An Approach for Modelling Wireless Sensor Networks: Focusing on the Design Concept and Energy Awareness

Michel Bakni, Guillaume Terrasson, Octavian Curea, Alvaro Llaria, Jessye Dos Santos
Univ. Bordeaux, ESTIA Institute of Technology
F-64210 Bidart, France
e-mail: {m.bakni, g.terrasson, o.curea, a.llaria, j.dossantos}@estia.fr

Abstract— **In the design stage, Wireless Sensor Network developers generally need simulation tools to save time and money. These simulators require accurate models to precisely describe the behaviors of network nodes. Nevertheless, although model complexity has grown from layered-stack to cross-level, the energy aspects are not yet well implemented. In this paper, we suggest an energy-aware cross-level model for Wireless Sensor Network. Our modelling approach allows parameters that belong to different levels to interact and affect each other. This approach is used to predict the nodes energy consumption and to estimate the lifetime of the system. First, the results obtained from the implementation of our approach will be compared with those collected from a well-known simulator, Network Simulator version 2 using a set of basic scenarios. Then, the utility of our approach in the Wireless Sensor Network design process is highlighted using detailed scenarios that cover different types of interactions.**

Keywords-Energy-aware design; Cross-level; Energy modelling; Wireless Sensor Networks.

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a set of battery-powered nodes that include sensors coupled with processing units and wireless transceivers. Since energy stored in the batteries is limited, both the autonomy of the node and the system's lifetime are impacted. Thus, energy-aware design is an important research topic and different solutions have been proposed in recent years to address the design concept from an energy point of view [1][2]. In this context, and to achieve the high-energy efficiency required by WSN to be implemented in domains like agriculture, industry, and healthcare [3], developers of WSN applications and researchers working in this field need to make the right decisions in the early stages of the design process. Consequently, and in order to deal with this challenge, simulators and emulators are widely used.

The scientific literature presents a large variety of single-level simulators, especially node-level or system-level [4]. This distribution can be traced back to the models on which the simulators were built. From what we studied, there is no WSN model designed to reveal the impacts of a given parameter on energy consumption from a cross-level perspective, and not only from a specific-level point of view [5]. Therefore, in [6], we briefly addressed the need of the cross-level design in WSN modelling. Later, we suggested a cross-level energy-aware model for WSN where interactions among model parameters were also highlighted, but only for the radiofrequency (RF) unit onboard the node [1]. In this paper, the application of the model is expanded to include other node components, precisely the processing unit and a temperature sensor, together with more detailed examples of cross-level interaction.

The structure of this paper is as follows. First, the evolution of WSN modelling is reviewed focusing on design concept and energy-awareness. Then, we propose a cross-level energy-aware model for WSN. After that, the result obtained from the implemented model is compared with results from a well-known WSN simulator: NS2. Additionally, a set of scenarios will be presented to show cross-level interactions among model parameters. This includes distance between the nodes and fragmentation threshold. Finally, we conclude with the perspective and future works.

II. WSN MODELLING AND SIMULATION

In order to better understand the main challenges related to WSN modelling and simulation in a power-aware design context, a short study on the evolution of WSN models is first provided. Throughout, the development of modelling techniques is traced. Then, WSN simulators, which were built on the previous model, are to be classified based on two characteristics: design concept and energy-awareness.

A. WSNs Models: from Layered-stack to Cross-level Design

Classical data network models, such as the Open Systems Interconnection (OSI) model or the Internet model (TCP/IP), suggest a layer-based design approach where a set of layers are stacked together. In this kind of model, the layers are defined based on their functionality, such as physical or logical connections. Each layer can only interact with the two adjacent layers in the stack using software interfaces [7]. However, in WSNs, parameters residing at different layers need to interact with each other [3]. Thus, services are to be provided across stack layers rather than a specific layer. Thereby, the traditional stack-layered model cannot answer the modelling challenges of this kind of networks [8].

For many years, several works have handled this problem by developing different WSN-oriented solutions. In [7], the authors suggest a modelling approach based on cross-layer design, where adaptivity and optimization across multiple layers of the stack is supported. A comparable cross-layered model is introduced in [9]. Furthermore, in [10], the cross-layer point of view is applied to network security, and additional considerations are included in this regard. In [11] a cross-layer concept is suggested through an infrastructure that supports a cross-layer design approach. According to the authors, the cross-layer interaction means that some data from a layer may be used as a parameter by a protocol that resides in another layer or it may affect another layer's operating process.

The suggestion of the “tier” or “level” concept is another approach that is used to answer the modelling challenges in WSN [12]. In the previous proposal, a level stands for a group of parameters that belong to different functions and features of the entire system, and not only limited to the network model. Alongside the previous methods, the solution outlined in [13] divides the model’s layers into two levels. The first level is mainly focused on non-physical parameters related to both software and application. On the other hand, the second level is concerned with hardware where the protocols are implemented, especially linking and routing, along with RF unit and sensor parameters.

This resulted in the development of a new approach for modelling WSNs: the multi-level approach. It expands the standard layered-stack model to cover not only node hardware and software, but also wireless media and parameters from the surrounding environment, such as temperature. Examples of this kind of multi-level models can be found in [14] and [15].

After that, several evolutions were suggested to develop a cross-level model starting from the multi-level approach. As an example, in [16], a description is provided for using the multiple-level model in the design and development of WSN from a cross-level perspective. However, as illustrated in [17], many proposed cross-level approaches are not used in an effective manner. Consequently, although energy modelling is a crucial issue in WSN design, this aspect is also not well implemented in the modelling stage.

B. *WSN Simulator classification: design concept and energy awareness*

In general, WSN models focus on one level of abstraction. Therefore, WSNs simulators built on these models are typically specific level [2]. Consequently, they trace parameters related to that particular level [3]. In this context, a parameter is a numeric value that characterizes one property of a given level of abstraction, such as power consumption or payload length.

For example, Network Simulator version 2, well-known as NS2 [17], is aimed at simulating network protocols. Thus, it has poor support for hardware simulation. Quite the opposite, TOSSIM [18] can precisely emulate hardware, but it provides a very abstract perspective for high-level network or routing protocols. Moreover, there are several multi-level simulators. In this category, the simulator can take into account parameters belonging to different levels simultaneously. By way of an example, Jsim [19] is a multi-level simulator dedicated to WSNs. It is multi-level because it can simulate both environment and network parameters at the same time. Finally, few simulators are classified as cross-level because they have the ability to show cross-level interaction among parameters belonging to different levels of abstraction. COOJA [20] is an example of this kind of simulator.

None of the simulators mentioned above is energy oriented. These simulators could trace energy parameters, but they were not really built purpose for that. Based on that, a level-based energy classification for WSN simulators is proposed in [4] and illustrated in Fig. 1. In this figure, simulators are first classified according to their energy-awareness: energy-oriented and non-energy-oriented. This classification is extended to include the simulator design concepts that are listed above.

Looking at energy-oriented simulators, PowerTOSSIM [21] is an extension of TOSSIM that is dedicated to the emulation of energy in hardware. There is also IDEA1 [22]. It is a multi-level energy-oriented simulator that handles three abstract levels: the wireless medium, the node and the environment.

