

# Efficient flooding-based routing protocol with random wake-up for very low duty-cycle WSNs

Affoua Thérèse Aby, Alexandre Guitton, Michel Misson

► **To cite this version:**

Affoua Thérèse Aby, Alexandre Guitton, Michel Misson. Efficient flooding-based routing protocol with random wake-up for very low duty-cycle WSNs. *Journal of Communications (JCM)*, ACADEMY PUBLISHER, 2015, <10.12720/jcm.10.6.385-395>. <hal-01569752>

**HAL Id: hal-01569752**

**<https://hal-clermont-univ.archives-ouvertes.fr/hal-01569752>**

Submitted on 27 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Efficient flooding-based routing protocol with random wake-up for very low duty-cycle WSNs

Affoua Thérèse Aby<sup>(1,2)</sup>, Alexandre Guitton<sup>(3,2)</sup>, Michel Misson<sup>(1,2)</sup>

(1) Clermont Université, Université d'Auvergne, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France

(2) CNRS, UMR 6158, LIMOS, F-63175 Aubière, France

(3) Clermont Université, Université Blaise Pascal, LIMOS, BP 10448, F-63000 Clermont-Ferrand, France

Emails: {aby,guitton,misson}@sancy.univ-bpclermont.fr

**Abstract**– Wireless sensor networks are used in many environmental monitoring applications (e.g., to monitor forest fires or volcanoes). In such applications, sensor nodes have a limited quantity of energy, but must operate for years without having their batteries changed. The main mechanism used to allow nodes to save energy is to sequence periods of activity and inactivity. However, the design of MAC and routing protocols for applications with low duty-cycle is still a challenge. In this paper, we propose an efficient flooding-based protocol combined with an unsynchronized MAC protocol operating at very low duty-cycle ( $\leq 1\%$ ) called E-ADCR (Energy-efficient Asynchronous low Duty-Cycle Routing protocol). Although flooding-based routing protocols are usually very simple to implement, they are often costly in terms of bandwidth, due to the large number of data copies. Our flooding-based routing protocol has the following features: (i) it benefits from rare common activities to reduce the amount of data copies generated by the flooding mechanism, (ii) it guarantees the reception of frames by a limited number of receivers by sending frames multiple times during the active period, and (iii) it uses a queue management policy to ensure that frames do not stay too long in queues. With these features and the strong cross-layer design with the MAC protocol, our protocol is able to achieve good performance with this very low duty-cycle.

**Index Terms**– WSN; flooding-based routing protocol; asynchronous MAC protocol; duty-cycle.

## I. INTRODUCTION

Wireless sensor networks (WSNs) are used in many applications for environmental monitoring, such as forest fire monitoring or volcanoes monitoring. In such applications, nodes are deployed over a large area and must operate for several months (or even years) without human intervention. The main method used to allow nodes to save energy is to sequence periods of activity (with their

radio component turned on) and inactivity (with their radio component turned off) according to a certain duty-cycle. The duty-cycle represents the proportion of time when the radio component of the node remains active. For example, an activity rate of 1% means that the nodes keep their radio off during 99% of the time, and only turn it on during 1% of the time.

The MAC protocols for applications with low duty-cycle can be generally classified into synchronous or asynchronous. Synchronous MAC protocols generate a large number of control messages to achieve synchronization prior to exchanging the data, which hinders their usage when the duration of the activity of nodes is very short. Asynchronous MAC protocols do not require synchronization but usually cause large delays due to rare period of common activities between neighboring nodes.

Routing protocols have to deal with the low data rate provided by MAC protocols (which can be even reduced by the high contention for the medium access). Thus, routing protocols with limited control overhead are preferred in our context.

In this paper, we propose the E-ADCR (Energy efficient Asynchronous low Duty-Cycle Routing) protocol, which is an energy efficient flooding-based routing protocol combined with the MAC protocol of [1]. The protocol in [1] is an asynchronous, blind and opportunistic MAC protocol, i.e., nodes do not attempt to predict the activities of other nodes. In our flooding-based routing protocol, nodes are constantly sending their packets during their active period, without waiting for the detection of a potential receiver (thus, reducing the overhead). Our protocol generates few copies for each message. Simulations show that our protocol provides good performance for very low duty-cycles, unlike existing protocols.

The remainder of this paper is organized as follows. In section II, we present existing MAC and routing protocols which operate with low duty-cycle. In section III, we describe our proposition of a flooding-based routing protocol. In section IV, we show by simulation that our routing protocol provides better performance at very low duty-cycle than the existing routing protocols. Finally, we conclude our work in section V.

## II. RELATED WORKS

In this section, we review some MAC and routing protocols proposed in the literature for WSNs.

### A. MAC protocols for WSNs

MAC protocols for WSNs can generally be classified into two categories : synchronous and asynchronous. In synchronous MAC protocols, nodes agree on a common schedule for their activities and inactivity. Generally, all nodes are active simultaneously, and switch to sleep mode at the same time. In asynchronous MAC protocols, there is no common schedule.

1) *MAC protocols with synchronous duty-cycle:* The standard IEEE 802.15.4 [2] in beacon enable mode is the main synchronous duty-cycle MAC protocol. The nodes are synchronized on the coordinator wake-up by receiving its beacon, which indicates the superframe duration, as shown in Figure 1. The superframe is divided into two periods: a contention access period (CAP) and a contention free period (CFP). During the CAP, nodes access to the medium with competition according to the slotted CSMA/CA algorithm. In this algorithm, time is divided into backoff periods and transmission attempts begin after a random number of backoff periods. During the CFP, time is divided into guaranteed time slots (GTS) and nodes which have obtained a GTS during the CAP can send directly their data frame.

