Crankshaft [8], SEA-MAC [9] and BurstMAC [10]) is compared with the SCP results but the way how analytical/simulations results have been obtained needs clarification. In [4], details on the parameters and assumptions for SCP are missing. Notwithstanding, this is one of the most complete papers, in concern of protocol description, real experiment results and analytical approach.

The contribution from this work is to provide a detailed SCP performance evaluation through the implementation of the SCP in MiXiM Framework [11] simulator (based on OMNeT++). We do not suggest any change to the protocol.

One of the main issues that have been clarified during our implementation of SCP in the MiXiM Framework is related with the time parameters in the SCP frame structure, synchronization procedures and traffic generation at the application layer. Moreover, we have addressed the performance results for different metrics (single- and multi-hop topologies).

The comparison of the power consumption is performed between the cases with explicit SYNC packets and piggybacking synchronization (given by the "adapted" analytical model from [4]) and the simulation results, obtained from the simulator with the CC1100 [12], CC2420 [13] and AT86RF231 [14] transceivers.

The remainder of this work is organized as follows. Section II addresses the related work and different studies about simulator's accuracy and experience gained from implementing other WSN MAC protocols in simulators. Section III describes the SCP, with particular focus on the two-phase scheduled channel polling mechanism. Section IV describes the SCP simulation framework implementation, namely the SCP simulation parameters and the available SCP simulator layer modes. In Section V, energy performance is addressed (with periodic and heavy traffic) in the presence and absence of piggyback synchronization, as well as results for the throughput and energy consumption. Section VI evaluates the multi-hop performance in a linear chain. Moreover, energy consumption is evaluated for different contention window sizes, as well as latency and data packet success rate. Section VII presents the conclusions and suggestions for future work.

II. RELATED WORK

WSN MAC protocol performance evaluation through simulation is becoming more common in the research community. Before evaluating the MAC protocol the first task is to choose the most suitable simulator to implement the protocol/algorithm. Recently, several works compare the most common simulation platforms, emulator and testbeds for WSNs. The authors from [7] propose a taxonomy for the different performance evaluation tools for WSNs, including simulation, emulation and testbeds. They propose a horizontal and vertical analysis at different stages of performance evaluation, comparing the cost, complexity, time and effort increases (whilst improving the accuracy) for all competitor tools.

Other authors [15] have studied the performance of recent network simulators used both in academia and industry. Reference [15] gives an answer to the question of which network simulator should be used, especially if the user (researchers and graduate students) are interested in achieving a high simulator performance. From this study, the authors concluded that OMNeT++ is one of three simulators (ns-3, OMNeT++, JiST) capable of carrying out large-scale network simulations in an efficient way.

Based on these previous studies, we decided to explore the SCP in the OMNeT++ simulator, specifically the Mobility Framework [16] and then the MiXiM Framework [11], both suitable for WSNs simulations.

Other studies decided to implement a slotted CSMA/CA for IEEE 802.15.4 wireless sensor network [17]. This study addressed the performance analysis of the slotted CSMA/CA, for different network settings, by means of simulation modelling. They intended to understand the impact of the protocol attributes (superframe order, beacon order and backoff exponent).

Other topic related with simulation is the accuracy of the simulation results. Simulations abstract the network behaviour from reality in several ways. The authors from [18] have focused their study on the simulation abstractions with respect to MAC protocols performance. Moreover, an analysis of what are the main sources of deviation is presented, while possible solutions to improve simulations are given. The performance analysis is performed for four well known MAC protocols: i) B-MAC, ii) T-MAC, iii) Crankshaft and iv) LMAC.

Recently, other authors [19] have tried to describe their experience from the implementation of the T-MAC protocol in the Castalia simulator. Their main objective is to give insight into some of the pitfalls where new solutions or protocols are

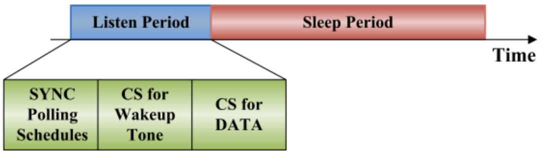

Figure 2. SCP time frame period structure

being proposed, in order to help researchers write better technical papers, with a more rigorous description of the used techniques/algorithms/mechanisms. Moreover, they intend to give a practical guide of how the T-MAC implementation was conducted.

As a matter of fact, this is not the first MAC protocol implementation in a simulator, where a performance evaluation is performed. The previous work tries to give hints and a practical guide to implement the T-MAC into Castalia simulator.

However, to the best of our knowledge an evaluation of the SCP performance does not exist, where a detailed description of the SCP techniques/mechanisms and main algorithms are explained, while the impact of certain parameters in the simulation is explained. Moreover, we intend to address the performance impact in a healthcare application, considering the SCP as the employed MAC protocol.

