

# Cross-layering in an Industrial Wireless Sensor Network: Case Study of OCARI

Khaldoun Al Agha<sup>1</sup>, Gérard Chalhoub<sup>2</sup>, Alexandre Guitton<sup>2</sup>, Erwan Livolant<sup>3</sup>, Saoucène Mahfoudh<sup>4</sup>,  
Pascale Minet<sup>4</sup>, Michel Misson<sup>2</sup>, Joseph Rahmé<sup>1</sup>, Thierry Val<sup>3</sup>, Adrien Van Den Bossche<sup>3</sup>

<sup>1</sup> LRI, Bât 490 Université Paris Sud 11, 91405 Orsay Cedex, France

Email {khaldoun.alagha, joseph.rahme}@lri.fr

<sup>2</sup> LIMOS-CNRS, Clermont Université, Les Cézeaux, 63177 Aubière Cedex, France

Email {gerard.chalhoub, alexandre.guitton, michel.misson}@sancy.univ-bpclermont.fr

<sup>3</sup> Université de Toulouse, UTM, LATTIS, 1 place G. Brassens, 31703 Blagnac, France

Email {erwan.livolant, thierry.val, adrien.vandenbossche}@lattis.univ-toulouse.fr

<sup>4</sup> INRIA, Rocquencourt, 78153 Le Chesnay Cedex, France

Email {saoucene.mahfoudh, pascale.minet}@inria.fr

**Abstract**—Wireless sensor networks are adapted to monitoring applications. Specific solutions have to be developed for industrial environments in order to deal with the harsh radio conditions and the QoS (quality of service) requirements of industrial applications. In this paper, we present the main protocols used in the OCARI project, and we describe their use of cross-layering techniques. We show how it enables us to improve the performance of the network. For each protocol, we give a performance evaluation of its main characteristic.

**Index Terms**—cross-layering techniques, wireless sensor network, industrial networks.

## I. INTRODUCTION

In many industries, there is a strong need to monitor factories, installations or the security of the human teams. Wireless sensor networks are well adapted to those applications and have an additional advantage: they are not expensive to deploy. However, it is an important challenge to have these networks operate for years without replacing the device batteries. Moreover, the presence of metallic equipments affect the propagation conditions of wireless signals, and it is an issue to propose various qualities of service. That is why specific solutions have to be designed in order to address the industrial needs.

OCARI<sup>1</sup> is a project funded by the French national research agency with seven academic and industrial partners. The goal of the project is to develop efficient new protocols for a industrial wireless sensor network. An OCARI network consists of several networks operating on independent radio channels. Each network is managed by a PAN (personal area network) coordinator.

The OCARI project uses a large panel of protocols that belong to several layers. As the context of the OCARI

project is targeted to specific industrial applications, the key concern of the developed protocols is not to be generic, but rather determinist and energy-efficient [1]. Soon in the project, it appeared that the protocols require high interactions with each other.

Cross-layering techniques are keys issues in many WSNs [2]. In this paper, we study the specific case of the OCARI project and show how the cross-layering techniques we use can lead to major improvements of the whole network behavior. In this paper, we present the main protocols of OCARI and we discuss their use of cross-layering techniques. Table I presents the protocols we describe in the following, and the layers they interact with. These protocols are described from the lower layers to the upper layers: MAC (medium access control) first, then NWK (network) and finally APS (application).

TABLE I.  
CROSS-LAYERING PROTOCOLS IN OCARI.

Protocol	Layer	Cross-layer
MaCARI	MAC	NWK
MBC	MAC	NWK
MSSS	MAC	APS
SERENA	NWK	MAC+APS
EOLSR	NWK	MAC+APS
OMBCE	APS	MAC+NWK

The remaining of this paper is organized as follows. Section II describes the MAC protocol of OCARI, called MaCARI, and shows its basic routing features. Section III describes a coding algorithm used by MaCARI to increase the number of coordinators per network it can deal with. Section V describes the node activity scheduling protocol of OCARI, called SERENA. To operate in good conditions, such a protocol has to interact with both the MAC layer and the application layer. Section VI describes the energy efficient routing protocol EOLSR. Section IV proposes to synchronize the sensor sampling frequency with the MAC activity. Section VII describes how a battery model can be taken into consideration by most

This study has been partly funded by the ANR OCARI project <http://www.lri.fr/ocari/>

Corresponding author: Pascale Minet, INRIA, Rocquencourt, 78153 Le Chesnay Cedex, France

<sup>1</sup>OCARI stands for Optimization of Communications in an Ad hoc Reliable Industrial network.

of the other protocols for energy efficiency. Finally, we conclude our work in Section VIII.

## II. MAC PROTOCOL FOR OCARI (MACARI)

MaCARI [3] is the MAC protocol of the OCARI project. It assumes the IEEE 802.15.4 physical layer. In this section, we describe this protocol, and show that it is a cross-layering protocol by itself.

### A. Description of MaCARI

The network has a tree topology and is constituted by coordinators, end-devices and a PANC (Personal Area Network Coordinator). A coordinator and the devices associated to it are called a star. Coordinators are the only devices that allow other devices to join the network, and that is according to three topology parameters, as specified in the ZigBee documentation [4]: (1)  $Lm$  which is the maximum depth of the tree, (2)  $Rm$  which is the maximum number of children coordinators per coordinator and (3)  $Cm$  which is the maximum number of children per coordinator.

MaCARI is the MAC layer protocol of OCARI. MaCARI divides time into four periods that constitute a global cycle [5]. At the start of the first period, called the synchronization period  $[T_0; T_1]$ , the PAN coordinator initiates the propagation of a synchronization beacon which is propagated along the tree by all the coordinators to reach all the entities of the network. The beacon indicates to all the devices of the network the time interval during which they are allowed to communicate. During the second period, called the scheduled activities period  $[T_1; T_2]$ , each coordinator is allocated a time interval during which it manages the activity of its star [6]. At the end of the activity period of a star, the coordinator exchanges information, when necessary, with its parent coordinator in a guaranteed time slot. During the third period, called the routing period  $[T_2; T_3]$ , only the coordinators are activated in order to route information using the EOLSR routing protocol. During the fourth period, called the inactivity period  $[T_3; T_0]$ , all the network devices are sleeping.