The previous review shows that several non-energy-oriented simulators, based on multi or cross-level models have been suggested. However, as demonstrated, regarding the energy aspect, the cross-level approach is not implemented. Therefore, with existing simulators, it is difficult to analyze the influence of parameters belonging to different levels of abstraction on both WSN lifetime and total energy consumption in a given node.

III. CROSS-LEVEL APPROACH FOR ENERGY AWARE MODELLING

Considering the previously mentioned limitations of existing simulators, we propose a cross-level approach for modelling WSN in the energy-aware context. First, the main definitions on which the model is built are introduced. After that, a global overview of the suggested concept is provided. Based on that, the relationship between parameters within each node component is explained to demonstrate the cross-level interactions.

A. *General definitions*

As previously mentioned, we start by defining the principles on which the proposed concept is based.

Definitions relating to the design concept are first introduced:

- Parameter: a configurable value that represents a specific property of the level it belongs to.
- Level: an abstract design concept. It stands for a set of parameters that describes the same part of the

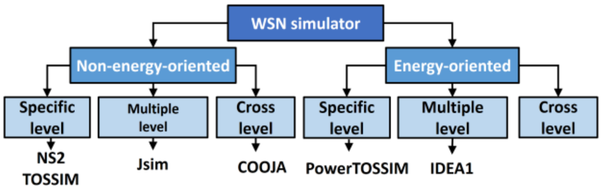

Figure 1. Energy-aware level-based classification for WSN simulators

system that is being modeled.

- Interaction: a relationship between two parameters that mutually affect each other. If the parameters belong to the same level, it is to be called an intra-level interaction. Otherwise, if the two parameters reside at different levels, the interaction is to be called cross-level.

Secondly, definitions related to the energy-aware design process are provided:

- Phase: a time period corresponds to one circuit activity, each phase is combined with a current consumption level.
- Pattern: a set of subsequent phases for several circuits. Each pattern corresponds to a particular node and it describes its behavior in terms of current consumption and time.
- Pattern frequency (F_P): the number of pattern occurrences in one second.

B. *Global Concept*

Regarding the definitions above, the global overview of our cross-level is illustrated in Fig. 2. Based on that, a WSN is divided into four separate levels of abstraction. Within each level, a set of parameters describes the level's properties. Interactions between the parameters can be cross-level or intra-level. A short description of each level is provided below.

- The Use case Level (UL) is the highest level in the concept hierarchy. It is concerned with the WSN application requirements. Parameters such as pattern frequency (Fp) or node activity sequences reside at this level.
- The System Level (SL) reflects the topological aspect of the WSN. It focuses on how algorithms and high-level protocols influence the performance of WSNs. Parameters related to the network topology, like the distance between nodes, or protocols specifications, such as fragmentation threshold belong to this level.
- The Node Level (NL) is concerned with the interactions among node software, such as the operating system, and onboard hardware, for example, the RF unit. In other words, this level focuses on the interactions between the node components. The parameters relating to the structure of node patterns, such as the order of the phases reside at this level.
- The Circuit Level (CL) is the lowest level in the model's hierarchy. It is used particularly to characterize node hardware. The modelling of the electronic circuits takes place at this level, and this is where hardware-specific parameters, such as the power levels or the supply voltage, reside. The circuit level includes an RF unit, a processing unit, a set of sensors, and a battery.

Therefore, as the four levels are stacked together, the higher the level is, the more general the parameters are. For example, parameters belonging to the UL are related to the scenario description, and they theoretically fit into any WSN application. On the other hand, CL parameters are very specific, they describe particular electronic circuits.

C. *Cross-level Interactions for Node Circuits*

In this section, the interactions, both intra-level and cross-level, between parameters that concern each circuit are explained.

As the proposed concept is energy-aware, in the following section, we focus on the energy aspects of WSN. Energy consumption in WSN is circuit-based, i.e., each circuit consumes energy independently. However, consumption is governed by the activities of the circuits which, in turn, depend on the interactions between parameters.

In the description below, next to each parameter's name, the abbreviated name of the level to which the parameter belongs will be added.

1) RF unit

The proposed concept is first applied to describe RF activities in a WSN. Fig. 3 provides an overview of all the interactions that takes in the different stages of RF activities.

In the first stage, the total number of bits to send is calculated. This includes the payload (UL) created by sensors or other applications that generates data on the node, as well as high-level protocol headers (SL) and the link layer protocol header (SL). Note that interactions at this stage are both intra-level and cross-level.

After that, the total amount of data is confronted against the fragmentation threshold (SL) identified by the wireless link protocol. If fragmentation is needed, the process will take place in this stage, and will result in two or more data frames. Next, the preamble (SL) is added to each data frame. Then, the length of each frame is calculated in term of seconds, thanks to the bitrate(s) (SL) supported by both the link protocol and the selected RF chipset.

The next step is to build the energy consumption pattern

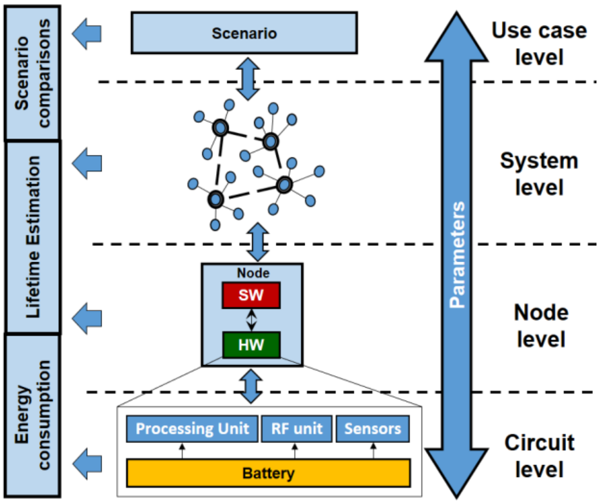

Figure 2. The Proposed Cross-level concept for Wireless Sensor Network

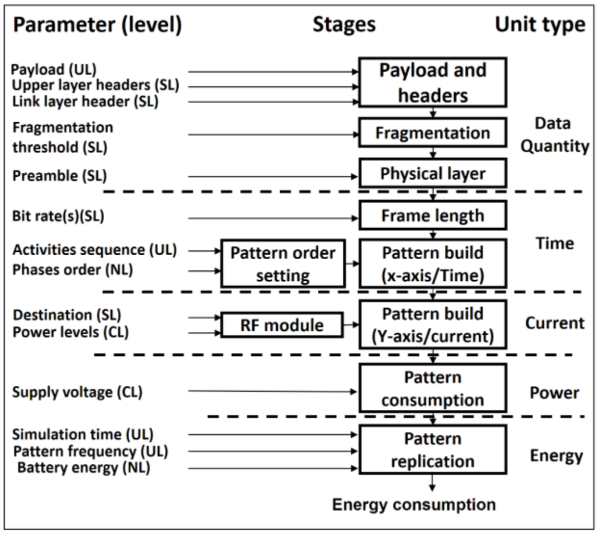

Figure 3. Cross-level design for parameters interaction in the RF unit

for the node's RF activities. Activity sequences (UL) and phase order (NL) are to be considered. The activity sequence helps to specify the actions taking place, such as sending or receiving frames. The link protocol defines the phases within each activity, as well as their order. For example, in the sending activity, the order of phases as follows: accessing the channel (ph_{acc}), exchanging data frames (ph_{exch}) and then waiting for acknowledgment (ph_{ack}).