Other synchronous MAC protocols have been proposed, such as: TSMP [3], D-MAC [4], DW-MAC [5], MC-LMAC [6], [7], SEA-MAC [8] and [9]. In these protocols, nodes agree on the same wake-up time. The first node that broadcasts its wake-up schedule becomes the synchronizer. Note that communications between synchronized nodes is easy, because all nodes share their activity, and save energy during their common inactivity period. However, the drawback in these protocols reside in the implementation of this synchronization. It is very costly in term of energy because it requires a large number of control messages. Moreover, the fact that several nodes are active simultaneously increases the contention and collisions, and therefore the waste of energy.

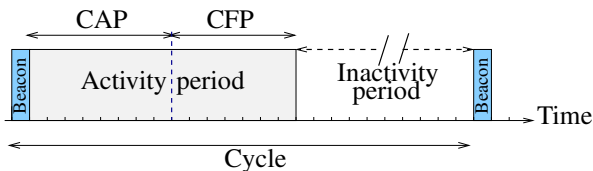


Fig. 1: Structure of the superframe in the standard IEEE 802.15.4

2) *MAC protocols with asynchronous duty-cycle:* Asynchronous MAC protocols do not require that nodes agree on a common schedule to operate. Such protocols can be classified into two categories according to the initiator of communications: *sender-initiated* if communications are initiated by the sender, and *receiver-initiated* if communications are initiated by the receiver.

In *sender-initiated* protocols, most of the communication load is supported by the sender. One of the first protocols based on this principle is B-MAC (Berkeley MAC) [10]. In B-MAC, the sender sends a long preamble before the effective transmission of data frames. When a neighboring node wakes up and detects the preamble, it remains active until the data transmission. This mechanism requires that neighbor nodes remain active until the end of the transmission, which consumes energy unnecessarily.

The WiseMAC [11] protocol has been proposed in order to reduce the over-listening in B-MAC. It uses several small preamble frames separated by short intervals, instead of one long preamble frame. Nodes include the date of their next wake-up in the acknowledgments. Thus, the sender can wake up just before the receiver, send a small preamble frame and quickly begin sending its data frames.

The X-MAC [12], [13] protocol has also been proposed to reduce the over-listening in B-MAC. In X-MAC, nodes wake up regularly and listen to the channel for 20 ms. If no preamble is detected, or if a node receives a preamble for a frame not intended for it, it switches to sleep mode for 500 ms. A node which has a packet to send wakes up immediately and listens to the channel. When the channel is free, the node sends a preamble frame with the receiver address and waits for an acknowledgment, as shown in Figure 2. When a node receives a preamble intended for it, it sends an acknowledgment to the source of the preamble to indicate its availability and remains active to receive the data. After the first sending of preamble frame, if a node hears the preamble of another node, it stops sending the preamble and waits for the acknowledgment of the current preamble before resuming with its own preambles. When a node that has stopped sending its preamble receives an acknowledgment of the expected receiver, that node waits for a random time after which it sends its packet. In this way, X-MAC provides a low end-to-end delay, but generates several collisions because the interval between two preambles can be interpreted as a free channel. Moreover, the fact that there is no acknowledgment for the data packet means that the sender has no knowledge of the reception of the packet by the receiver, which impacts greatly the data delivery rate. Finally, like most asynchronous sender-initiated MAC protocols, some nodes remain active much longer than others, which causes a problem of fairness in energy consumption, as shown in [14].

In *receiver-initiated* protocols, nodes wake up independently and indicate their availability to receive data by sending beacons. The main protocol based on this mechanism is RI-MAC [15]. In RI-MAC, when a node has a data frame to send, it wakes up immediately and waits for the beacon of a potential receiver. When a node which has no packets to send wakes up, sends the beacon and does not detect any data frame during a given period, it returns to sleep mode. Just after reception of a receiver beacon, the senders send their data frames

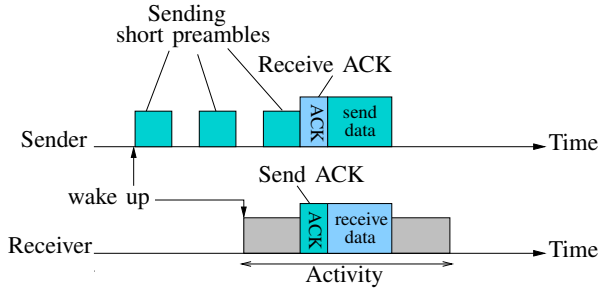


Fig. 2: X-MAC's short preamble mechanism.

immediately. This mechanism can generate collisions, but RI-MAC includes a collision management mechanism with a new broadcast of the beacon when the sender detects collisions. RI-MAC reduces the occupation of the channel caused by the preambles of X-MAC, and provides a good delivery rate ratio with low end-to-end delay. However, it does not guarantee a good trade-off between energy consumption and throughput.