III. Scheduled Channel Polling Protocol

In Scheduled Channel Polling (SCP) [4], time is divided into time frames which are indefinitely repeated. Fig. 2 shows how such a time frame is further subdivided into Listen and Sleep Periods. The Listen Period contains a period to synchronize the polling schedules, a period to perform Carrier Sense (CS) for wake-up tone detection, and a period to carrier sense for data packets. During the Sleep Period, a node is able to send/receive data (after sensing a free medium) and can receive/send an acknowledgment (ACK) packet, after which it goes to the sleep mode for the rest of the Sleep Period. SCP uses channel polling synchronization of neighbouring nodes using SYNC packets. As shown in Fig. 1, nodes sample the channel at the same time throughout the network. This means that a sending node sends only a short preamble, which starts right before the receiver start listening.

SCP combines scheduling with channel polling to minimize the energy consumption in case of periodic traffic. In SCP, nodes can synchronize their schedules in two ways. The first option is for each node to broadcast its schedule in a SYNC packet, at every synchronization period. This is the method used by S-MAC. In our simulation framework we define a maximum time interval for the synchronization period of 60 ms. The second option is to piggyback the schedules in the data packets when the packets exchanged are broadcast. This technique is more efficient.

Once neighbour nodes are synchronized, each of them wakes up after a certain channel polling period, T_p, and polls the channel for activity sensing. If a node detects an idle channel and it has no data to send, it schedules its next wake-up time and shuts down its radio. If the node has data in its Transmission (Tx) MAC queue, it uses the process depicted in Fig. 1, enabling the data packet transmission in the next time instant the node wakes up, t_{wakeup}. Besides, it schedules the next wake-up before going to the sleep mode again. SCP reduces collisions, overhearing, control overhead and idle listening. It solves packet collision by using a two phase contention window scheme (when the sender node wants to transmit a data packet). Contention Window 1 (CW_1) is divided into CW_1^{max} equal time slots. When a node wants to transmit a data packet, it randomly and uniformly chooses a time slot $\varphi_{slot}^{CW_1}$ in CW_1 and perform a carrier sense for the duration of that time slot. The slot choice from the ones available in the interval follows a uniform distribution, as follows (1):

$$\varphi_{slot}^{CW_1} \epsilon [1;\ CW_1^{max}] \quad (1)$$

where CW_1^{max} is the maximum size of CW_1, in slots. Each node from the network contends to access the channel when it has data packets to be sent. If the node detects an idle channel it starts to send a preamble. This preamble contains the destination address of the node to receive the data packet. The preamble has a minimum duration of 16 bytes, plus the equivalent time of remaining CW_1 time slots the node did not use for contention. After sending the wake-up tone, the node immediately enters into the Contention Window 2 (CW_2), which is divided into CW_2^{max} equal time slots.

When the node enters CW_2, it randomly chooses a slot $\varphi_{slot}^{CW_2}$, (from the following interval) by using a uniform distribution, as described by (2):

$$\varphi_{slot}^{CW_2} \epsilon [1;\ CW_2^{max}] \quad (2)$$

where CW_2^{max} is the maximum size of CW_2. The node then starts a carrier sense in CW_2 (which has the same duration as CW_1), as shown in Fig. 1. If the node detects an idle channel, it starts sending the data packet. After sending/receiving the data packet, the sender/receiver node checks whether it has any data packets pending to be sent. If so, the node repeats the process described above before going into sleep mode.

IV. Implementation of the SCP Simulation Framework

A. SCP Simulator Parameters and General

The SCP has been first implemented by us in the OMNeT++ simulator [20], using the Mobility Framework initially from [16]. Then, the SCP code has been ported to the new MiXiM Framework [11]. The MiXiM Framework supports CC1100 [12] and CC2420 [13] radio energy consumption models, as well as several propagation models. Moreover, we have extended it to work with AT86RF231 [14] radio transceiver. The latter is the best radio available in the market in terms of energy consumption. From all the radio transceivers the CC1100 is the only one that is non-compliant IEEE 802.15.4. We intend to compare the performance of SCP with IEEE 802.15.4 compliant and non-compliant transceivers.