After that, the sequence of activities is to be matched with the power levels (CL) provided by the RF unit datasheet. This includes considering the distance (SL) that has a direct impact on the power level of the sending activity. Then, considering the supply voltage of the RF circuit (CL), the energy consumed by the RF activities is calculated.

In the final stage, the total simulation time (UL), the frequency of the pattern F_p (UL), as well as the initial amount of the energy stored in the battery (NL) are taken into consideration in order to estimate the system's lifetime.

2) *Processing Unit*

All the activities relating to the processing unit are node-based. Interactions only occur with other onboard components. However, the processing unit controls the RF unit's activities in terms of wake-up and sleep times. Additionally, it manages the activities of the sensors in the same way. Thus, synchronizing the node's activities is the main task of the processing unit.

The time sequence for the processing unit pattern starts in the sleep phase, in which, all node components are in sleep state. Then, the unit wakes up and enters the active phase where all other component activities take place. Finally, the processing unit return to the sleep state. Fig. 4 shows an abstract pattern of the processing unit, where the three phases: sleep, wake-up, and active take place, respectively. As illustrated, the amount of time spent in each phase depends on cross-level and intra-level interactions between the parameters.

After that, the processing unit's pattern will be created. To achieve that, the corresponding power level (CL) for each phase is derived from the circuit datasheet.

As a result, and considering the supply voltage (CL), the energy consumption of the processing unit in one pattern is calculated. Finally, the simulation time (UL), frequency pattern (UL) and battery energy (NL) are added as well, and the effects of the processing unit's activities on the node's lifetime can be estimated. Fig. 5 displayed in the sequential stages is used to calculate energy consumption in the processing unit.

3) *Sensing Unit*

Sensors interact with the physical environment and with other nodes' components only, which means there will be no inter-node activities. Fig. 6 provides an overview of all the interactions that take place in one sensing unit. Note that multiple sensing units can exist onboard the WSN node.

When sensing a physical phenomenon (SL), the sensor measures a physical quantity and converts the measured value into an electric signal. The conversion time (CL) is specified in the sensor datasheet provided by the manufacturer. This is the time when most energy is consumed by the sensor. We consider that the result of this stage is a set of bits captured from the sensor's environment.

The sensor's activity sequence (UL) is then to be

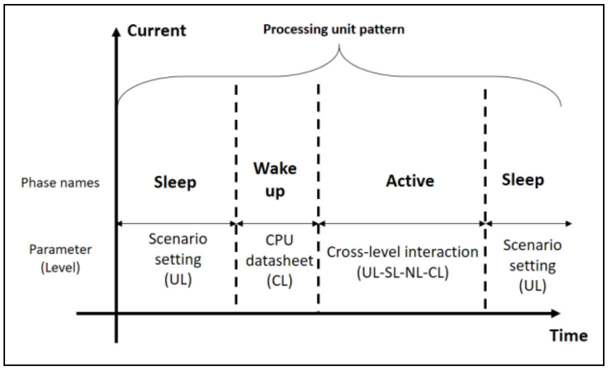

Figure 4. Pattern construction and cross-level design for parameters in the processing unit

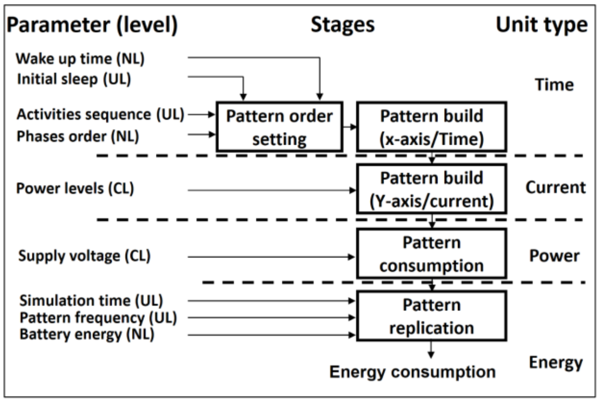

Figure 5. Cross-level design for parameters interaction in the processing unit

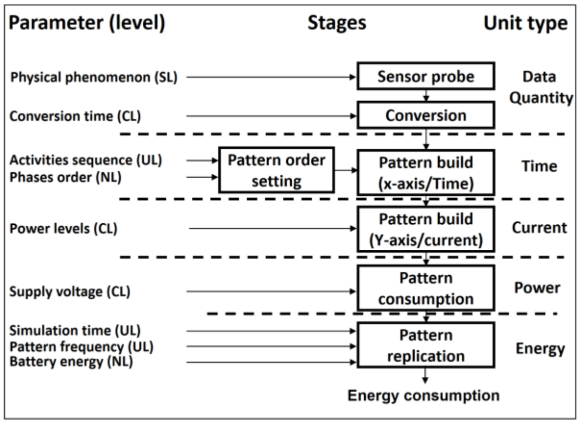

Figure 6. Cross-level design for parameters interaction in the consumption in the sensing unit

included. This includes the number occurrences in which the measurements take place in one pattern, as well as intervals between measurements. When the sensor is not sensing, it is in sleep mode. The result of this stage is the time sequence of the sensor's phases.

Next, the power level (CL) derived from the sensor datasheet is included. For each phase, there will be a corresponding consumption level. After that, the battery's voltage will be added, and energy consumption for one pattern of sensor activities is calculated. Finally, simulation time, pattern frequency and the energy level are to be included, thus, an estimation for the effects of the sensor's activities on the node's lifetime can be generated.

D. *Model design*

As illustrated previously, the proposed model is divided into four levels of abstraction. Going one step further in the modelling process, we introduce the block. It is an abstract modelling object that exists in the level's boundary. Each block includes a set of parameters. Blocks connect with each other using one of the following types of relationships:

1) Deriver-source connection: this connection links two blocks: the derivative and the source. The derivative block inherits all source attributes and behavior. Moreover, it has additional extensions that create variants of the source to serve the derivation purpose.
2) Child-parent connection: it couples two blocks: the child and the parent. The child block is part of the parent, which can have more than one child block of the same type.

The blocks can be classified based on the level in which they reside. Fig. 7 shows a block diagram for the proposed model, the levels are separated using dashed lines, and the highest level of abstraction is at the far right of the diagram. Blocks are represented using a rectangle. Additionally, block connections are shown, as well as the number of possible children or parents for each child-parent connection. Note that interactions link the model's parameters, while connections represent relationships between the blocks.

Hereafter, starting from the highest level of abstraction, there is a short description of the blocks used in the modelling process.