The PW-MAC [16] protocol improves RI-MAC by reducing the energy waste of the sender when it is waiting for the receiver to wake up. To do this, PW-MAC uses a node-dependent, pseudo-random generator for the wake-up times. When a node knows the parameters of the pseudo-random generator of its receiver and has data frames for this receiver, it predicts the receiver wake-up and wakes up at the right time. PW-MAC incorporates a prediction error correction mechanism and efficiently manages retransmissions compared to RI-MAC. Indeed, in case of unsuccessful retransmissions, the nodes do not remain awake until the next wake-up of the receiver, but make a new prediction of the next wake-up of the receiver and switch to sleep mode until this time. However, PW-MAC does not solve the problem of unfairness in energy consumption in RI-MAC. It also causes many collisions when multiple senders send simultaneously their data frames to the same receiver. In addition, the parameters of the pseudo-random generator (which is a linear congruential generator with step of 1, of the type :  $X_{n+1} = (aX_n + c) \bmod m$ ) chosen by the authors generates sequences of period 1. Indeed, the choice of  $a = 20 \times \text{nodeID}$ ,  $c = 7$  and  $m = 1000$  does not respect the property of maximum period of [17], which states that a sequence has a maximum period (that is, of length  $m$ ) if and only if :  $c$  is prime with  $m$ ,  $b = a - 1$  is a multiple of  $p$  for each prime  $p$  dividing  $m$ , and  $b$  is a multiple of 4 if  $m$  is a multiple of 4. For example, if we consider node 1 (with  $\text{nodeID} = 1$ ), we have  $a = 20 \times 1 = 20$  and  $b = a - 1 = 19$ . The prime numbers 2 and 5 both divide  $m = 1000$ , but are not multiples of  $b$ . Node 1 generates the sequence 7, 147, 947, 947, 947, etc. A choice of  $a = 21$ ,  $X_0 = \text{nodeID}$ ,  $c = 7$  and  $m = 1000$  would have been more suitable.

In OC-MAC [18], the senders cooperate to elect one sender that awaits the availability of a given receiver, thereby reducing the energy consumption of senders.

In HKMAC [19], an hybrid approach is used. Time is

divided into random activation periods (RPs) and scheduled activation periods (SPs). During RPs, nodes operate as in RI-MAC. During SPs, receivers adaptively adjust their beacon sending time to allow the senders to schedule their wake-up time in order to reduce the large listening time of senders in RI-MAC.

Note that asynchronous MAC protocols eliminate the complexity of synchronization, but can generate a long end-to-end delay. Moreover, they do not provide fairness in the energy consumption of nodes, which reduces the overall network life time.

We proposed a receiver-initiated MAC protocol which provides fairness between nodes in term of energy consumption, in [1]. This protocol operates as follows: each node knows the duration of the global cycle, denoted  $c$ , and the duration of its activity within each global cycle, denoted  $a$ . Note that  $a$  and  $c$  is the same for all the nodes. Each node turns its radio component on during  $a$  time units every  $c$  time units, thus resulting into a duty-cycle of  $a/c$ . However, each node chooses uniformly at random the beginning of the activity within each global cycle, in the interval  $[0; c - a[$ . With this mechanism, the nodes are not synchronized. When a node is active, it uses the unslotted CSMA/CA method of the standard IEEE 802.15.4 to access the medium and sends a beacon to express its availability (see Figure 3). This protocol is blind, in the sense that the nodes do not make assumptions about the activities of other nodes. Figure 4 shows an example of the activities of three neighbor nodes  $n_1$ ,  $n_2$  and  $n_3$ . It can be noticed that the global cycles of each node are not synchronized. During the first global cycle of  $n_1$ , nodes  $n_1$  and  $n_2$  can communicate. However, nodes  $n_1$  and  $n_3$  must wait until the middle of the second global cycle of  $n_1$  to communicate.

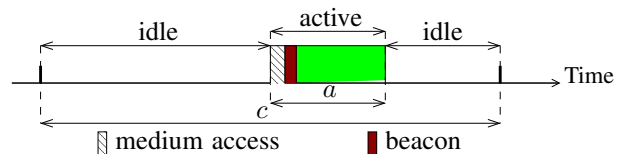


Fig. 3: Zoom on the activity of a node during a cycle.

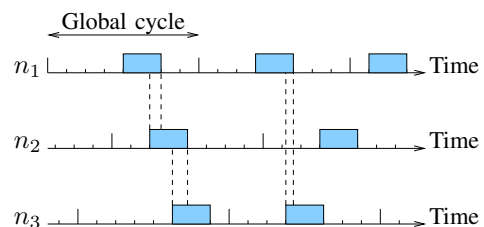


Fig. 4: Activities of three nodes, with a duty-cycle of 25% (this large duty-cycle is shown here only for clarification purposes).

The main advantages of this protocol can be summarized as : it does not require synchronization (and is thus robust to clock drifts), the nodes do not need to keep information about the topology (which makes the protocol appropriate in the presence of node mobility or changing

propagation conditions), it requires few control messages, and it can work with very low and fixed duty-cycle for all nodes (thus providing both fairness and energy-efficiency).

### B. Routing protocols for WSNs

Many routing protocols for WSN have been proposed in the literature. Most of these protocols assume that nodes are always active, or that the MAC protocol is synchronous, so that the activities of nodes overlap completely. We focus in this paper on the routing protocols that do not make such assumptions and that take advantage of opportunistic meetings. We describe more specifically some flooding-based routing protocols, as our contribution relates to this category.

1) *Opportunistic gradient routing protocols* : Opportunistic routing protocols allow nodes to communicate randomly during the meetings. They are often used in the context of MANET (Mobile AdHoc NETWORKS), but are also suitable for WSNs with infrequent and non-periodic meetings. Generally, these protocols are based on the calculation of a gradient, which is a distance from node to sink (using the number of hops, the residual energy, links reliability, etc.). The nodes send their packets to neighbors having a smaller gradient than their own. The main advantage of these protocols is that opportunistic communications provide some robustness to topology changes.

In ORW [20], the transmission of a packet by a sender node is acknowledged only by the first neighbor which provides a better path to the destination. This neighbor is not necessarily known in advance by the sender.

In [21], GMRP [22] and [23], the relay nodes select the paths with minimum delay to the sink.

ASSORT [24] aims to minimize the energy cost while sending each packet. Each node calculates the energy cost of a transmission by taking into account the cost of all its potential relays. Thus, the paths are chosen according to relays that provide the lowest energy cost from the sender to the sink.