An end-device has no routing capacities and can only communicate with its coordinator during the time interval allocated to its star. Coordinators have two routing strategies depending on the traffic priority: (1) for the high priority traffic, a hierarchical routing protocol is applied to communicate along the tree during the scheduled activities period, (2) if it is a low priority traffic, frames are routed using the EOLSR protocol [7] during the routing period.

### B. Cross-layering in MaCARI

As explained previously, coordinators in the OCARI network have two routing protocols. The hierarchical routing protocol is used to relay high priority traffic along the tree and the EOLSR routing protocol is used to exchange low priority traffic during the routing period.

The higher priority traffic is exchanged during guaranteed time slots. Guaranteed time slots are allocated for each coordinator to communicate with its parent. This time allocation requires a precise synchronization for the transmissions. Since it is MaCARI that takes in charge this time allocation, to be able to apply the hierarchical protocol in the appropriate time interval, EOLSR and MaCARI have to be perfectly synchronized. To avoid this synchronization between the MAC layer and the network layer, we added the hierarchical routing protocol functionalities to MaCARI. Thus, MaCARI takes in charge, as well, the creation of the network and the allocation of the hierarchical logical addresses which are NWK layer addresses.

## III. MACARI BEACON COMPRESSION (MBC) WITH ASSOCIATION CONTROL

The duration of the synchronization period  $[T_0; T_1]$  in MaCARI is an important issue, as all the network entities are awake during this period. To ensure that MaCARI is energy-efficient, it is required to make this period as short as possible. MBC (MaCARI beacon compression) is a coding technique that allows to compress the list of coordinators in the beacon. Recall that this list takes most of the space in the beacon format. Thus, MBC can reduce the beacon transmission time. In this section, we describe how MBC can code and decode the list of coordinators in the beacon.

### A. Description of MBC

During  $[T_0; T_1]$ , each coordinator of the network sends a beacon containing the list of all the coordinators. Storing the list in each beacon has two drawbacks: (1) it increases the size of each beacon, and therefore its duration and in turn the duration of the whole period  $[T_0; T_1]$ , and (2) it limits the total number of coordinators to approximately 50 due to the limited size of 127 bytes for a 802.15.4 physical layer frame.

MBC is a coding technique that allows to compress the list of coordinators in the beacon. It also features a fast decoding technique, so that coordinators do not need to run long computations to decompress the list of coordinators.

MBC is based on the way addresses are allocated. In the following, we describe the hierarchical address allocation mechanism of ZigBee, which is also used in OCARI. Then, we describe the MBC coding and decoding algorithms.

1) *Hierarchical address allocation*: OCARI uses the hierarchical address allocation mechanism described in [4]. It assumes that every nodes of the network knows three network parameters  $Lm$ ,  $Rm$  and  $Cm$ .  $Lm$  is the maximum depth of the tree,  $Rm$  is the maximum number of children coordinators that can be associated to a coordinator and  $Cm$  is the maximum number of children that can be associated to a coordinator.

The PAN coordinator is assigned address 0. Upon association, a coordinator of depth  $d$  in the tree is attributed an address range  $[a; a + \text{cskip}(d - 1) - 1]$ . The coordinator chooses for itself address  $a$ . Addresses from  $a + \text{cskip}(d - 1) - C_m + R_m$  to  $a + \text{cskip}(d - 1) - 1$  are reserved for the association of end-devices. The  $i$ -th coordinator that associates with the coordinator is attributed the address range  $[a + 1 + (i - 1) \times \text{cskip}(d); a + i \times \text{cskip}(d)]$ . Function  $\text{cskip}$  is defined in the following way:  $\text{cskip}(d) = 1 + C_m(L_m - d - 1)$  if  $R_m = 1$ , and  $\text{cskip}(d) = (1 + C_m - R_m - C_m \times R_m^{L_m - d - 1}) / (1 - R_m)$  otherwise. It ensures that each node is allocated a unique address and that each address allocated to the PAN coordinator can be used. An example of a topology is shown on Fig. 1, pointing out the addresses allocated to each network entity.

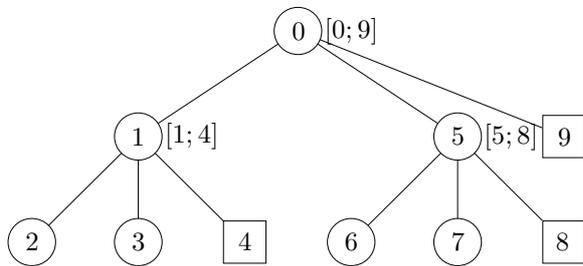


Figure 1. The hierarchical address allocation mechanism of ZigBee ensures that addresses are unique and that no address is wasted, for  $L_m = 2$ ,  $R_m = 2$  and  $C_m = 3$ .

2) *Coding algorithm:* It is important to notice that address allocation is topology-aware. Thus, the existence of an address  $a$  implies the knowledge of other addresses, such as the addresses of all the coordinators on the branch between the root of the tree (the PAN coordinator) and the node of address  $a$ . This is the key concept behind MBC. Instead of transmitting in a beacon the set of all the coordinator nodes of the tree, MBC transmits only the set of leaves of the tree. For example, on the topology shown on Fig. 1, MBC only sends  $\{2, 3, 4, 6, 7, 8, 9\}$  instead of sending  $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ . In this simple example we managed to save two addresses in the beacon.

Figure 2 shows the maximum number of nodes in the list (without and with MBC) as the number of nodes in the network varies. The maximum number of leaves in the tree is computed over 1000 randomly generated topologies, with the following network parameters:  $L_m = 10$ ,  $R_m = 3$ ,  $C_m = 3$ . As it can be seen on the figure, MBC significantly reduces the number of nodes in the list, and therefore the size of the beacon.