- UL blocks: UL includes one or more scenario blocks, each of which represents a full scenario. A given scenario includes scenario-level parameters such as the scenario time as well as blocks from the topology level like the network topology block. In the modelling hierarchy, the scenario block is the highest.
- SL blocks: this is the level where information relating to the whole system is parameterized. Examples of parameters are the number of nodes and the distance separating them. This level includes two topology-related blocks: network topology and wireless medium blocks. The topology block includes a set of node blocks, each of which represents one node in the network, and all those nodes are node-level blocks. The wireless medium characterizes the communication channel.
- NL blocks: these reside at the node-level and cover the parameters at that level, such as node position. There are three blocks at this level: node, type, and pattern blocks. The node block stands for the physical node in the modeled system. Each node is associated with a pattern and type blocks. The type serves a design requirement: heterogeneity. A type includes circuit blocks. Additionally, a type block can be shared among a set of nodes. The pattern block represents the node's periodic behavior. If two nodes share the same pattern, they will behave identically, in terms of activities (sending, sensing, etc.), but the consumption associated with each activity might be different based on other parameters, such as node positions. The type block also includes a battery, processing unit, RF unit, and sensors block. These are all circuit-level blocks.
- CL blocks: these describe the hardware components of the node. The battery block is a circuit-level block. It represents the physical battery and includes energy parameters, such as the battery's nominal values. The circuit is a circuit-level abstract block, i.e., other blocks can be derived from it, they are the processing unit, RF unit and the sensors, each of which describes one specific component. These blocks include parameters related to the physical circuits. The parameters can usually be obtained from the circuit's datasheets, like the power level. Finally, the RF unit block is associated with the protocol block. The protocol block describes the wireless link protocol used by the RF unit. It covers parameters related to the link protocol such as bit rates.

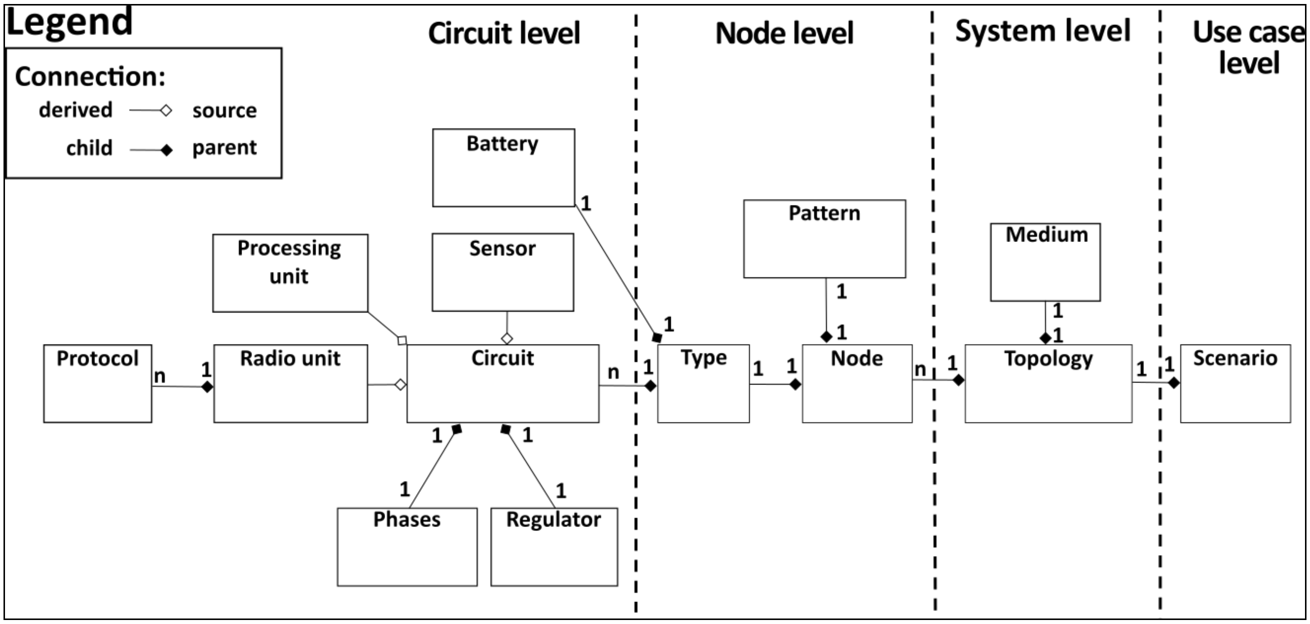

Figure 7. Block diagram for the proposed model

As mentioned above, the model supports network heterogeneity. This is a property describing the ability to simulate heterogeneous systems where different types of nodes can exist in the same scenario [23]. The connection between the type and the node blocks at the node level answers heterogeneity. Each node includes one type block, but they can vary in its components and protocols. Note that the pattern block is not a part of the type, which means that two identical nodes can have different energy consumption patterns, where each consumes energy based on its parameters. The distance to the destination parameter is an example of that. Different distances will create various patterns, thus, non-equal energy consumption among the nodes, although they might be identical in terms of components.

As we have seen in the WSN modelling, this is the first model to include circuit regulators. Although the benefits of this addition were not directly explained in this paper, the transferring from one unit to another when representing consumption, precisely from current to energy, was done thanks to the regulators. The regulator modelling allows to distinguish the voltage levels for each point on the node, thus, facilitating the consumption calculation and make it more accurate.

IV. APPLICATION OF OUR MODEL

A. General settings

Our model is implemented using Matlab. Then tested scenarios will be proposed. These are designed for two purposes. First, presenting interactions between parameters, both cross-level and intra-level. Second, comparing the results obtained from the proposed model with those of a well-known simulator, namely NS2.

All the scenarios take place in an open area where there are two wireless nodes named Node A and Node B. These settings for the scenarios where selected to show the interactions between the parameters. Thus, simple two-node scenarios were developed. Periodically, Node A sends a fixed-length payload to Node B through the wireless medium using its RF unit.

Each node applies a TCP/IP network model. The implementation of the protocols starts at the network layer where Internet Protocol version 4 (IPv4) [24] has been chosen. The Internet Control Message Protocol (ICMP) [25], which is an integral part of IPv4, is used to create echo messages when necessary. Consequently, when the IPv4 module receives a data packet, it is going to send the same data back to the original source. In all scenarios, the length of IPv4 and ICMP headers are 20 and 4 bytes, respectively.

Next, the link-layer parameters are to be set. Two different protocols are to be used, these are: IEEE 802.15.4 as [26] and IEEE 802.11a is used [27]. The energy specification for the IEEE 802.15.4 module is derived from the CC2420 transceiver (Texas Instruments) [28]. For IEEE802.11a the specifications will be derived from an implemented chipset named HDG204 (H&D wireless) [29]. Table I shows the settings for the two wireless link protocols. For each scenario, the considered time used to calculate energy consumption is 100 seconds, and it begins after initializing the nodes.

The previous settings are general. They are to be implemented in all the scenarios. However, in the following sections, additional settings are to be added will be explicitly mentioned.

B. Payload and pattern frequency

Pattern frequency and payload size are UL parameters. In this set of scenarios, we trace the cross-level effects of the two on the consumed energy, which is a CL parameter. First, we suggest a scenario using only the RF unit and we compare the obtained results with those from NS2. The reason behind this choice is the need to have comparable results, i.e., NS2 does not include models for other hardware components. Second, the scenario will be extended to include other hardware components, namely a processing unit and a sensor. This will highlight the contribution of the proposed modelling approach.