2) *Opportunistic flooding-based routing protocols* : Flooding routing protocols broadcast multiple copies of each packet in the network, in order to ensure that the destination is eventually reached. For reducing the number of data copies, these protocols are often based on a logical tree topology for broadcasting.

Guo *et al.* have proposed in [25] an opportunistic flooding-based routing protocol, from sinks to nodes, that operates with a low duty-cycle and with unreliable links.

In [25], the packets are broadcasted via an optimal energy tree, from parent nodes to child nodes. This protocol reduces the flooding delay by exploiting links outside of the optimal energy tree (which are considered unreliable links) for sending the new packets. The senders select a relay according to a probabilistic decision based on the distribution of the delay of neighbors.

In [26] and [27], the flooding is performed from sinks to nodes according to a tree topology. The nodes having

the same parent wake up simultaneously to receive the data frames from their father.

In [28], the authors have proposed a centralized algorithm to estimate the cost of forwarding messages and the latency to cover a given network. They proposed a dynamic programming algorithm to determine the best trade-off in cost between the transmission sequences.

Note that most of the flooding-based routing protocols of the literature assume one-to-many communications, where the sink sends data to all the nodes of the network. Such protocols are not suitable for applications that require many-to-one communications, where the sensors send data to the sink.

Zhang and Fromherz proposed in [29] a flooding-based routing protocol for WSNs called constrained flooding (CF) for many-to-one communications. In CF, nodes compute a gradient called *cost-to-go* according to the routing objectives (e.g., shortest path from nodes to the sink). Each node  $n$  maintains an estimation of its cost-to-go  $c(n)$  and of the cost-to-go of its neighbors. Each time a node  $n$  hears a packet from a neighbor  $v$ , if  $n$  is the expected destination of the frame,  $n$  updates the value of the cost-to-go of its neighbor  $c_n(v)$  and re-estimates its own cost-to-go  $c(n)$  using the following formula:  $c(n) \leftarrow (1 - \alpha)c(n) + \alpha(o(v) + \min_v c_n(v))$ , where  $\alpha$  ( $0 < \alpha \leq 1$ ) is the listening rate and  $o$  is the cost function. CF does not require periodic updates to re-estimate the cost-to-go. The rules used before a packet from a neighbor  $v$  is broadcasted are the following.

- 1) The difference between  $c(n)$  and  $c_n(v)$  must be lower than a value called temperature and denoted  $T$ . This temperature gradually decreases.
- 2) In order to avoid collisions, a waiting time is set before the broadcast of each packet. This delay  $\Delta$  is computed as a function of the difference between  $c(n)$  and  $c_n(v)$ . The greater is  $\Delta$ , the smaller is the delay. For example, the delay function  $\delta$  can be expressed as  $\delta(\Delta) = D/e^\Delta$ , where  $D$  is a constant.
- 3) After this delay, a probabilistic policy is used by  $n$  to decide to send the packet or not. The more a packet is heard by a node, the less likely it will be broadcasted. The broadcast probability can be expressed as  $p(C) = 1/C^\gamma$ , where  $C$  is the number of times the packet is heard and  $\gamma \leq 0$  is the trade-off between robustness and energy.

CF reduces the number of packet copies in the network while ensuring that nodes with a large cost-to-go have their packets forwarded by neighbors with a lower cost-to-go. However, the fact that nodes automatically update their cost-to-go when they hear other nodes can yield many packet losses. Indeed, let us consider a node  $n$  with  $c(n) = 4$  which hears a neighbor  $v$  through a low quality link, and with  $c(v) = 2$ . Since  $n$  was able to hear a message from  $v$ ,  $n$  updates  $c(n)$  with value 3. From now on,  $n$  will only forward messages to  $v$  (or to other nodes having a cost-to-go of 2), even if  $v$  cannot receive all messages from  $n$ . Likewise, the delay mechanism before broadcasting packets proposed in CF

has no random parameter, which may increase the number of collisions. For example, two neighbors that receive a given packet for the first time will have exactly the same delay before packet transmission,  $D$  being fixed.

Note that the main problem of many of the previous routing protocols is that they require maintaining a neighbor table or routing information (e.g., a logical tree, or the knowledge of a cost to the destination). This maintenance requires several control messages and it is difficult to achieve when the common activities of neighboring nodes are rare.

### III. DESCRIPTION OF E-ADCR

In this section, we propose to design a flooding-based routing protocol using the asynchronous, low duty-cycle MAC protocol described in [1] (see Section II). This MAC protocol operates with very low duty-cycle, but the neighbors nodes meet rarely and for short durations. In our flooding-based routing protocol, the nodes maintain little information about the topology. The nodes constantly send their data frames (using the CSMA/CA algorithm for medium access) during their active period, without waiting to detect a potential receiver. Thus, when two neighbor nodes are active simultaneously, they begin exchanging data packets, instead of wasting time (within an already short time interval) exchanging control packets. It is clear that this approach increases the competition for channel access, however, we consider that this increase is small since the probability of having many active nodes simultaneously is also low (due to the low duty-cycle). Thus, the energy load is limited, since the sensor nodes consume approximately the same amount of energy when listening or transmitting (with a transmission power of 0 dBm, the CC2420 component consumes 18.8 mA in listen mode and 17.4 mA in transmission mode [30]).