IEEE 802.15.4 is a double standard: Physical layer (PHY) and MAC. By definition, if we program a board with our own MAC, it cannot be IEEE 802.15.4 compliant. The reason for using an IEEE 802.15.4-compliant radio is that hundreds of millions of 15.4 chips have been sold in the world and it is by the far the *de facto* standard for those low power networks. The key aspects of the IEEE 802.15.4 standard are the following: the Tx output power can be tuned from approximately -30 dBm to +3 dBm on the majority of the radios; the sensitivity has to be <-85 dBm, but the sensitivity for some radios can go as low

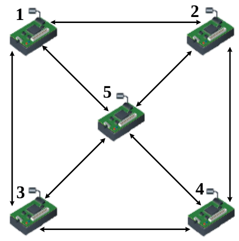

Figure 3. Single-hop scenario for SCP

TABLE I. TYPICAL VALUES FOR CC1100, CC2420 AND AT86RF231 RADIOS

Symbol	Meaning	CC1100	CC2420	AT86RF231
I_{tx}	Current in transmitting	16.9 mA	17.4 mA	11.6 mA
$I_{rx}= I_{listen}$	Current in receiving/listening	16.4 mA	18.8 mA	12.3 mA
I_{poll}	Current in polling	16.4 mA	18.8 mA	12.3 mA
I_{sleep}	Current in sleep	0.02 mA	0.000021 mA	0.00002 mA
r_B	Data Rate	250 kbps	250 kbps	250 kbps
S_{min}	Sensitivity	-111 dBm	-94 dBm	-101 dBm
f_b	Carrier Frequency	868 MHz	2.4 GHz	2.4 GHz

as -101 dBm; there are 16 different frequency channels (2.405 GHz+n×5 MHz), which are orthogonal (i.e., adjacent channels do not interfere); a radio chip consumes power when it is on (about the same power when transmitting, receiving or idle listening), and no power when the radio is off; it takes 192 µs for a radio chip to perform the transition from Tx to Reception (Rx) mode, and vice-versa; The maximum length of the IEEE 802.15.4 frame should be 127 bytes. We consider an IEEE 802.15.4 non-beacon scheme in our simulations.

Our simulator considers single and multi-hop network topologies, as shown in Figs. 3 and 4. The nodes may communicate only with reachable neighbour nodes. The deployment area defined in the configuration file for the single-hop scenario is 400×400 m^2, while for the multi-hop scenario it is 800×100 m^2. For each experiment, a set of six seeds is chosen for the random generator of SCP.

In our simulator protocol stack, each layer is connected with the higher and lower layers. There is an application module layer connected to the MAC and PHY layer modules. Besides these modules, a battery module and statistics battery module are included in order to simulate the network or node lifetime, while varying the capacity of the batteries. In the end of the simulation the statistics battery module presents the battery statistics collected during the simulation. Nodes are assumed static.

The typical values for the current consumption, data rate, and sensitivity of the radios are also defined in the simulation configuration file, as presented in Table I. Several channel parameters are defined in the MiXiM simulation Framework: *maxTXPower* corresponds to the maximum transmission power allowed in the channel (1.99 mW); *txPower* is the transmission power (1 mW), while *thermaNoise* is the electronic noise generated by the thermal agitation (-110 dBm). The *Analogue Model Type* parameter is the propagation model used in the channel (SimplePathlossModel), while *alpha* is the propagation exponent for the free-space path loss formula (2). The *berLowerBound* parameter is the lower bound of the bit-error-rate (1×10^{-8}). *Modulation type* is the modulation that the transceiver works with (MSK for IEEE 802.15.4 non-compliant radio transceivers and OQPSK for IEEE802.15.4 compliant ones). The *queueLength* parameter is the maximum number of packets waiting to be transmitted allowed in the queue (50). In our simulations, the three radio models (from MiXiM) do not include the poll state [20]: when a node wakes up and does not have data packets in its MAC queue, it polls the channel for 2 ms, while the power consumption is the same as in the receiving state.

B. SCP Simulator Layer Modes

During the SCP implementation in the MiXiM we have identified the need of implementing some modes in the simulator, in order to optimize it. Moreover, we need to be able to conduct diverse types of experiments regarding single and multi-hop topologies, with different traffic patterns generation, and enabling/disabling ACK feature packets. These requirements lead to the addition of the following modes in our simulator:

a) RTS/CTS mode

The overhearing avoidance problem is solved based on MAC frame headers. The receiver node examines the destination address of a packet immediately after receiving its MAC header. If the packet destination is for other node, it immediately stops receiving the packet and goes to sleep.

Since control packets cause protocol overhead, SCP has the option to enable or disable the RTS/CTS/DATA/ACK, or simply the DATA/ACK exchange scheme for unicast traffic. When the RTS/CTS mechanism is enabled, overhearing avoidance is performed in the same way as in the S-MAC. In our simulator, this option is named as the *acknowledgment packet* one.

b) Throughput mode

Another mode we have implemented in the application layer is the *throughput* one. It is useful to test the network‘s response capacity as the number of transmitting nodes increases. With this mode enabled, the application layer generates a data packet, and then sends it to the lower layers, until it reaches the MAC layer, where it is stored in the Tx MAC queue. Then, the node schedules the next wake-up to send the data packet received from the upper layers the fastest way possible, starting to send the next packet as soon as the prior packet is sent. When the node sends the packet the physical layer informs the MAC layer that the transmission has finished and sends the received message to the application layer. As soon as the application layer receives this message it builds another data packet and sends it to the lower layers until it reaches the MAC layer, repeating all the scheduling process described previously.

c) Synchronization mode

There are two modes to choose the synchronization type: the sync slave one, which uses explicit SYNC packets, and the *piggyback* one, which piggybacks the schedule information into the broadcasted data packets.