In order to illustrate the cross-level interaction, we go back to Fig. 7. When the payload size (UL) is modified, the length of the data frame to be sent (CL) by the RF unit will be changed. This, in turn, will impact the pattern structure (NL) and will have an influence on the total energy consumed by the node (CL), as well as the system's estimated lifetime (UL). The cross-level interaction for the frequency pattern can be traced in the same way. However, in this case, the change of F_P (UL) will directly affect the pattern structure (NL), without passing by the circuit level. Then, the chain interaction will follow the same way as the payload size interactions.

1) *Scenarios with only RF unit*

To effectively trace payload size and F_P, the following settings are to be added to the general settings. First, the two nodes will reside 10m apart. Second, both IEEE 802.11a IEEE 802.15.4 are to be used. For each protocol, three different values of F_p: 0.1, 1 and 2 Hz, and ten values for the payload length ranging between 10 to 100 bytes are used. Each scenario requires a combination of the three parameters previously mentioned. As a result, there are 30 scenarios to be run for each wireless link protocol. Table II summarizes the setting for the scenarios with only RF unit included.

These scenarios are configured both in NS2 and in the proposed model implemented in Matlab. The obtained results are the energy consumed by different activities of the RF unit. These activities are categorized into 4 phases:

- Access phase (ph_{acc}): RF unit tries to access the wireless channel.
- Exchange phase (ph_{exch}): RF unit sends or receives data frames.
- Acknowledge phase (ph_{ack}): RF unit sends or receives acknowledgment frames.
- Sleep phase (ph_{slp}): The RF unit is in sleep state.

In these phases, the cross-level interaction between parameters is taking place, i.e., the energy consumed in each phase is a result of interplay between parameters related to different levels. For example, in the exchange phase, the consumed energy depends on of the headers' lengths, payload length, and bit rate, which belongs to the following levels: UL, SL, and NL, respectively.

The results obtained from the implementation of the previously described scenarios in NS2 and in Matlab using our model can be found in Table III. The upper part shows the obtained results from IEEE 802.11a and the lower part for IEEE 802.15.4. Each part is further divided into two subparts, corresponding to scenarios with 10 and 100 bytes of the payload length, respectively.

For the two protocols, when comparing obtained results from NS2 and our proposed model, differences and similarities can be found as follows. The energy consumed in ph_{acc} or in ph_{ack} are identical and there is a slight difference in the energy consumed in ph_{slp}. In ph_{exch}, the difference is notable, but stable and this is due to different interpretations of the link protocol specifications. For example, in our model, the ICMP header is considered to be part of the data packet, contrary to NS2 where this header is added to the data packet later.

Fig. 8 shows the pattern obtained from the proposed model. The left side of the figure is dedicated to IEEE 802.11a and the right side to IEEE 802.15.4. Different values of F_p are displayed, namely 0.1, 1, and 2 Hz. For each of these values, a set of corresponding errors is provided. Each of these is also related to a simulation where the payload length is 10, 50, or 100 bytes.

Next, the relative error between the results obtained from NS2 and our model is to be calculated. For each scenario, the relative error is calculated as follows. The value obtained from out model is subtracted from the value obtained from NS2, and the result is to be divided by the latter. In all the simulation results, the relative errors obtained from IEEE 802.11a are greater than those of the corresponding scenarios of IEEE 802.15.4. This difference can be explained by unplanned and non-periodic radio activities that appear periodically in NS2 IEEE 802.11a simulations. These activities have a fixed duration regardless of F_p and the payload length. Each of these activities appears as a single pulse of transmission or reception causing an additional energy consumption of around 5 µJ and 2 µJ, respectively.

Finally, as illustrated in Fig. 9, the value of the relative error between NS2 and the proposed model did not exceed 3.5%. Based on that, we can consider the proposed model is validated with NS2. Although not shown in this scenario, our proposed WSNs model also allows adding energy consumption phases for other hardware on the node, such as the processing unit or sensors. As a result, an accurate pattern can be constructed representing precisely the real consumption of the node contrary to NS2 that has a poor support for hardware, as mentioned before.

TABLE I. WIRELESS LINK PROTOCOL SETTINGS

Parameter	IEEE 802.11a (HDG204)	IEEE 802.15.4 (CC2420)
Bitrate [bps]	12 M	250 K
Carrier Sense Mechanism	Pure CSMA/CA	CCA-ED
Transmitter power [mW]	725	52
Receiver Power [mW]	220	59
Sleep Power [mW]	0.2	0.06

TABLE II. GENERAL SETTINGS OF THE RF UNIT ONLY SCENARIOS

Parameter	Value
Number of the nodes	2
Node positions	(10,10), (10,20) [m]
Scenario duration T_{Sce}	100 [s]
Pattern Frequency F_p	0.1, 1, 2 [Hz]
Payload length	10, 20, …, 100 [Byte]
Link protocol	IEEE 802.11a, IEEE 802.15.4

2) Complete node Scenario

A typical WSN node includes other components in addition to the RF unit, such as processing unit and sensors. Together, they serve the node functionality. In order for the components to serve the node, they consume a considerable amount of energy that cannot be neglected. For example, energy profiling in [30] shows that the RF unit is responsible for 62% of the node consumption, where the sensors and the microcontroller share the remaining 38% of the energy consumed. Another study shows different energy profiles based on the scenario setting, the consumption of the RF unit varying between 7 to 65% of the total consumed energy, while the other components are responsible for the remaining consumption [31].

NS2 is not capable of tracking the energy consumed in sensors and the microcontroller. However, the proposed model is able to do that. This is simply achieved by extending the consumption pattern to include new phases for the other components.

In this scenario, the setting of the RF only scenario will be implemented considering the following extensions. We assume that the node includes a microcontroller and a sensor. The microcontroller is PIC18F4620 [32] and the sensor is TMP102 [33], a temperature sensor. The scenario is still the same, except for the following: The node wakes up after 0.5 seconds from the pattern start. Then, the sensor captures a value directly after the node is up it sends the data through the RF unit. Finally, it goes back to sleep again without waiting for the response. In addition to that, the size of the payload is only 2 bytes since the sensor creates a 12-bit sample.