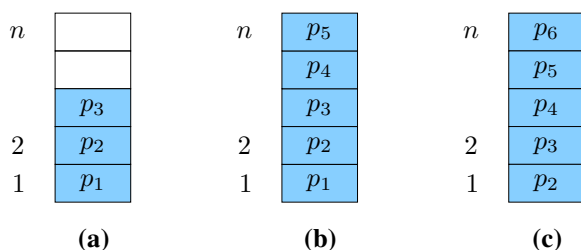


Fig. 5: Packets queuing mechanism

The E-ADCR (Energy efficient Asynchronous low Duty-Cycle Routing protocol) routing protocol works as follows. Initially, the sink broadcasts a beacon with a parameter  $hop = 0$ . All nodes that hear this beacon with higher reception power than a given threshold (called robustness threshold), set their  $hop$  to the value received plus one, then rebroadcast a beacon (with their  $hop$ ) and switch to sleep mode until the next wake-up time. After a short initialization phase, each node has an estimation of its number of hops to the sink. It can be noted that it is possible to update the parameter  $hop$  in case of change of topology or mobility, by inserting the  $hop$  parameter into

data packets. This  $hop$  parameter allows each node to set the  $TTL$  parameter for packets, which is the maximum number of links a packet can travel according to reach the destination.

Our protocol uses a packet queue, shown in Figure 5. Part (a) of Figure 5 shows the case where the queue is not full : the new packets are put at the end of queue. When a new packet arrives and the queue is full (see Figure 5(b)), the first packet in the queue (position 1) is removed and all other packets are shifted down to release the end position in order to add the new packet (see Figure 5(c)).

The  $TTL$  parameter is set by the nodes when generating their packet, and is initialized with  $TTL \leftarrow 2 \times hop$ . The factor 2 ensures that on average, at each  $hop$ , packets can be broadcast at most 2 times. Another parameter called  $queueDate$  and representing the queuing time is set in each packet. Each packet also includes a unique number (composed of the generated node ID and the packet generation number for this node).

The routing protocol is described in Algorithm 1 and operates as follows.

- 1) Just before its next activity, a node removes expired packets from its queue. Indeed, to ensure that the packets do not stay indefinitely in the queue, the packets have a limited queue lifetime noted  $maxQueueTime$ .
- 2) When a node receives a packet, if the node is the packet destination, the packet is sent to the upper layers. Otherwise, the node checks whether this packet is new in the queue (using the packet unique number) and whether  $TTL > 0$ . When both conditions hold, the node decrements the  $TTL$  parameter and puts the packet in the queue. This mechanism prevents a packet from circulating for a long time in the network.
- 3) When a node is active and has packets in its queue, it starts broadcasting those packets from the top (i.e., the last packet queued) to the bottom of the queue, until the end of its activity. The packets are not systematically removed from the queue after their broadcast because it is likely that these packets are not actually received (recall that a node sends packets without prior knowledge of the activity of any potential receiver). If a node has sent all the packets in its queue and is still active, it restarts the broadcast from the top of the queue. For example, in Figure 5(a), the node will send  $p_3, p_2, p_1, p_3, p_2$ , etc. Indeed, it is possible that a neighbor node starts its activity towards the end of the activity of the current node : if the packets are not retransmitted, the neighbor will not have the opportunity to receive this data.

The advantages of our flooding routing protocol can be summarized as follows.

- 1) It requires a limited amount of control messages before sending the data packets.
- 2) It takes advantage of the simplicity and efficiency of the shortest path, like all flooding-based routing

---

**Algorithm 1** Algorithm of our flooding-based routing protocol

---

**Require:** node  $n$  switches to active mode

**if** queue of  $n$  is not empty **then**

$current \leftarrow$  position of the last packet in queue

**for** each position of packets in queue (from 0 to  $current$ ) **do**

$p \leftarrow$  packet at position  $current$  in queue

**if**  $currentDate - queueDate(p) \geq maxQueueTime$  **then**

remove packet  $p$  from queue

$current \leftarrow current - 1$

**end if**

**end for**

**end if**

**while** node  $n$  is active **do**

**if** node  $n$  receives a packet  $p$  **then**

**if** ( $dest(p) = n$ ) **then**

send  $p$  to the upper layers

**else**

**if**  $p$  is not already in queue **and**  $TTL(p) > 0$  **then**

$TTL(p) \leftarrow TTL(p) - 1$

$queueDate(p) \leftarrow currentDate$

put  $p$  in queue

**end if**

**end if**

**end if**

**if** queue of  $n$  is not empty **then**

$current \leftarrow$  position of the last packet in the queue

$n$  broadcasts the packet of position  $current$  in queue

$current \leftarrow current - 1$

**if**  $current = 0$  **then**

$current \leftarrow$  position of the last packet in the queue

**end if**

**end if**

**end while**

---

protocols.

- 3) It does not inherit the overhead of traditional flooding protocols caused by a large number of packet copies. Indeed, the MAC protocol combined with the routing protocol allows only short common activities between nodes (which is due to the low duty-cycle) and the average number of active nodes at a given time is always low, so there are few data copies and low contention for medium access.
- 4) It does not require periodic update of the *hop* (which is just an indication to avoid packets traveling more links than needed to reach the destination).

#### IV. SIMULATION RESULTS

In order to evaluate the performance of our routing protocol, we performed several simulations. In Subsection IV-A, we describe our simulation environment. In

Subsection IV-B, we compare by simulation the MAC protocols of the literature (including the protocol proposed in [1]). In Subsection IV-C, we compare by simulation our flooding-based routing protocol E-ADCR with the routing protocols of the literature.

##### A. Simulation parameters

Our simulations are performed using the network simulator NS-2 [31]. The parameters that are common in all simulations are contained in Table I. In our simulations, we used 10 topologies of 100 nodes. In each topology, nodes are deployed uniformly at random in the area. The sink node is the node closest to the bottom right-hand corner. We generated traffic from 30 random source nodes to the sink. We observed in our topologies a maximum hop count of 7 on the paths from the sources to the sink. In all the following results, each point is an average of 10 repetitions for each topology.