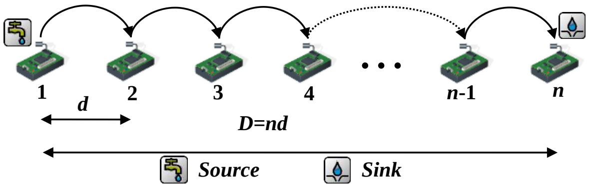

Figure 4. Multi-hop linear chain scenario for the SCP

d) Multi-hop mode

Another mode available in the simulation framework is the *multi-hop* one, which enables the nodes to organize themselves. Figure 4 presents the linear chain of a WSN, where *d* is the distance between the sensor nodes, *D* is the distance between the source and sink, and *n* is the number of nodes. Only the application layer from node 1 generates data packets to be sent to the other nodes. The node *n* receives data packets and answers with ACK packets when a data packet is received correctly. The remaining nodes are the so-called forwarder's nodes, which receive and forward the data packets and the answer and receive ACK packets through the linear chain. For this mode, node 1 is the source node and node *n* is the sink one. The remaining ones only had to forward the packets to next sensor node with the identifier i_d+1.

V. Single-hop Performance Results

A. Power Consumption without Piggyback (periodic traffic)

It is worthwhile to compare simulation and analytical results for the average power consumption for each node. When no piggybacking is considered, we define the "adapted" analytical model (lower and upper bounds) from the original SCP work [4] and ran the simulator to extract the values.

The employed topology is the one from Fig. 3 (network composed by five nodes). In this simulations, the *sync slave mode* is enabled, the synchronization time is defined as $t_{sync}=60$ ms and the inter arrival period is variable. Moreover the simulator also considers, t_{sync}, the synchronization period, which depends on the inter arrival period (already defined by the authors from [4]), T_{sync}. As the synchronization period increases the wake-up tone duration also increases. We consider for the maximum contention window size CW_1^{max}= 8 and CW_2^{max}= 16. For this experiment all five nodes from the network form in a single-hop configuration, where each node generates and broadcasts 50 data packets (with 50 Bytes each). Since the inter arrival period varies between 50 and 300 s, the traffic load is considered to be light. Simulations are run five times for each of the six seeds, generating a total of thirty experiments. For each experiment the average power consumption of each node is obtained, as shown in Fig. 5. The achieved 95 % confidence intervals are negligible.

Figure 5 presents a power consumption comparison between simulation and analytical results (lower and upper bound), as a function of the inter arrival period for three different radio transceivers. The CC1100 is the one that presents the highest power consumption when is compared with the CC2420 and AT86RF231 one.

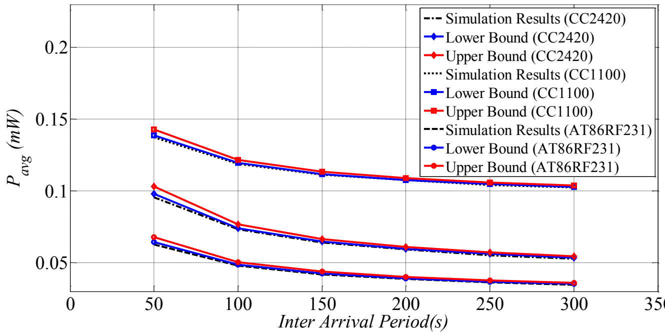

Figure 5. Lower/upper bound and simulation results with no piggyback for the power consumption per node of CC1100, CC2420 and AT86RF231

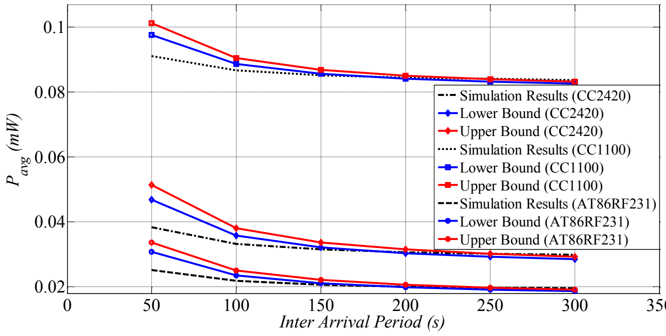

Figure 6. Lower/upper bound and simulation results with piggyback for the power consumption per node of CC1100, CC2420 and AT86RF231