TABLE III. ENERGY CONSUMPTION OF THE PHASES IN DIFFERENT SCENARIOS (F_P = 1 Hz)

Simulation	**Consumed energy per phase [µJ]**				
	Sleep	**Exchange**	**Access**	**Ack.**	**Total**
IEEE 802.11a					
10 Bytes					
Matlab	51.03	50.08	23.63	199.959	324.69
NS2	51.03	55.76	23.63	199.950	330.37
100 Bytes					
Matlab	51.03	106.79	23.63	199.935	381.38
NS2	51.03	112.46	23.63	199.927	387.04
IEEE 802.15.4					
10 Bytes					
Matlab	18.89	145.73	39.07	59.78	263.47
NS2	18.88	152.84	39.07	59.78	270.47
100 Bytes					
Matlab	18.89	465.37	39.07	59.44	582.77
NS2	18.88	472.42	39.07	59.45	589.82

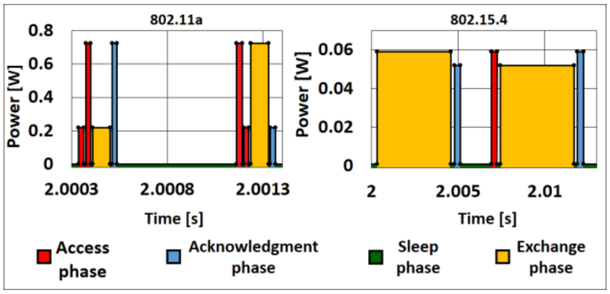

Figure 8. Energy consumption patterns for different wireless link protocols (Node B, Payload length = 100 Bytes, F_p = 1 Hz)

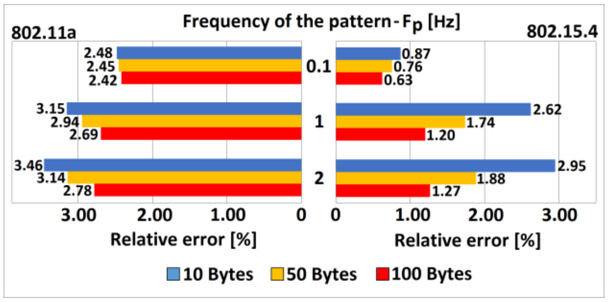

Figure 9. Relative error between the proposed model and NS2

All the circuits are supplied with 3.3 V. The electrical characteristic for the RF unit will not be changed. Regarding the processing unit and sensor, Table IV provides supply current in the active and sleep phases. Note that the wake-up duration of the processing unit lasts for 1 µs, during which we consider the supply current to be identical to the active phase.

In Fig. 10, a pie chart is shown for a per-component energy consumption percentage for the node A. The energy consumed by the processing unit is 60% of the total consumption and sensor consumes only 4% of the total energy. However, energy consumed by the RF unit is 36%. However, the selecting the hardware components, such as those using different processing units or the scenario settings, like activity sequences can significantly impact the energy profile.

In addition to that, the model is also capable of tracing the energy consumption down to the circuit level, i.e., the consumption of the phases for each circuit. Table V provides results obtained from the sending node, the energy consumed by the processing unit is divided into three phases: Sleep,

TABLE IV. ELECTRICAL CHARACTERISTICS FOR THE PROCESSING UNIT AND THE SENSOR IN THE COMPLETE NODE SCENARIO

Phase	**TMP102**	**PIC16F4620**
Sleep [µW]	1.65	0.33
Active [µW]	280.5	4290

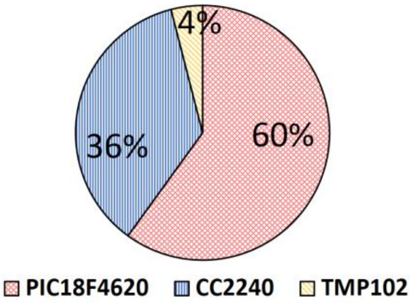

Figure 10. Per-component energy consumption profile for the complete node scenario

wake up and active, and consumption in the active phase has the higher value. For the sensor, the energy consumed is categorized into two phases: sleep and active. Following the same consumption trend, most consumption takes place during the active phase.

The proposed model answers the needs of WSN designers and researchers. Regarding the first group, the model provides a way to combine different circuits and protocols. This is useful in the application design stage where decisions relating to the construction of the node are to be made. On the other hand, the model can be used in the field of research as well, because it can show the cross-level interactions, and this is interesting for energy-related studies.

The suggested model is extensible, i.e., new blocks can be added to support new functionalities. For example, a wave propagation model can be added at the system level. Additionally, an energy harvester unit can extend the model at the circuit level. However, the model activities and the corresponding phases need to be added to the pattern at the node level as well.

C. Distance and power levels

Distance between nodes has an impact on energy consumption in the RF module. Manufacturers of the RF chipsets provide a range of transmission power levels. The further the destination is, the higher the transmission power level selected. The proposed model provides a way to trace this cross-level interaction.

The cross-level interactions are illustrated using arrows in Fig. 11. First, a distance between two nodes changes, it is a system-level parameter. This, in turn, interacts with the power level selection in the RF unit at the circuit level. Through intra-level interaction, the activity sequence in the RF unit is affected. This is illustrated using a blue arrow on the same figure. After that, the cross-level chain interaction reaches the node pattern at the node level. Finally, energy consumption in the battery at the circuit level, and thus, the system's lifetime at the scenario level are affected.

TABLE V. ENERGY CONSUMPTION FOR THE NODE COMPONENTS IN ONE PATTERN FOR THE COMPLETE NODE SCENARIO. (PAYLOAD = 2 BYTES)

Processing unit: PIC16F4620			
Phase	**Sleep**	**Wake-up**	**Active**
Energy [μJ]	0.359	4.81 x 10^{-3}	143.03
RF Module: CC2240			
Phase	**Sleep**	**Access**	**Exchange**	**Ack**
Energy [μJ]	1.31	17.8	40.75	24.48
Sensor: TMP102				
Phase	**Sleep**	**Active**		
Energy [μJ]	1.8	7.69		

In order to show the previous cross-level interactions, the general settings will be extended as follows. First, the payload length will be 20 bytes in all scenarios. Second, the distance between the two nodes will change to 25, 50, 100, 200, and 400 meters, respectively. Those values were intentionally selected to cover the whole power level range proposed in the RF unit's datasheet.

Fig. 12 shows the obtained values for energy consumption of the RF unit exchange phases from the distance and power level scenario. As the distance between nodes increases, the energy consumption in the exchange phase increases linearly. This can be explained from the chipset datasheet specification. As the distance rises above a particular threshold, the model automatically changes the transmission threshold to use the next value. This change has an impact on the energy consumed by the node. For example, when the distance between the two nodes changed from 25 to 400m, the corresponding consumed energy in the transmission phase changed from 60 to 120 μJ.