Table I: SIMULATION PARAMETERS.

Deployment areas	170 m x 170 m
Transmission range	about 30 m
Total number of nodes	100
Number of source nodes	30
Transmission power	-1 dBm
Propagation model	Shadowing
Path loss exponent	2.74
Standard deviation	2.0
Packet size	30 bytes
Queue size	20 packets
Number of topologies	10
Number of repetitions per topology	10
Simulation duration	1 hour (3600 s)

##### B. Performance evaluation of the MAC protocols

In order to justify the use of [1] as the underlying MAC protocol for our E-ADCR protocol, we compare the MAC protocol of [1] with X-MAC [12], RI-MAC [15] and PW-MAC [16], which are the most representative MAC protocols with asynchronous duty cycles. We also include in the comparison the IEEE 802.15.4 standard [2], as it is the main MAC protocol with synchronous duty cycles. Note that, in this subsection, the same gradient-based routing protocol is used to route the packets to the sink.

Figures 6 and 7 show the delivery ratio and the end-to-end delay of data packets, respectively, as a function of the traffic generation period, for the IEEE 802.15.4 standard [2], X-MAC [12], RI-MAC [15], PW-MAC [16] and the protocol of [1]. The traffic generation period ranges from 5 s (which corresponds to a high traffic generation for a duty-cycle of 1%) to 30 s (which corresponds to a relatively low traffic generation for this duty-cycle).

For IEEE 802.15.4, the packet delivery ratio increases from 12% to 50% and the delay decreases from 163 s to 44 s, as a function of the traffic generation period. This low packet delivery ratio is due to a strong contention, as nodes are all synchronized during short time periods. This strong contention generates many collisions and causes

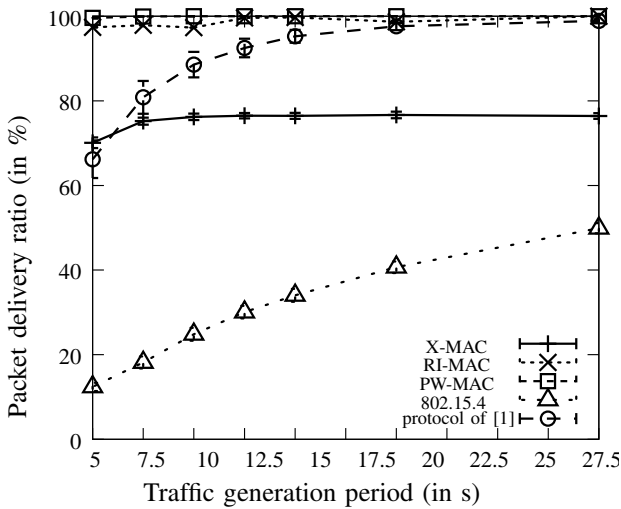


Fig. 6: Packet delivery ratio as a function of the traffic generation period, with a fixed duty-cycle of 1% for IEEE 802.15.4 and the protocol of [1], and with a variable duty-cycle for X-MAC, RI-MAC and PW-MAC.

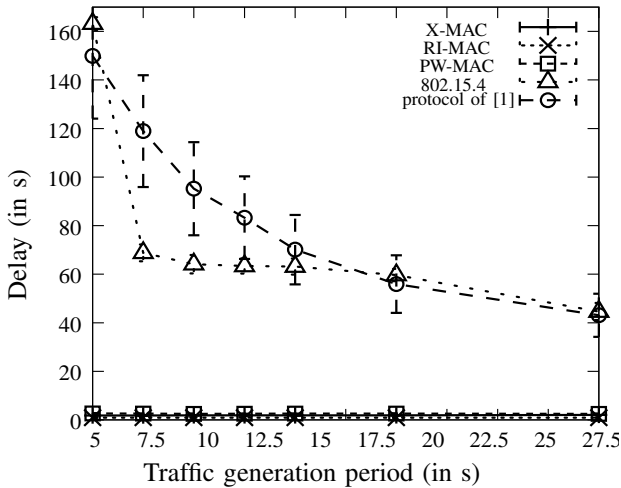


Fig. 7: End-to-end delay of data packets as a function of the traffic generation period, with a fixed duty-cycle of 1% for 802.15.4 and the protocol of [1], and with a variable duty-cycle for X-MAC, RI-MAC and PW-MAC.

overflows in nodes queues, causing a large packet loss ratio. It is also important to note that these results do not take into account the cost of synchronization, which is not implemented (all nodes are natively and perfectly synchronized in the simulation).

For the X-MAC protocol, the packet delivery ratio increases from 70% to 76% and the delay is 2 s. Packet losses occur mainly due to the relatively high number of preambles, and because a sender has no knowledge of the successful reception of packets by the receiver. The low delay is due to the fact that X-MAC does not set a fixed duty-cycle for each node: when a node has packets to transmit, it remains active to send them.

For the RI-MAC protocol, the packet delivery ratio increases from 97% to 100% and the delay is 1 s. The

high delivery ratio and the very low delay is due to the fact that as X-MAC, RI-MAC does not set a fixed duty-cycle for each node. RI-MAC reduces the channel occupation with respect to X-MAC, thus generating less collisions, and senders in RI-MAC are informed about the successful reception of packets by acknowledgments.

For the PW-MAC protocol, the packet delivery ratio is always 100% and the delay decreases from 3 s to 2 s. The delay with PW-MAC is slightly larger than with RI-MAC because nodes do not remain active until the receiver awakens. Indeed, the senders predict the awakening of the receiver. This reduces the energy consumption, but leads to an increase in the average delay every time there is a non-successful prediction.