The CC2420 transceiver consumes approximately 0.04 mW less than the CC1100 one, while AT86RF231 consumes around 0.08 mW less. This is due to the low power consumption that the AT86RF231 presents, combined with the long times the node remains in sleep mode. Besides, the mean square error (MSE) is calculated with respect to the analytical lower bound curve, and a value of 5.1319×10^{-7} mW for CC1100, a value of 1.4213×10^{-6} mW for CC2420 and a value of 6.3434×10^{-7} mW for AT86RF231 is obtained. This shows a high similarity between the simulations and analytical results.

If the t_{sync} parameter is changed from 60 to 50 ms while maintaining the remaining parameters, the power consumption becomes less similar to the lower bound (in comparison with the previous one). From these results, one may conclude that the power consumption with the *sync slave* mode enabled presents better results when t_{sync}= 60 ms. However, for values of t_{sync} lower than 50 ms it was not possible to achieve stable results with the SCP simulations. Moreover, for values of t_{sync} higher than 60 ms, higher values for power consumption are achieved.

B. Power Consumption with Piggyback (periodic traffic)

The lower and upper bound for the "adapted" analytical model are also defined for the case when synchronization

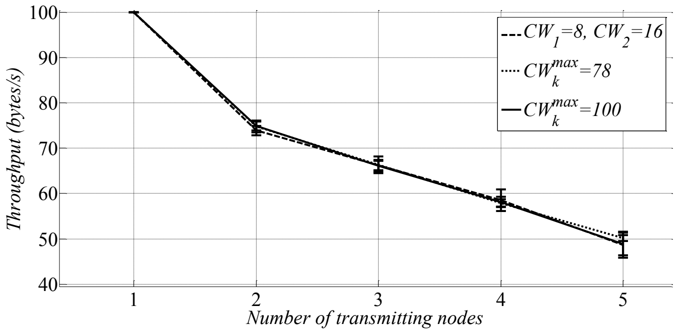

Figure 7. SCP throughput in function of the number of transmitting nodes and maximum contention windows sizes, $k \in [1, 2]$

schedules are piggybacked, whilst maintaining the simulator topology. However, since all the exchanged traffic is of the broadcast type, all the SYNC packets can be suppressed and the schedule synchronization is performed by using the piggyback technique. Hence, the *piggyback mode* is enabled for this experiment. There is no need to define the values for the parameter t_{sync}, as the schedule information is already piggybacked into the data packets, leading to a consequent decrease of node energy consumption. For this set of simulations, the wake-up tone duration is always equal to 2 ms. The only parameter that varies is $T_{data} \in \{50; 100; 150; 200; 250; 300\}$.

The dependence of the average power consumption of each node on the inter arrival period is shown in Fig. 6, where the simulations and analytical results (upper and lower bounds) are also presented for comparison purposes. The achieved 95 % confidence intervals are negligible.

For CW_1^{max}= 8 and CW_2^{max}= 16, the obtained simulation results are similar to the lower bound curve defined by the analytical model until the inter arrival period achieves 200 s. For an inter arrival period higher than 200 s, the simulation results are similar to the upper bound curve and even superimposes the upper bound when it achieves 300 s. The MSE values obtained for this experiment with respect to the lower bound curve are equal to 7.9825×10^{-6} mW, 1.3524×10^{-5} mW and 5.8479×10^{-6} mW, for the CC1100, CC2420 and AT86RF231 transceivers, respectively.

By comparing the power consumptions with explicit SYNC packets and with piggyback synchronization, we observe that the piggyback technique is truly efficient, since the power consumption with piggybacking achieves lower values than the one with explicit SYNC packets. In this set of experiments the poll period is set to 5 s, which is more than enough for a healthcare monitoring application reporting sensors values. Therefore in a healthcare application, if we use a sensor node with the AT86RF231 radio transceiver, whilst enabling the piggyback synchronization the node lifetime will increase significantly, due to the low associated power consumption.

C. Throughput Performance with heavy traffic load

In real world applications, WSNs do not have the capability to predict what type of traffic load is being exchanged with high accuracy. For example, in a healthcare scenario, the sensor nodes deliver the monitoring data from the vital signs of the patients at a lower data rate (in usual situations). However, when a patient presents anomalies in the vital signs the sensor node may deliver data at a higher rate, leading to the need for higher network throughput.