3) *Decoding algorithm:* To compute the set of all coordinators from the list of the addresses of the tree leaves, MBC applies the following algorithm to each address in the beacon. Let us suppose that MBC knows the address of a leaf  $x$  and the network parameters  $L_m$ ,  $R_m$  and  $C_m$ . MBC computes the addresses of all the possible children of the PAN coordinator. From this list, it can deduce which coordinator is the ancestor of  $x$  (it is the coordinator of the list having the largest address smaller

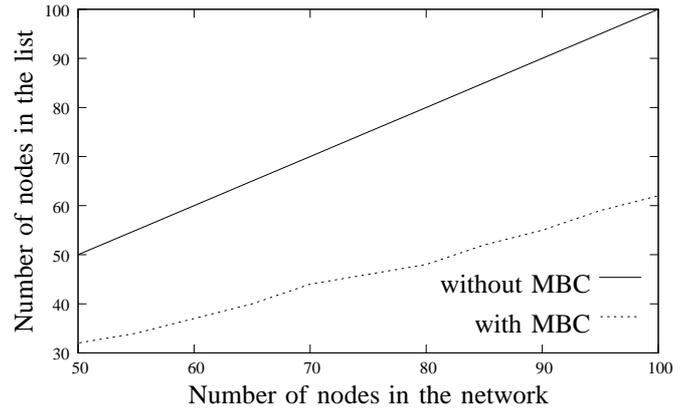


Figure 2. MBC impact on the beacon size.

than  $x$ ). Then, MBC computes the addresses of all the possible children of the ancestor, and determines the next ancestor. This process is repeated until all the ancestors of  $x$  are computed. This process requires  $\mathcal{O}(C_m L_m)$  per node, plus the time required to merge all the lists (note that a sorted merge can be used).

4) *Association control:* As explained in III-A.2, MBC is not able to compress efficiently trees that have many leaves. In order to reduce the number of leaves nodes in the tree we adopt an association control mechanism<sup>2</sup>. Each coordinator includes the number of its children coordinators in the beacon it propagates. During the scan procedure, the node requesting an association sends its request, when possible, to the coordinator that has no children coordinators. Thus, the association control mechanism prioritizes the creation of deep tree topologies.

### B. Cross-layering in MBC

MBC is an algorithm that is applied at the MAC layer, in order to reduce the beacon size and therefore to reduce the duration of the synchronization period of MaCARI. It is a cross-layering algorithm according to two features:

- it uses the knowledge of network addresses in order to infer the topology,
- it suggests the use of a specific association control mechanism.

These two features are traditionally out of the scope of a MAC protocol.

## IV. MAC-SENSOR SAMPLING SYNCHRONIZATION (MSSS)

In this chapter dedicated to the MAC-Sensor Sampling Synchronization (MSSS) protocol, we focus on a cross-layering technique: the temporal synchronization between sensors (APS Layer) and MaCARI (MAC Layer).

In the OCARI network, which is a WSN, nodes are mostly composed of sensors [8]. Two categories of sensors could be distinguished in accordance to their data transmission mode (see Chap. 2.2 of [9]). Event based

<sup>2</sup>The association control mechanism will be studied in future work to optimize the duration of the synchronization period of MaCARI.

sensors are used to monitor unpredictable physical phenomena. These sensors, called asynchronous, may send an alarm at any time. On the other hand, synchronous sensors sample and transmit data regularly.

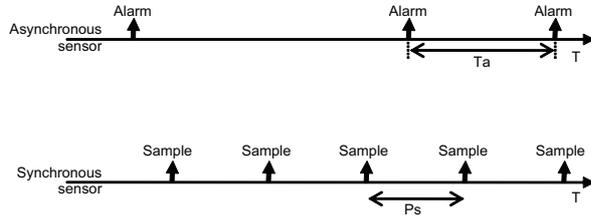


Figure 3. Asynchronous and synchronous sensor transmission chronogram.

In a first approach, synchronous sensors have their own clock according to the Nyquist-Shannon theorem. The sensor samples at least two times faster than the physical phenomenon changes and less than the bandwidth capacity of the transmission channel dedicated to it. This synchronous clock period  $P_S$ , see Fig. 3, should generally be shorter than the duration  $T_a$  between two potential alarms performed by asynchronous sensors.

#### A. Description of MSSS

1) *Relevance of cross-layering between MaCARI and sensors:* As we have seen in section II, MaCARI aims at offering energy savings and a QoS in terms of deterministic channel access. MaCARI is therefore based on a slotted mechanism with slotted CSMA/CA and Guaranteed Time Slot (GTS) for data transmission inside each star [6].

In this part, we focus on the MAC layer and the sensors at the APS layer which are associated with the end-devices of the OCARI network. According to the OSI model, the layers MAC and APS do not have the same clock reference. Since MaCARI has a clock for its slotted behavior and sensors should sample with a frequency corresponding only to physical measurement constraints, the two layers are thus asynchronous.

On Fig. 4, we can see on the lower chronogram the simplified functioning of MaCARI for each star of the OCARI network [3]. A superframe is delimited by the interval  $P_B$  between two beacons (B) emitted by the coordinator (when there is no inactivity period). In a superframe, several GTSs can be allocated to several end-devices by the coordinator. On Fig. 4, there is only one GTS allocated at the end of each superframe according to the IEEE 802.15.4 standard [10].

On the upper chronogram of Fig. 4, the autonomous behavior of a sensor is presented. The sensor has to transmit a sample with a period  $P_S$  according to its own clock. We can notice that there is a delay  $D$  between the measure and the medium access instant inside the GTS allocated to the sensor. This delay is variable and can be in the worse case equal to  $D_{max}$ .

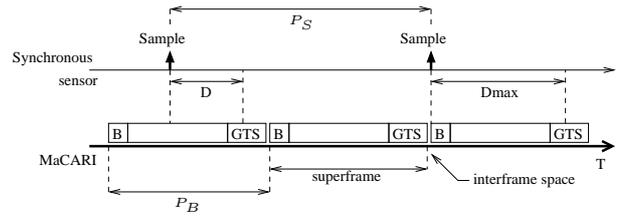


Figure 4. Case of two different clock chronograms.

We implemented the intra-star MaCARI mechanism on the CC2420 hardware<sup>3</sup>. The results presented on Fig. 5 concern the GTS transmission mode in slotted CSMA/CA. Results have been obtained with the regular beacons IEEE 802.15.4 MAC-layer, with a superframe duration of 245.76 ms ( $15.36 \times 2^4$  ms) and no inactivity period. We varied the number of end-devices in the star from 1 to 4. The period  $P_B$  is equal to 245.76 ms. Each end-device had to emit 1, 3 or 5 packets with 30 bytes of MAC useful payload. Each line represents the maximal duration from the beacon to the end of the last transmission. The solid line corresponds to the example of Fig. 4. The duration of one packet emission corresponds to the duration  $D_{max}$  described above. We can notice that  $D_{max}$  is equal to 206 ms.