For the protocol of [1], the packet delivery ratio increases from 66% to 99% and the delay decreases from 150 s to 43 s. The packet delivery ratio is low when the traffic is high, because the duration of the common activities is not sufficient to support a high traffic load. In contrast, the protocol of [1] achieves a large packet delivery when the traffic load is low, at the cost of a larger delay.

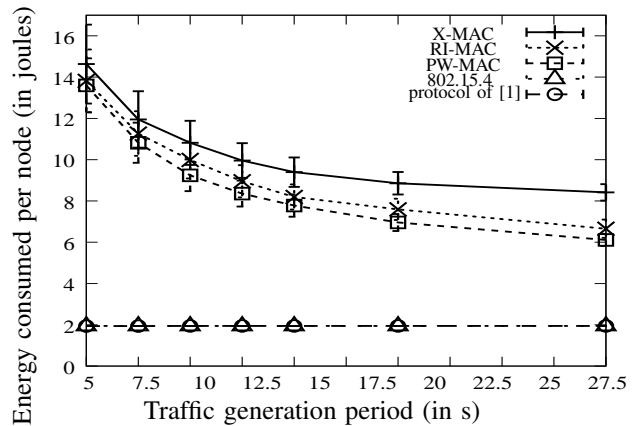


Fig. 8: Average energy consumed per node in 1 hour as a function of the traffic generation period, with a fixed duty-cycle of 1% for IEEE 802.15.4 and the protocol of [1], and with a variable duty-cycle for X-MAC, RI-MAC and PW-MAC.

Figure 8 shows the average energy consumption in joules for a period of 1 hour, as a function of the traffic generation period (from 5 s to 30 s). This energy consumption is computed using the following formula:

$$Energy = (TxTime * TxMode + ListenTime * RxMode + WakeUpTime * RxMode + IdleTime * IdleMode) * Volt,$$

where  $TxMode$ ,  $RxMode$ ,  $IdleMode$  are the energy spent in transmission, reception or idle mode, and are defined according to [30].  $TxTime$ ,  $ListenTime$ ,  $WakeUpTime$  and  $IdleTime$  correspond to the time spent in the following states : transmission, reception, radio activation/deactivation, and sleep. Finally,  $Volt$  represents the battery voltage.

For IEEE 802.15.4 and the protocol of [1], the energy consumption is always below 2 J. For X-MAC, the energy



consumption decreases from 15 J to 8 J. For RI-MAC, it decreases from 14 J to 7 J. For PW-MAC, it decreases from 14 J to 6 J. The energy consumption in X-MAC, RI-MAC and PW-MAC is high because nodes have to be active more than 1% of the time when they have packets to send, while nodes in IEEE 802.15.4 and the protocol of [1] keep the fixed duty-cycle.

The results for the MAC protocols show that synchronous duty-cycle MAC protocols are not adapted to low duty-cycles. Likewise, the asynchronous MAC protocols that do not operate at a fixed duty-cycle for all nodes are not adapted to low duty-cycles, as they yield large energy consumption. It can also be noticed that the energy consumption is not distributed fairly among nodes. For instance, the maximum energy consumption of a node for a small traffic generation period of 5 s is 60 J for X-MAC, 61 J for RI-MAC and 45 J for PW-MAC. The maximum energy consumption of a node for a large traffic generation period of 30 s is 18 J for X-MAC, 23 J for RI-MAC and 13 J for PW-MAC. The protocol of [1] provides the best trade-off in terms of energy consumption, packet delivery ratio, and end-to-end data delay for environmental monitoring applications which require a large network lifetime. Thus, we use the MAC protocol of [1] as the underlying MAC protocol in the following.

### C. Evaluation of E-ADCR

In this subsection, we evaluate the performance of our E-ADCR protocol by comparing it with two other protocols from the literature : CF [29] and a perfect gradient-base routing protocol (called Gradient in the following). In both CF and Gradient, nodes know their own distance to the sink. Nodes select as next hop any neighbor having a smaller distance to the sink.

1) *Impact of low duty-cycles*: In the first simulation scenario, we study the impact of a low duty cycle on the performance of the three routing protocols. We consider here duty cycles that are below 1%.

Figures 9 and 10 show the packet delivery ratio as a function of the duty cycle, with respectively a traffic generation period of 5 s (which corresponds to a high traffic generation given the low duty-cycle) and a period of 60 s (which corresponds to a relatively low traffic generation). The delivery ratio increases with the duty cycle for all three protocols. For CF, the delivery ratio increases from 60% to 74% when the traffic period is 5 s, and from 59% to 68% when the traffic period is 60 s. The packet delivery ratio does not vary greatly with the increase of the duty-cycle nor with the traffic load and remains low, because of the update mechanism of the most crucial parameter *cost-to-go* in the protocol. Indeed, when a node  $n$  hears another node  $v$  having a very small *cost-to-go*  $c(v)$ ,  $n$  updates  $c(n)$  accordingly.  $n$  stops forwarding packets to neighbors having a larger *cost-to-go*. This behavior reduces the performance of the protocol if the link between  $n$  and  $v$  is lossy, which is likely if  $v$  is far away from  $n$ . For the Gradient protocol,