For this set of experiments, we have enabled the *throughput* mode at the application layer, in order to test the network‘s response capability as the number of transmitting nodes increases. By varying the number of transmitting nodes the contention time of the network will increase, leading to an increase of the collision probability. Each node generates 20 packets (with 100 Bytes each). The network, where topology is composed by five nodes, is shown in Fig 3. In these experiences the nodes poll the channel each second. Besides varying the number of transmitting nodes, the CW_1 and CW_2 contention window sizes also vary. The throughput is computed by dividing the amount of data packets received with success by the destination node with the time taken to arrive at this node.

By analysing Fig. 7, all the the results for the throughput (for all the three radio transceivers) match while the contention window sizes varies. The throughput is maximum (for all contending time configurations) when there is only one node transmitting, as expected, since there is no more nodes competing for the channel to send packets. When two nodes are competing for the channel, the throughput drops around 25 % of the maximum achievable throughput, because there is more than one node trying to transmit the data packet. As the number of transmitting nodes increases, the contending time will also increase, leading to a collision probability increase.

In terms of energy consumption when considering the IEEE 802.15.4 compliant radios the CC2420 energy consumption is higher than the CC1100 one, because the CC2420 achieves higher power consumption when the radio transceiver remains more time in sleep mode. Since the throughput mode enables the node to wake-up every second to poll the channel, the energy consumption will increase when compared with the CC1100. The AT86RF231 radio transceiver is the one that presents the lowest power consumption (from all the three radio transceivers), mainly due to the lowest values of the power consumptions it presents whatever the operating mode is.

VI. MULTI-HOP PERFORMANCE RESULTS

By considering experiences with multi-hop topology in a linear chain, we intend to evaluate the SCP energy and latency performance. We enabled the *multi-hop* mode, where only the application layer from node 1 generates data packets to be sent to the other nodes. Node *n* receives data packets and answers with ACK packets when a data packet is received correctly, as presented in Fig. 4. The wake-up tone duration is always set to be equal to 2 ms, whilst each node polls every second. In our topology, we have deployed *n-2* forwarders nodes, one sink node, and one source node. For the multi-hop evaluation the *acknowledgment* mode may be enabled/disabled.

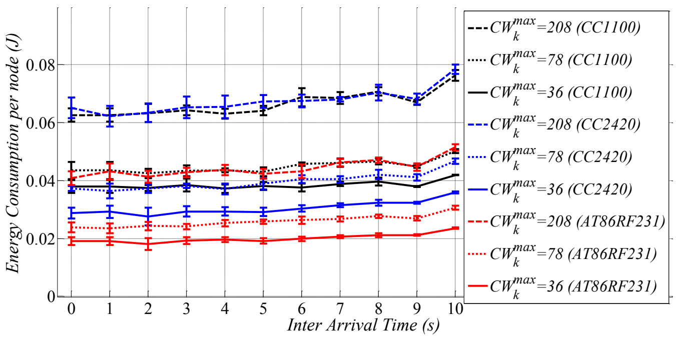

Figure 8. Average energy consumption in function of the inter arrival time and different contention windows sizes for multi-hop experiments, $k \in [1, 2]$

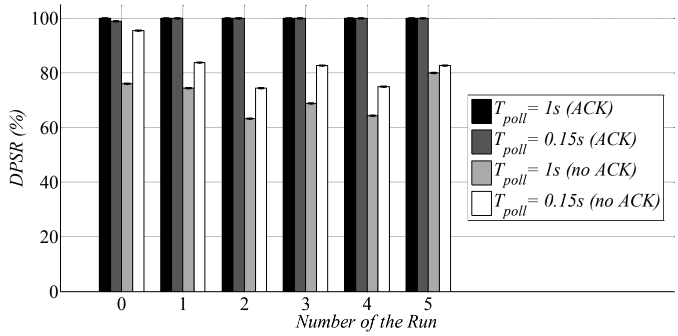

Figure 10. DPSR in function of the simulator run over 9 hops, for $CW_1^{max}=CW_2^{max}=16$, while enabling/disabling the ACK feature

A. Node energy consumption for Linear Chain

In multi-hop networks, there is no optimal working point and network performance depends on what application the network supports. The topology used in this set of experiences is the one presented in Fig. 4, with a network composed by a 9-hop linear topology with ten nodes (each one separated by 50 m from each other).

The source node generates 20 packets (each with 50 Bytes), with a inter arrival period that varies between 0 and 10 s. Nodes will rebroadcast the data packets along the linear chain until the packet reaches the most distant node from the chain. Considering an inter arrival time equal to zero means the application layer generates data packets and sends it to the lower layers the fastest way possible, as previously explained in the *Throughput* mode.