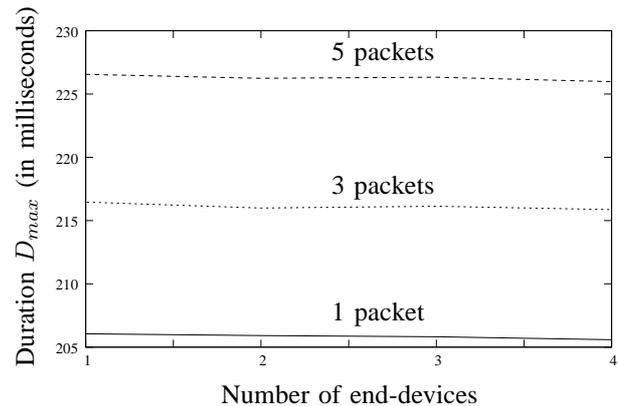


Figure 5. Transmission duration for  $n$  packets for each end-device in GTS transmission mode.

This desynchronization involves a penalty in regards of the transmission delay between sensors and receivers (e.g. smart actuators, data gathering system...). Moreover, sensors require a low sample frequency (i.e.  $T_a > P_S$  according to the applications focused on by the OCARI project). Therefore, the delay  $D$  is mostly due to the desynchronization between the APS and the MAC layers.

2) *MSSS protocol details:* The MSSS protocol consists in designing a cross-layer approach between the MaCARI layer and the APS layer. We assume that the sensor and MaCARI have then the same clock. Considering a sample from the sensor which requires a GTS at the transmission state, MaCARI requests the sensor to sample before the start of the GTS (see Fig. 6). Of course, the measure delay (sample control, digitization, data frame construction, data frame transfer between sensor and MAC, ...) should be

<sup>3</sup><http://www.one-rf.fr/en/b2400zb-tiny250.html>

taken into account before the start time of the GTS.

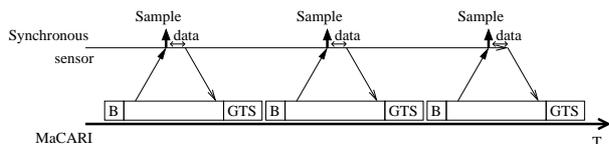


Figure 6. Call for data from MaCARI to the synchronous sensor.

If a sensor requires a lower sampling frequency, it is not necessary to use one GTS every superframe. The concept called  $GTS(n)$ , presented in [3], offers the possibility to have one GTS out of  $2^n$  superframes.

For asynchronous sensors, the command sent by MaCARI to the synchronous sensors can also be used. In this case, the signal is a pop-type command, ordering the sensor to produce a data. The data is encapsulated in a MAC frame and sent in the next GTS. It is also possible to send the message as soon as possible by using CSMA/CA. This option offers a possibility to send the message without waiting for the GTS but without guarantee. This pop-type mechanism is completely opposite to the regular method which consists, for the sensor, in sending the data to the MAC layer as soon as it has some data to transmit (push-type mode). In fact, in a pop-context, the sensor has to keep its data until the MAC-layer requests for them. In a push-context, the MAC-layer has to test if the buffer contains data.

*B. Cross-layering in MSSS*

On the MaCARI demonstration prototype, the traffic generation for the MAC layer has been simplified thanks to MSSS. The MAC micro-controller controls the sensor, e.g. a binary/analogical sensor via a digital/analogical I/O. The unique clock reference on the micro-controller (MaCARI) also controls the activation of the sensors on the prototype. With the protocol MSSS, transfer delays are reduced since the sensor is directly synchronized by the MAC layer. This cross-layer approach does not respect the OSI-model as the data take a shortcut through the protocol stack.

V. SCHEDULING ROUTER NODES ACTIVITY (SERENA)

In wireless sensor networks, it can be expensive, difficult or even impossible to renew the energy of battery operated nodes. That is why energy efficiency is required for each protocol running in such networks [1]. Since the sleeping state of a node is the state consuming the least energy, making nodes sleep provides substantial benefits in terms of energy. However, as it is wasteful to transmit messages to a sleeping node, nodes activity must be coordinated. In OCARI, SERENA (SchEDule RoutEr Nodes Activity) is the protocol in charge of scheduling router node activities.

A. Description of SERENA

SERENA is based on a three-hop coloring algorithm: two nodes that are 1, 2 or 3-hop neighbors have different colors. Hence, a color is reused four hops away. Since interferences are assumed to be limited to 2 times the transmission range (two hops), at least two hop coloring is needed to avoid message collisions. As we target unicast transfer with an immediate acknowledgement, three hop coloring is necessary to prevent collisions. Then, slots are granted to nodes according to their color. A node is awake during the slots granted to its color to transmit its messages and during the slots granted to the colors of its 1-hop neighbors to receive their messages. It sleeps the remaining time.

This coloring algorithm is distributed and localized: nodes color themselves according to their priority order [11]. The priority of a node is equal to the couple (number of neighbors up to three hops, node identifier). A node, with the highest priority among its uncolored neighbors up to three hops, colors itself with the smallest color available in its neighborhood up to three hops. Colors are represented by positive integers.

Figure 7 depicts the number of colors used in a network whose number of nodes ranges from 25 to 150 nodes and for densities of 10 and 15. The density represents the average number of neighbors per node. We notice that the number of colors is less than the number of nodes for all studied topologies, except for 25 nodes with a density greater than or equal to 10. For 150 nodes with density 10, the benefit reaches 375 %. The activity period contains 40 slots with SERENA instead of 150 without. In a dual way, if the cycle duration is given by the application, the nodes can sleep 110 slots longer with SERENA.

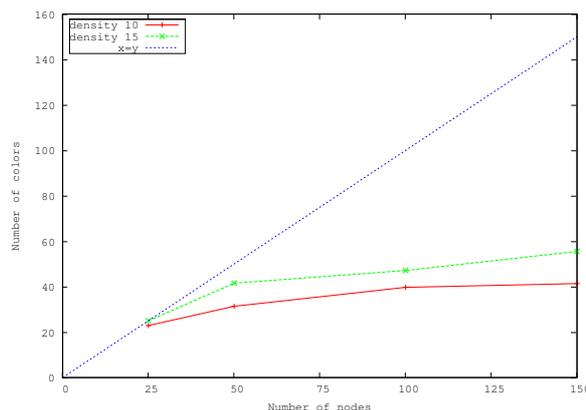


Figure 7. Impact of the density on the number of colors.