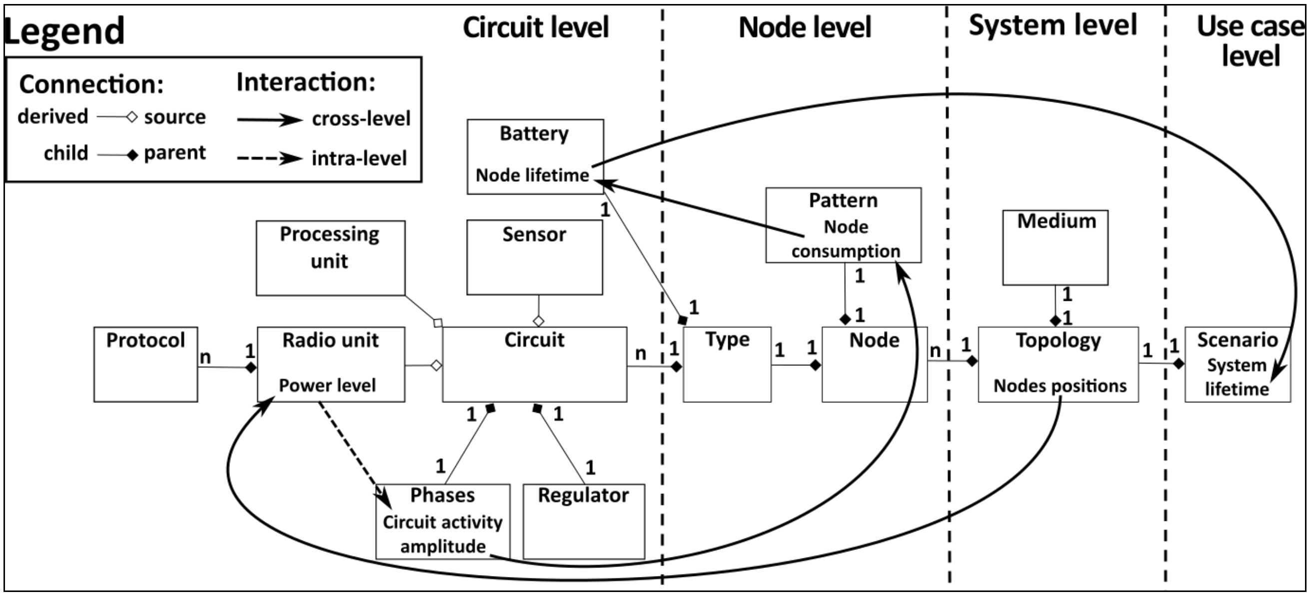

Figure 11. Block diagram for the proposed model

D. *Fragmentation threshold*

Fragmentation is a mechanism to divide Protocol data units; i.e., frames or packets, into two or more units that are shorter in length. Fragmentation is used when WSN applications contain a constraint on the size of data units. For example, IEEE 802.15.4 has a maximum fragmentation threshold set to 127 bytes, but any lower value down to one, can be set. Fragmentation threshold is a system-level parameter which impacts the energy consumption of the RF unit at the circuit level. The proposed model can be used to show the previous cross-level interaction.

In this case, the cross-level interactions take place as follows: first, the fragmentation threshold will be changed. It is a system-level parameter. This change will affect the protocol object, which is attached to the RF unit object, and both are circuit-level parameters. These changes will affect the pattern structure at the node level, which, in turn, affects the energy consumption in the node, which is a circuit-level parameter. In the end, the system's lifetime will be affected by this cross-level chain interaction.

In practice, the settings of the complete node scenario will be implemented. However, the following changes will take place. The scenario will include sending a payload of 100 bytes, and the fragmentation threshold is set to {40, 60, 80, 100} bytes, respectively. TMP102 is not used. It will always remain in the sleep state, thus, its consumption is constant in all the scenarios and it will be ignored. On the other hand, the consumption of the RF unit and the processing unit is changed as a response to the fragmentation setting, and these changes are to be traced. The node wakes up after 0.5 seconds from the pattern start and begins sending the data packets.

Fig. 13 presents the results obtained from the fragmentation scenario. The energy consumption by the RF unit phases is illustrated according to the fragmentation threshold. The consumption of access, exchange, and acknowledgment phases drop significantly after the first test. Energy consumption in the sleep phase changes slightly.

These results can be explained with the help of the number of frames sent in each scenario: they are 15, 4, 3, and 2, respectively. This sharp drop in the number of the sent frames is due to the effect of another system-level parameter, namely protocol headers. The used protocols need to add headers to the payload. It is 33 bytes in total, and that leaves only 7 bytes for the payload when the fragmentation threshold is set to 40 bytes. However, the payload size is up to 27 bytes when the threshold is 60 bytes. Thus, 100 bytes of payload need only 4 frames to be completely sent. In general, these results show that energy consumption in the node decreases as the fragmentation threshold increases.

V. CONCLUSION AND FUTURE WORK

In this paper, a cross-level energy-aware modelling approach for WSN is proposed. This approach was applied by mean of several scenarios to reflect how parameters from different levels can interact and affect energy consumption. The suggested model is built on the assumption that WSN node activities can be described in a pattern, which is periodically repeated. A pattern consists of sequential phases belonging to all the circuits composing the node.

Moreover, the obtained results show the model's ability to provide energy consumption at different levels of abstraction. This includes the total energy consumed by each node, the consumption of one specific circuit, as well as that of a particular phase. Furthermore, this capability will be extended in future work to include the estimation of the system's lifetime.

The characterization of energy consumption requires handling time on a very large scale. On the one hand, it is necessary to calculate energy consumption in the node pattern accurately using microsecond scale. On the other hand, the system's lifetime tends to be expressed in years. Our approach can answer those issues because the simulation time is a linear gain between the pattern energy and the total amount of consumed energy.

However, for the moment, our approach cannot integrate non-repetitive or unpredictable activities. For example, whenever the network topology changes during the scenario, routing protocols should be activated to find new routes and these activities will consume energy in an irregular way.

Currently, we are developing a WSN simulator based on the proposed approach. The simulator will be designed for both researchers and WSN application developers. In parallel, we are developing a test bench based on physical nodes. The objective is to compare the simulation results with measurement from real applications, when the same scenarios are implemented in both environments.

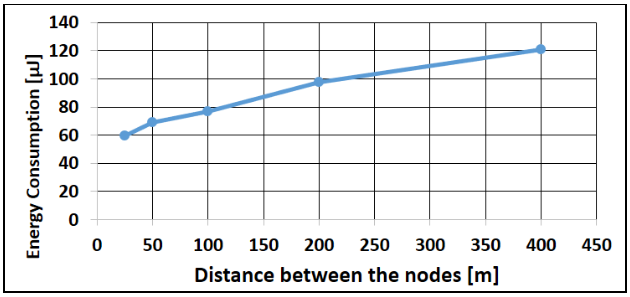

Figure 12. Energy consumption of the RF unit exchange phase in the distance and power level scenario

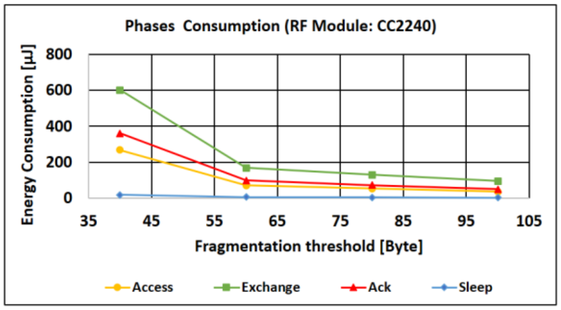

Figure 13. Per-phase energy consumption from the fragmentation scenario for the RF unit

ACKNOWLEDGMENT

The work presented in this paper has been financially supported by the Regional Council of New Aquitaine (Call for Projects 2016 funds), as part of the OUDINI research project.

REFERENCES

[1] M. Bakni, G. Terrasson, O. Curea, A. Llaria, and J. Dos Santos, “Energy-aware Cross-level Model for Wireless Sensor Networks,” SENSORCOMM, 2019. pp. 46-51. ISBN: 9781612087443.

[2] R. Chéour, M.W. Jmal, O. Kanoun, and M. Abid, “Evaluation of simulator tools and power-aware scheduling model for wireless sensor networks,” IET Computers & Digital Techniques, vol. 11, no. 5. 2017, pp. 173-182, doi: 10.1049/iet-cdt.2017.0003.