For a contention window sizes configuration of $CW_1^{max}=CW_2^{max}=208$, this combination is the one that presents the highest energy consumption per node for all the three radio transceivers, Fig. 8, due to the maximum contention time nodes use to avoid the packets collisions. Comparing the energy consumptions between the three radio transceivers the one that presents the lowest mean energy consumption is the AT86RF231 one. The CC2420 radio transceiver presents energy consumptions similar to the CC1100 radio transceiver. The energy consumptions are lower when comparing with the energy consumptions for multi-hop experiments in the original paper [4].

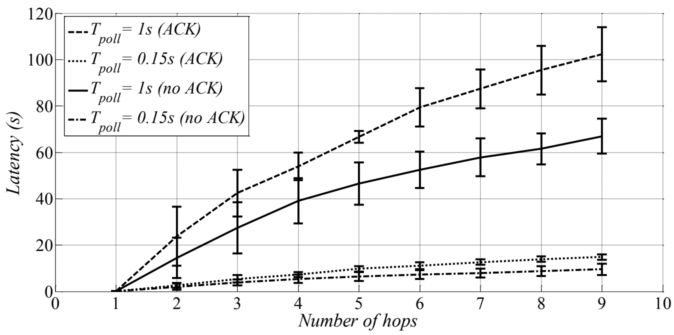

Figure 9. Mean packet latency over 9 hops in function of number of hops, for $CW_1^{max}=CW_2^{max}=16$, while enabling/disabling ACK packets feature

For all the configurations the energy consumption behaviour is approximately stable as the inter arrival time is increasing.

B. Node latency

This section addresses the SCP latency over a 9-hop network. We alternate the polling period between 0.15 s and 1s, enable or disable the *acknowledgment* mode, and set the contention windows maximum sizes to $CW_1^{max}=CW_2^{max}=16$ slots. When the inter arrival time is equal or higher than 1s (light traffic) it is expected that only one packet is sent through one hop, during one polling interval. When the polling period is changed to 0.15 s we expect that the latency decreases significantly, since the node will wake-up almost seven times in one second, and will schedule and send more data packets in less time.

Regarding Fig. 9, we may conclude that the latency curve when SCP polls each second (with the ACK packets feature enabled) is the one that presents the highest delay, due to the data packets retransmissions. When the polling period is set to 0.15 s, the maximum data packet latency is less than 20 s, but the energy consumption increases due to the number of times the node wakes up. The other two latency curves are related with the same experiments, but the ACK packets feature is disabled. With the ACK packets feature disabled for all three radio transceivers, we observe that the maximum data packet latency is less than 70 s (at the end of the linear chain), when polling each second. Moreover, when polling at each 0.15 s (with the ACK packets feature disabled), the latency is less than 10 s.

C. Node data packet success rate

To observe if there is any tradeoff when the ACK packets feature is enabled or disabled we have plotted the data packet success rate (DPSR) for all four configurations described previously. Figure 10 shows that, with the ACK feature enabled for both polling periods, there is around 100 % packet delivery success for all the three radio transceivers. When the ACK feature is disabled for both polling periods, the packet delivery in both experiments varies around 75 and 89 % for all the considered radio transceivers in this paper. Regarding these results, enabling ACK packets feature will result into an increase of data packet delivery. However, the tradeoff is the increase of the latency and energy consumption.

VII. CONCLUSIONS

In this paper, we provide a detailed analysis of the mechanism used in SCP, learned during our own experience while implementing the SCP protocol in the MiXiM Framework.

Our results help to clarify some missing aspects in the original SCP protocol description, as well as providing an evaluation of performance metrics (for single and multi-hop topology) by means of simulation. By comparing the power consumptions in the presence and absence of a fixed wake-up tone duration, the energy savings obtained with a fixed wake-up tone duration are adequate (more or less 0.005 mW). The use of IEEE 802.15.4 compliant radio transceivers leads to an energy consumption decrease in both cases (in the presence and absence of piggybacking). Considering a multi-hop linear chain, the SCP has a stable energy consumption as the inter arrival time varies. However, only the AT86RF231 transceiver presents the lowest mean energy consumption compared with the radio transceiver that does no complies for the IEEE 802.15.4 standard. In terms of delay, all the three radio transceivers present the same data packet latency, since the transmission data rate is the same and the specific time parameters of all these radio transceivers are approximately the same.

Considering a healthcare application where the nodes report periodically to a sink node the values from the sensors, the SCP can be selected has the MAC protocol that can deliver data efficiently to the sink with low power consumption cost. However, some tradeoffs should be taken into account when choosing the reporting period of the node. The tradeoffs between the latency, the DPSR and the energy consumption (considering a linear chain of sensor nodes in the Hospital) should be well balanced in order to fulfil the minimum requirements for the application.

Future work includes the adaptation of the SCP protocol to the IEEE 802.15.6 standard, which has been specially developed to optimize the WBAN performance at the physical (PHY) and Medium Access Control (MAC) layers. Moreover, the adaptive channel polling technique could be added to our simulator, in order to improve the throughput.