We can notice that SERENA has no impact on the loss rate, insofar as nodes activities are scheduled by the colors. We have shown in [11] that the increase in network lifetime is followed by an increase in the amount of user data delivered. SERENA can increase the end-to-end delay if colors are not assigned carefully. We show in section V-B.2 how to assign colors in order to reduce the time needed either to collect or to disseminate information.

B. Cross-layering in SERENA

In order to improve its performance and to better adapt to the real wireless environment and application requirements, SERENA uses cross layering with two layers: the MAC layer and the APS layer.

1) *Cross-layering with the MAC layer:* There are two main functionalities that benefit from cross-layering:

- Slot allocation according to the node color: SERENA provides to the MAC layer its color and the colors of its one hop neighbors. The color of the node informs the MAC layer in which slots it must be awake to send its messages. However, the colors of its one hop neighbors are needed to know in which slots the node must be awake to receive messages.
- Detection of color conflicts created by node mobility, late node arrivals or unidirectional links: Normally, three hop coloring does not create a color conflict. However, in a real wireless environment, because of the existence of unidirectional links, node mobility, late arrivals nodes or instability of propagation conditions, colors conflict can occur.

A color conflict occurs between two nodes of the same color when they prevent the intended destination from receiving the message or acknowledgement that was intended to it. For instance, as depicted on Fig. 8, nodes A and B have chosen the same color (denoted by integer 1) because they are neither 1, nor 2 nor 3 hops neighbors. Nodes B and C are 1-hop neighbors because of the symmetric link between them, represented by a plain line in Figure 8. However, the unidirectional link from A to B, represented by a dotted line in Figure 8, can lead to a collision on B between a message sent by A and an acknowledgement sent to B.

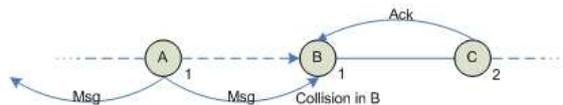


Figure 8. Example of a color conflict between A and B causing a collision in B.

Figure 9 depicts the number of conflicts and the number of colors in a network, where in average, 20% of the links of a node are unidirectional. We notice that the number of conflicts is much smaller than the number of colors and that, even in the presence of a large number of unidirectional links in the network.

As color conflicts are unavoidable because of unidirectional links, node mobility, and late arrivals in the network, a mechanism to detect and solve these conflicts is introduced. In order to increase the reactivity to color conflicts, the MAC layer detects and notifies any color conflict to SERENA. If the sender does not receive the acknowledgement of its frame, it delays its transmission by a random backoff. This backoff removes the synchronization between the two conflicting nodes and allows the receiver to receive an undesired frame (i.e., a frame that a node is not supposed to receive). The MAC layer

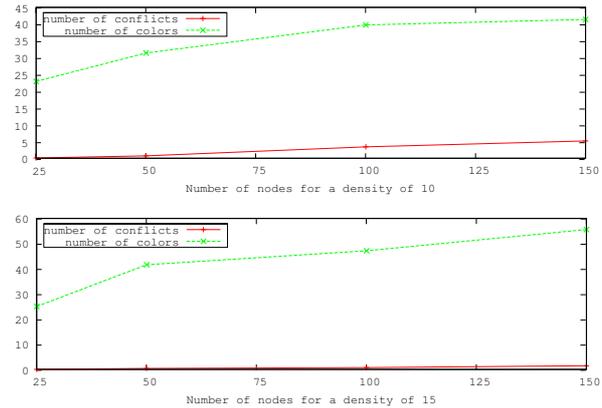


Figure 9. Number of color conflicts.

can determine the two conflicting nodes identified by the addresses in the frames. Upon notification of a color conflict by the MAC layer, SERENA reassigns the color of the conflicting node with the smallest priority.

2) *Cross-layering with the APS layer:* The knowledge of application requirements can be used by SERENA in two ways:

- The application scenario can be used to reduce the time needed to collect data or disseminate data. For instance, in a data gathering application, colors can be granted according to the depth in the data gathering tree. Hence, slots can be activated according to an order that increases the priority of transmissions from the leaves of the tree to the root. In a data dissemination, the reverse order is used. This can be easily integrated in SERENA.
- The number of slots allocated to a color can vary dynamically depending on the traffic of the nodes having this color.

VI. ENERGY-EFFICIENT ROUTING PROTOCOL (EOLSR)

The aim of the EOLSR (Energy-efficient OLSR) protocol is to minimize the energy consumed by the end-to-end transmission of a packet and avoid nodes with a low residual energy to maximize network lifetime.

A. Description of EOLSR

EOLSR [7] is an energy efficient routing protocol based on OLSR. Like OLSR [12], it consists of two main functionalities:

- Neighborhood discovery: Each node acquires the knowledge of its one-hop and two-hop neighborhood by exchanging periodic *Hello* messages.
- Topology dissemination: Each node maintains topological information about the network obtained by means of *Topology Control (TC)* messages broadcast in the whole network.

To avoid nodes with low residual energy and since in OLSR intermediate nodes in paths are MPRs, MPRs

selection is changed to take into account the nodes residual energy. These new MPRs are called EMPRs (Energy Multi Point Relay). To achieve that, first, energy information is included in *Hello* messages. Then, EMPRs are selected according to the residual energy of themselves and their one hop neighbors. To avoid frequent route changes and assure load balancing, the selection of EMPRs changes only when the topology changes or the residual energy decreases over a given threshold. Then, EOLSR selects the route with the lowest cost. This cost takes into account the energy dissipated by an end-to-end transmission which includes the energy dissipated by 1) the transmission and the reception of a packet, 2) the overhearing and 3) the radio interferences. EOLSR can be used in different types of wireless networks, only the computation of the cost function is changed.

*B. Cross layering with EOLSR*

The performances of any routing protocol are improved by the knowledge of the wireless environment and the application requirements. Indeed, the routing protocol can be optimized using this knowledge that can be provided by cross layering. From the point of view of routing, cross layering exists with both the MAC layer and the application layer.

1) *Cross layering with the MAC layer:* The MAC layer knows the power of the received signal. This information is used by the routing protocol to build routes. Only links with a good quality (i.e. the received power is higher than a given threshold) are considered in computing routes. This improves route stability and increases the message delivery rate. Moreover, the MAC layer is the first to detect the appearance of a new link and the disappearance of an existing link. EOLSR uses this information to react more promptly to topology changes. By consequence, the information provided by the MAC layer improves on the one hand the EOLSR reactivity to topology changes by an early detection of broken links or new links and on the other hand routes stability by using only good quality links.