[3] J. Yick, M. Biswanath, and G. Dipak. “Wireless sensor network survey,” Computer networks, vol. 52, no. 12 .2008. pp. 2292-2330. doi: 10.1016/j.comnet.2008.04.002.

[4] M. Bakni, L. Chacón, Y. Cardinale, G. Terrasson, and O. Curea, “WSN Simulators Evaluation: An Approach Focusing on Energy awareness,” International journal of wireless & mobile networks (IJWMN), 2020, 11 (6), pp. 1-20. doi: 10.5121/ijwmn.2019.11601.

[5] J. Haase, J. M. Molina, and D. Dietrich, “Power-aware system design of wireless sensor networks: Power estimation and power profiling strategies,” IEEE Transactions on Industrial Informatics, vol. 7, no. 4, 2011. pp. 601-613. doi: 10.1109/TII.2011.2166793.

[6] M. Bakni, G. Terrasson, O. Curea, A. Llaria, and J. Dos Santos, “A Cross-level model for power-aware Wireless Sensor Networks design,” 13ème Colloque National du GDR SOC2, 2019.

[7] H. Zimmermann, “OSI Reference Model - The ISO Model of Architecture for Open Systems Interconnection,” in IEEE Transactions on Communications. vol. 28, no. 4. 1980. pp. 425-432. doi: 10.1109/TCOM.1980.1094702.

[8] A. J. Goldsmith and S. B. Wicker, “Design challenges for energy-constrained ad hoc wireless networks,” IEEE Wireless Communications, vol. 9, no. 4, 2002, pp. 8-27. doi: 10.1109/MWC.2002.1028874.

[9] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, 2002. pp. 393-422. doi: 10.1016/S1389-1286(01)00302-4.

[10] W. Charfi, M. Masmoudi, and F. Derbel, “A layered model for wireless sensor networks,” 2009 6th International Multi-Conference on Systems, Signals and Devices, 2009, pp. 1-5. doi: 10.1109/SSD.2009.4956693.

[11] I. Korkmaz, O. Dagdeviren and M. E. Dalkilic, “A 2-hop coloring-based collision free infrastructure design for Wireless Sensor Networks,” 2016 HONET-ICT, Nicosia. 2016, pp. 65-69, doi: 10.1109/HONET.2016.7753421.

[12] E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, P. Pavón-Mariño, and J. García-Haro, “Simulation tools for wireless sensor networks,” International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS05), Society for Modeling & Simulation International, 2005, pp. 1-8, ISBN: 9781622763504.

[13] M. Kuorilehto, M. Kohvakka, M. Hännikäinen, and T.D. Hämäläinen, “High abstraction level design and implementation framework for wireless sensor networks,” International Workshop on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS 2005), Lecture Notes in Computer Science, 2005, pp. 384-393. doi: 10.1007/11512622_41.

[14] M. Korkalainen, M. Sallinen, N. Kärkkäinen, and P. Tukeva, “Survey of wireless sensor networks simulation tools for demanding applications,” 2009 Fifth International Conference on Networking and Services, 2009, pp. 102-106, doi: 10.1109/ICNS.2009.75.

[15] W. Du, D. Navarro, F. Mieyeville, and F. Gaffiot, “Towards a taxonomy of simulation tools for wireless sensor networks,” Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, 2010. doi: 10.4108/ICST.SIMUTOOLS2010.8659.

[16] M. Imran, A.M. Said, and H. Hasbullah, “A survey of simulators, emulators and testbeds for wireless sensor networks,” 2010 International Symposium on Information Technology, 2010, pp. 897-902. doi: 10.1109/ITSIM.2010.5561571.

[17] T. Issariyakul and E. Hossain, “Introduction to Network Simulator 2 (NS2),” in Introduction to Network Simulator NS2. Springer US, pp. 21-40, 2012, doi: 10.1007/978-0-387-71760-9_2.

[18] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and scalable simulation of entire TinyOS applications,” Proceedings of the 1st international conference on Embedded networked sensor systems, 2003, pp. 126-137, doi: 10.1145/958491.958506.

[19] A. Sobeih, J.C. Hou, L-C. Kung, N. Li, H. Zhang, W-P. Chen, H-Y. Tyan, and H. Lim, “J-Sim: A simulation and emulation environment for wireless sensor networks,” IEEE Wireless Communications, vol. 13, no. 4. pp. 104-119. 2006, doi: 10.1109/MWC.2006.1678171.

[20] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Cross-Level sensor network simulation with COOJA,” Proceedings. 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 641-648. doi: 10.1109/LCN.2006.322172.

[21] E. Perla, A.Ó. Catháin, R.S. Carbajo, M. Huggard, and C. McGoldric, “PowerTOSSIM z: Realistic energy modelling for wireless sensor network environments,” Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 2008, pp. 35-42. doi: 10.1145/1454630.1454636.

[22] W. Du, D. Navarro, F. Mieyeville, and I. O’Connor, “IDEA1: A validated system C-based simulator for wireless sensor networks,” Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011, pp. 825-830. doi: 10.1109/MASS.2011.95.

[23] M. Bakni, L. Chacón, Y. Cardinale, G. Terrasson, and O. Curea. “Methodology to Evaluate WSN Simulators: Focusing on Energy Consumption Awareness”. 6th International Conference on Computer Science, Engineering and Information Technology (CSEIT-2019), 2019. pp. 331-351, doi: 10.5121/csit.2019.91327.

[24] J. Postel, “RFC 791, Internet Protocol”, 1981. doi: 10.17487/RFC0791.

[25] J. Postel, “RFC 792, Internet Control Message Protocol”, 1981. doi: 10.17487/RFC0792.

[26] “IEEE Standard for Information technology -- Local and metropolitan area networks -- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs),” in IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 1-320, 2006. doi:10.1109/IEEESTD.2006.232110.

[27] “IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: High Speed Physical Layer in the 5 GHz band,” in IEEE Std 802.11a-1999, pp. 1-102, 1999. doi: 10.1109/IEEESTD.1999.90606.

[28] "2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver", SWRS041c, Texas Instruments, 2019.

[29] "HDG204-WiFi 802.11b+g System in Package", 1451 - Rev. PA5 02/2011 , H&D Wireless AB, 2011.

[30] G. Terrasson, R. Briand, S. Basrour, V. Dupé, O. Arrijuria, "Energy Model for the Design of Ultra-Low Power Nodes for Wireless Sensor Networks", Procedia Chemistry, vol. 1, no. 1, 2009, pp. 1195-1198. doi: 10.1016/j.proche.2009.07.298.

[31] M. Navarro, Y. Li and Y. Liang, "Energy profile for environmental monitoring wireless sensor networks," 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), 2014, pp. 1-6. doi: 10.1109/ColComCon.2014.6860416.

[32] "PIC18F2525/2620/4525/4620 Data Sheet 28/40/44-Pin Enhanced Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology", DS39626E, Microchip Technology, 2008.

[33] "TMP102 Low-Power Digital Temperature Sensor With SMBus and Two-Wire Serial Interface in SOT563", SBOS397H, Texas Instruments, 2018.