ACKNOWLEDGMENT

This work was supported by Unidade de Detecção Remota (UDR), Department of Physics from University of Beira Interior, by UBIQUIMESH, by Marie Curie Intra-European Fellowship OPTIMOBILE (FP7-PEOPLE-2007-2-1-IEF), PLANOPTI (FP7-PEOPLE-2009-RG), OPPORTUNISTIC-CR (PTDC/EEA - TEL/115981/2009), by the PhD Fundação para a Ciência e Tecnologia (FCT) grant SFRH/BD / 38356 / 2007 and by the programmatic budget from Instituto de Telecomunicações, as it enabled to make WSN hardware and software available to our work. We would specially like to acknowledge the contribution from J. Heidemann and W. Ye, who helped us in the interpretation of the SCP.

REFERENCES

[1] M. L. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," in *Proc. of the IEEE INFOCOM*, pp. 1740–1750, Hong Kong, China, Mar. 2004.

[2] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Networking Transactions*, vol. 12, no. 3, pp. 493–506, 2004.

[3] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, 2010.

[4] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *Proc. of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys'06)*, pp. 321–333, Boulder, Colorado, USA, Nov. 2006

[5] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys'04)*, pp. 95–107, New York, NY, USA, Nov. 2004.

[6] B. Yahya and J. Ben-Othman, "Towards a classification of energy aware mac protocols for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 9, no. 12, pp. 1572–1607, 2009.

[7] M. Imran, A.M. Said, H. Hasbullah, "A survey of Simulators, Emulators and Testbeds for Wireless Sensor Networks," *in Proc. of the International Symposium in Information Technology 2010 (ITSim 2010)*, pp. 897 – 902, Kuala Lumpur, Malasya, Jun. 2010.

[8] G. P. Halkes and K. Langendoen, "Crankshaft: An energy-efficient mac-protocol for dense wireless sensor networks," in *Proc. of the 4th European Wireless Sensor Network (EWSN07)*, pp. 228–244, Delft, The Netherlands, Jan. 2007.

[9] M. A. Erazo, Q. Yi, "SEA-MAC: A Simple Energy Aware MAC Protocol for Wireless Sensor Networks for Environmental Monitoring Applications," in *Proc. of the 2nd International Symposium on Wireless Pervasive Computing 2007 (ISWPC '07)*, pp. 5-7, San Juan, Puerto Rico, Feb. 2007.

[10] M. Ringwald, K. Romer, "BurstMAC - An Efficient MAC Protocol for Correlated Traffic Bursts," in *Proc. of the 6th International Conference on Networked Sensing Systems 2009 (INSS 2009)*, pp. 1-9, Pittsburgh, Pennsylvania, USA, Jun. 2009.

[11] A. Köpke, M.Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," in *Proc. of the 1st International Workshop on OMNeT++*, Mar. 2008.

[12] Chipcon, "CC1100 low power" http://www. chipcon.com, 2010.

[13] Chipcon, "CC2420 low power" http://www. chipcon.com, 2010.

[14] Atmel, "AT86RF231 low power" http://www.atmel .com, 2010.

[15] E. Weingartner, H. vom Lehn, K. Wehrle, "A performance comparison of recent network simulators," in *Proc. of the IEEE International Conference on Communications 2009 (ICC 2009)*, pp. 1-5, Dresden, Germany, Jun. 2009.

[16] J. Rousselot, A. El-Hoiydi, and J.-D. Decotignie, "Low power medium access control protocols for wireless sensor networks," in *Proc. of the 14th European Wireless Conference (EW 2008)*, pp. 1–5, Prague, Czech Republic, Jun. 2008,.

[17] A. Koubaa, M. Alves, E. Tovar, "A comprehensive simulation study of slotted CSMA/CA for IEEE 802.15.4 wireless sensor networks," in *Proc. of the IEEE International Workshop on Factory Communication Systems 2006*, pp. 897 – 902, Torino, Italy, Jun. 2006.

[18] G. P. Halkes, K. G. Langendoen, "Experimental Evaluation of Simulation Abstractions for Wireless Sensor Network MAC Protocols," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, 2010.

[19] Y. Tselishchev, A. Boulis, L. Libman, "Experiences and Lessons from Implementing a Wireless Sensor Network MAC Protocol in the Castalia Simulator," in *Proc. of the IEEE Wireless Communications and Networking Conference 2010* pp. 1–6, Sydney, Australia, Apr. 2010.

[20] A. Varga, "OMNeT++," in Modeling and Tools for Network Simulation, K. Wehrle, M. Güneş, and J. Gross, Eds. Springer Verlag, September 2010, pp. 35–58.