2) *Cross layering with the Energy management:* Energy management of a node knows its residual energy. EOLSR uses this information to select the EMPRs which take into account the residual energy of candidate nodes as well as the residual energy of its one hop neighbors. This information allows on the one hand to avoid nodes with a small amount of residual energy and on the other hand to enable a load balancing by reselecting the EMPRs if the residual energy of one of them decreases over a given threshold.

3) *Cross layering with the Application:* In some applications, like for instance data gathering running in wireless sensor networks, we can distinguish two types of nodes: sensor nodes and strategic nodes. By strategic node, we mean a supervision node, a sink node... In such applications, maintaining a route to any other node in the network is not required. Indeed, it is sufficient to maintain a route to any strategic node in the network.

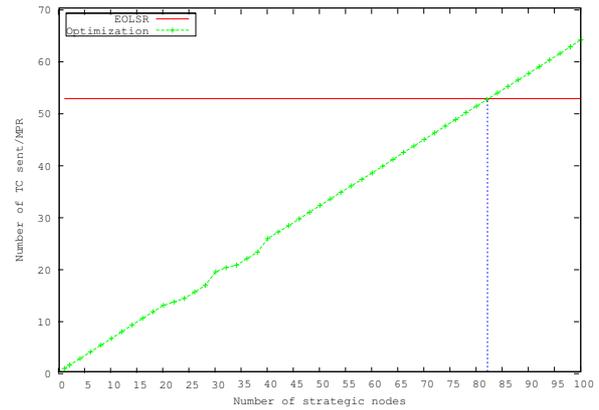


Figure 10. Number of TCs transmitted by a MPR.

Figure 10 depicts the average number of TCs transmitted by a MPR node per TC period as a function of the number of strategic nodes. We consider a network with 100 nodes and density 10. As shown in figure 10, this optimization improves the EOLSR performance as long as the number of strategic nodes is smaller than the number of MPRs, which is 82 in this example. This number 82 is obtained as the x-coordinate of the intersection between the EOLSR and the optimization curves. Hence, this optimization considerably reduces the overhead in the network.

We can easily conclude that cross layering with the application enables important benefits in terms of bandwidth, memory, processing capacity and energy.

VII. ON-LINE MODEL FOR BATTERY CAPACITY ESTIMATION (OMBCE)

Most of the protocols of the OCARI project need a battery model to be implemented in a sensor node. This model must determine the battery residual energy and keep this information up-to-date, in order to give a consistent view of the nodes energy. In this section, we present an  $L_k$ -based model that updates the energy of the battery every  $L_k$  duration.

Our model uses the model proposed in [13], which has the following form:

$$\alpha = \int_0^L i(\tau) d\tau + 2 \sum_{m=1}^{\infty} \int_0^L i(\tau) e^{-\beta^2 m^2 (L-\tau)} d\tau, \quad (1)$$

where

- $\alpha$  is the total charge consumed by the battery during time  $L$ ,
- $\beta$  is the rate at which the active charge carriers are replenished at the electrode surface.

$\alpha$  and  $\beta$  are two parameters determined experimentally. The first term of equation (1) represents the consumption of the load having a current  $i(\tau)$  during a period  $[0, L]$ . The second term accounts for the charges that exist in the battery, but were unavailable at the electrode surface at time  $L$ .

### A. Description of OMBCE

To describe our model, we use the following parameters:

- $\delta_k$  is the time duration of a state  $k$ ,
- $I_k$  is the constant current value during state  $k$ ;  $I_k$  is considered null in the sleep state,
- $\alpha_k$  is the apparent charge consumed by the sensor node during state  $k$ ,
- $L_k$  is the duration defined as follows:
 
$$\begin{cases} L_k = \delta_k + \delta_{k+1} & \text{if state } k+1 \text{ is a sleep state} \\ L_k = \delta_k & \text{otherwise,} \end{cases}$$
- $C(k)$  is the battery charge level at state  $k$ ,
- $C$  is the theoretical capacity of the battery.

The model in [13] uses a set of consecutive loads as an input and tests whether the battery can survive the load. In addition, the total charge  $\alpha$  consumed during  $L$  ( $L = \sum \delta_k$ ) is the sum of the charges consumed in all the states forming  $L$  ( $\alpha = \sum \alpha_k$ ). Therefore, we reduce the parameter  $L$  to a duration  $L_k$ , permitting the node to calculate the amount of charge consumed during  $L_k$  and to update its energy dynamically. Hence, our model is on-line, since the battery charge level is updated dynamically. Moreover, in our model, a node can only be in the following states: reception, transmission or sleep, and the state duration is supplied by the MAC layer. However, the long computation time of the model makes it hard to implement on a sensor node. This issue is addressed in the following.

In a sensor, each state consumes a current with a small variation over the duration of a state. Therefore, we consider without loss of generality, that the current can be approximated by the average of all the current values during a state, *i.e.*, the current consumption of any state can be approximated by a constant current consumption value. Moreover, in equation (1),  $i(\tau)$  is the current load consumed for the duration  $L$ . Since we want the current consumption for a duration  $L_k$ , we let  $i(\tau) = I_k$  in (1), where  $I_k$  is the current consumed during state  $k$ . Therefore, the constant  $I_k$  can be brought out of the integrals, thus (1) reads:

$$\alpha_k = I_k \times \left[ \delta_k + 2 \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L_k - \delta_k)} - e^{-\beta^2 m^2 L_k}}{\beta^2 m^2} \right]. \quad (2)$$

To obtain an exact representation of the current at a given state, we can approximate the original signal using linear regression for small interval durations. This can be done using Heaviside function [14].

#### 1) Energy consumption per state (without recovery):

In this part, we simplify the form of equation (2).

Let us consider the following case that applies to OCARI communications. A node in state  $k$  is active for a duration  $\delta_k$  while transmitting a packet. After the transmission, the node remains active, switches to state  $k+1$ , and starts listening to the medium. The node does not pass to the idle state after state  $k+1$ . The sensor must now update its energy, taking in account the energy of the transmission and of the reception. We consider that

state  $k$  consumes a constant current  $I_k$  and state  $k+1$  consumes a current  $I_{k+1}$  (see Fig. 11).

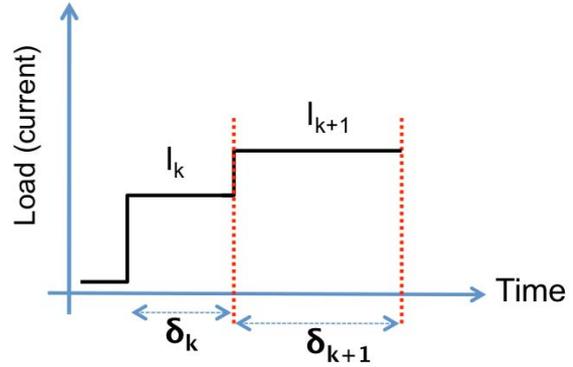


Figure 11. Load for states  $k$  and  $k+1$ .

Using (2) for state  $k$ , we have  $L_k = \delta_k$  and in this case,  $\alpha_k$  represents the charge used during this state. After replacing  $L_k$  by  $\delta_k$  in (2), we obtain,

$$\alpha_k = I_k \times \left[ \delta_k + 2 \sum_{m=1}^{\infty} \frac{1 - e^{-\beta^2 m^2 \delta_k}}{\beta^2 m^2} \right]. \quad (3)$$

Similarly with state  $k+1$ ,  $L_{k+1} = \delta_{k+1}$ , thus (2) becomes,

$$\alpha_{k+1} = I_{k+1} \times \left[ \delta_{k+1} + 2 \sum_{m=1}^{\infty} \frac{1 - e^{-\beta^2 m^2 \delta_{k+1}}}{\beta^2 m^2} \right]. \quad (4)$$

If  $C(k-1)$  is the battery charge level before state  $k$ , then after state  $k+1$ , the charge available in the battery is:

$$C(k+2) = C(k-1) - (\alpha_k + \alpha_{k+1}),$$

where

- $C(k+1)$  is the charge available in the battery after state  $k$ ,
- $C(k-1)$  is the battery charge level before state  $k$ ,
- $\alpha_k$  represents the apparent charge lost during state  $k$ ,
- $\alpha_{k+1}$  is the apparent charge used during state  $k+1$ .

Hence, having the battery theoretical capacity  $C$ , the energy in the battery is updated after each  $L_k$  by subtracting the value of the apparent charge lost during  $L_k$ . Equations (3) and (4) give the charge consumed in every state, however, they cannot be implemented in a node with low computational ability. Therefore, we still have to simplify them.

We begin by rewriting (3) in the following form:

$$\alpha_k = I_k \times \left[ \delta_k + \frac{2}{\beta^2} \times \left( \sum_{m=1}^{\infty} \frac{1}{m^2} - \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 \delta_k}}{m^2} \right) \right]. \quad (5)$$

Recall that:

$$\sum_{m=1}^{\infty} \frac{1}{m^2} = \frac{\pi^2}{6}. \quad (6)$$

Using Taylor series for the logarithm function, we get:

$$\sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 \delta_k}}{m^2} = \ln\left(\frac{e^{\beta^2 \delta_k}}{e^{\beta^2 \delta_k} - 1}\right) - \frac{1}{2 \times (e^{\beta^2 \delta_k})^2} - \dots \quad (7)$$

Note that since  $\beta > 0$  and  $\delta_k > 0$ ,  $\beta^2 \times \delta_k > 0$ .

There exists  $n \in \mathbb{N}^*$  such that  $ne^{n\beta^2\delta_k} > n$ . Finding the reciprocal of this inequality, we get:

$$\frac{1}{ne^{n\beta^2\delta_k}} < \frac{1}{n}$$

For a constant  $\delta_k$ , the value of  $1/(ne^{n\beta^2\delta_k})$  has an upper bound of  $1/n$ . In addition,  $1/n$  shrinks to 0 for large  $n$  values, which makes the terms beyond  $1/(3 \times (e^{\beta^2\delta_k})^3)$  in (7) small for large  $n$ . Thus, these terms can be neglected.

Substituting (6) and (7) in (5), we obtain:

$$\alpha_k \approx I_k \times \left[ \delta_k + \frac{2}{\beta^2} \times \left( \frac{\pi^2}{6} - \ln\left(\frac{x}{x-1}\right) + \frac{1}{2x^2} + \frac{1}{3x^3} \right) \right], \quad (8)$$

with  $x = e^{\beta^2\delta_k}$ .

2) *Energy consumption per state (with recovery)*: The recovery effect is when the battery recovers some of its energy, when discharged with currents separated by idle time slots. Since (1) considers this effect, the battery apparent charge lost during a state  $k$  followed by an idle period can be calculated using (1). In this case, the duration  $L_k$  is equal to the duration of the active state  $\delta_k$  and the idle state  $\delta_{k+1}$  (see Fig. 12).

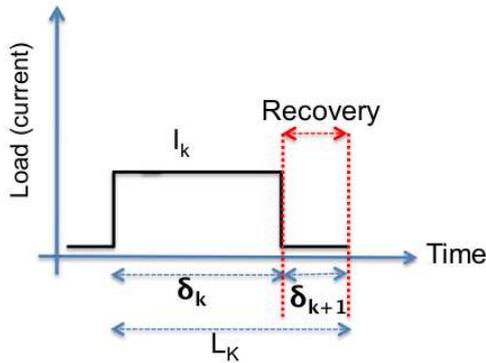


Figure 12. Load for state  $k$  with recovery.

In the actual case, *i.e.* in Fig. 12,  $L_k > \delta_k$  and the current has a duration  $\delta_k$ . Hence, if we replace  $i(\tau)$  by  $I_k$  in (1), after integrating, we obtain:

$$\alpha_k = I_k \times \left[ \delta_k + 2 \sum_{m=1}^{\infty} \frac{e^{-\beta^2 m^2 (L_k - \delta_k)} - e^{-\beta^2 m^2 L_k}}{\beta^2 m^2} \right],$$

where  $\alpha_k$  is the apparent charge of the battery lost during the duration  $L_k$ .

Using the same method to simplify equation (7), we obtain the following form of  $\alpha_k$ :

$$\alpha_k \approx I_k \left[ \delta_k + \frac{2}{\beta^2} \left( \ln\left(\frac{x-z}{x-1}\right) - A_{\beta,k} \right) \right], \quad (9)$$

where:

$$A_{\beta,k} = \frac{1}{2} \left( \frac{1}{y^2} - \frac{1}{x^2} \right) + \frac{1}{3} \left( \frac{1}{y^3} - \frac{1}{x^3} \right), \quad (10)$$

with  $x = e^{\beta^2(L_k - \delta_k)}$ ,  $y = e^{\beta^2 L_k}$  and  $z = e^{-\beta^2 \delta_k}$ .

3) *Performance evaluation*: In this part, we compare our approximation to the original model to verify its validity. To do the comparison, we need to have the value of the battery parameters  $\alpha$  and  $\beta$ . In our simulation, we use the battery parameters  $\alpha = 39668$  and  $\beta = 0.574$  obtained by direct measurements in [13]. In addition, we consider a simple current discharge profile, where the battery is discharged with a current value of 912 mA for 25 minutes without interruption. An idle period of duration 10 minutes follows the discharge period. Then, another discharge of the battery with a current of 912 mA is performed. The simulation stops when the battery is depleted, *i.e.*, when it reaches the cutoff voltage.

We use Matlab software to code both the original and the approximated models. The code of the original model is an exact representation of equation (2) with  $m$  going from 1 to 1000, while our approximation is coded using equations (8) and (9). The two models begin with an initial residual charge of 39668 ( $C = \alpha = 39668$ ) when the simulation starts.

Furthermore, in the interval  $[0, 25)$  minutes, there is no idle time periods. Thus, the charge consumed after each step (1 minute in our example) is calculated, and the residual charge of the battery (in mA/min) is updated using equation (8). In the interval  $[25, 35)$  minutes, there is no current consumption since it is an idle period, and the residual energy is updated using equation (9). In the interval  $[35, +\infty)$  minutes, (8) is used to update the energy.

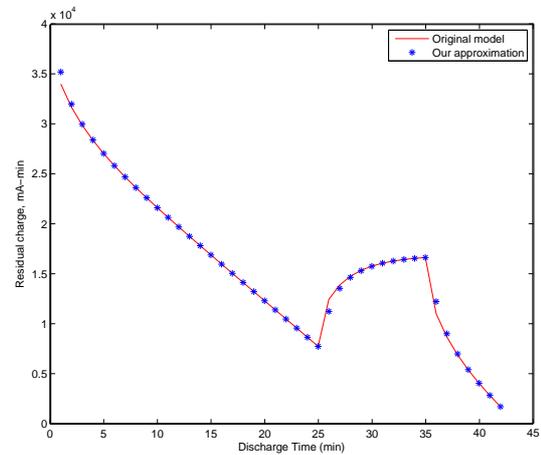


Figure 13. Comparison between the original and the approximated models.

The result of our simulation is shown on Fig. 13. The x-axis represents the discharge time in minutes and the y-axis the residual charge in mA/min. We can clearly see that the residual charge using our approximation matches the charge given by the original model. We also notice that when the battery is discharged without idle periods, its residual charge decreases sharply. This can be seen in the interval 1 to 25 on Fig. 13. In interval 25 to 35, since there is an idle period, the residual charge of the battery increases. This is normal because the concentration near

the electrode surface starts to increase, or to recover, due to diffusion.

### B. Cross-layering in OMBCE

As said at the beginning of this section, OMBCE is a transversal protocol that provides battery estimations to all the other protocols. These estimations are fed by the MAC layer, which informs OMBCE about the time spent in each state. Then, the estimation is provided to EOLSR in order to compute energy efficient routes to improve the network lifetime.

## VIII. CONCLUSION

Industrial wireless sensor networks have to address multiple issues such as long energetic autonomy, harsh radio conditions, multiple QoS requirements. Based on the case study of the OCARI project, we developed several protocols in the MAC, NWK and APS layers in order to deal with these challenges. This paper describes the cross-layering strategies used by those protocols in order to improve the performance of the network. The benefits range from the reduction of the delay (for instance using MBC and MSSS) to lower energy consumption (for instance using EOLSR and OMBCE). We validate each protocol by providing simulation results or measurements from a real hardware testbed.

## REFERENCES

- [1] S. Mahfoudh and P. Minet, "Survey of energy efficient strategies in wireless ad hoc and sensor networks," in *IEEE ICN*, 2008.
- [2] C. Buratti, A. Giorgetti, and R. Verdone, "Cross-layer design of an energy-efficient cluster formation algorithm with carrier-sensing multiple access for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking archive*, vol. 5, no. 5, pp. 672–685, 2005.
- [3] G. Chalhoub, E. Livolant, A. Guitton, A. van den Bossche, M. Misson, and T. Val, "Specifications and evaluation of a MAC protocol for a LP-WPAN," *Ad Hoc & Sensor Wireless Networks journal*, 2009.
- [4] Zigbee, "Zigbee Specification," ZigBee Standards Organization, Standard Zigbee 053474r17, January 2008.
- [5] G. Chalhoub, A. Guitton, and M. Misson, "MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part A: MaCARI: a synchronized tree-based MAC protocol," in *IFIP WSAW*, 2008.
- [6] E. Livolant, A. Van den Bossche, and T. Val, "MAC specifications for a WPAN allowing both energy saving and guaranteed delay - Part B: Optimisation of the intra-star exchanges for MaCARI," in *IFIP WSAW*, 2008.
- [7] S. Mahfoudh and P. Minet, "EOLSR: an energy efficient routing protocol in wireless ad hoc and sensor networks," *Journal of Interconnection Networks*, vol. 9, no. 4, 2008.
- [8] M.-H. Bertin, G. Chalhoub, T. Dang, S. Mahfoudh, J. Rahmé, A. van den Bossche, and J.-B. Viollet, "OCARI for industrial wireless sensor networks," in *IFIP Wireless Days*, 2008.
- [9] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks – Technologies, Analysis and Design*, 2008.
- [10] IEEE 802.15, "Part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs)," ANSI/IEEE, Standard 802.15.4 R2006, 2006.
- [11] S. Mahfoudh and P. Minet, "Maximization of energy efficiency in wireless ad hoc and sensor networks with SERENA," *Journal on Mobile Information System*, 2009.
- [12] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," IETF, RFC 3626, 2003.
- [13] D. Rakhmatov and S. Vruthula, "Energy management for battery-powered embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 277–324, 2003.
- [14] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, 1972.