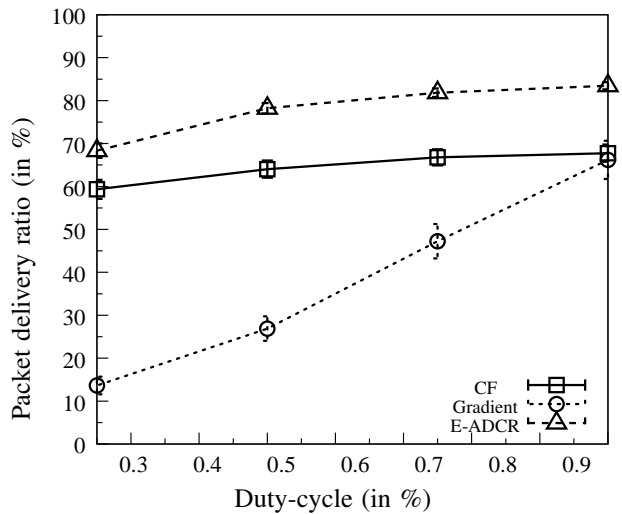


Fig. 9: Packet delivery ratio as a function of the duty-cycle for a traffic generation period of 5 s, and for the routing protocols.

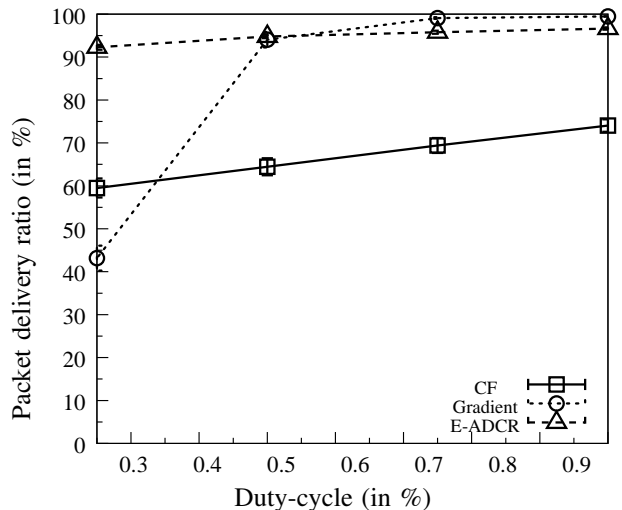


Fig. 10: Packet delivery ratio as a function of the duty-cycle for a traffic generation period of 60 s, and for the routing protocols.

the delivery ratio increases from 13% to 66% when the traffic period is 5 s, and from 43% to 100% when the traffic period is 60 s. For E-ADCR, the delivery ratio increases from 68% to 84% when the traffic period is 5 s, and from 92% to 97% when the traffic period is 60 s. E-ADCR shows good performance for low duty-cycles : it is beneficial to reduce the control overhead by sending packets frequently, rather than to wait for a neighbor which is closer to the destination.

Figures 11 and 12 show the end-to-end delay of data packets as a function of the duty cycle, with respectively a traffic generation period of 5 s and of 60 s. For CF, the end-to-end delay of data packets varies between 3 s to 4 s as the duty cycle increases, regardless of the traffic load. This can be explained by the fact that in CF, the impact of the activity duration on the number of packets

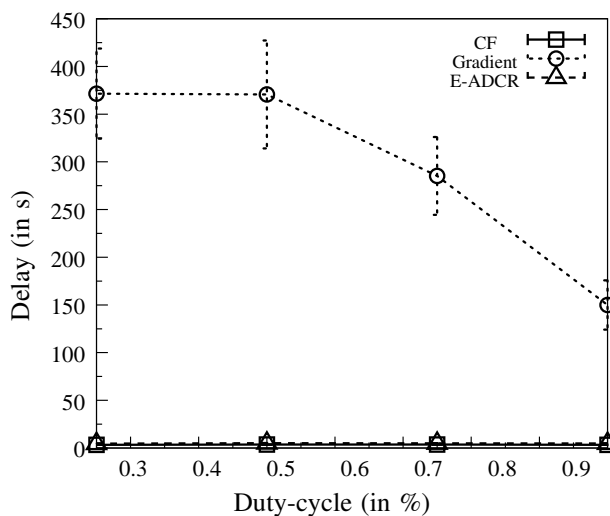


Fig. 11: End-to-end delay of data packets as a function of the duty-cycle for a traffic generation period of 5 s, and for the routing protocols.

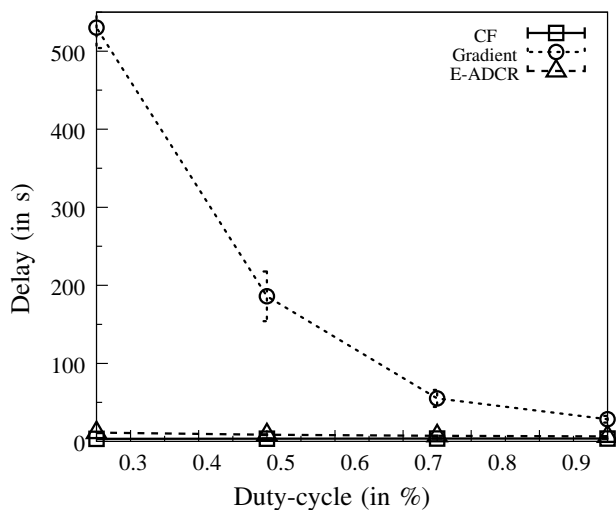


Fig. 12: End-to-end delay of data packets as a function of the duty-cycle for a traffic generation period of 60 s, and for the routing protocols.

that are rebroadcasted is small. For the Gradient protocol, the end-to-end delay of data packets decreases rapidly when the duty cycle increases, but is still high (from 371 s to 150 s with a traffic period of 5 s, and from 530 s to 28 s with a traffic period of 60 s). Indeed, as the duty cycle increases, there are more opportunities to meet neighbors. The meetings with neighbors also last longer, which means that more packets are exchanged and the beacon overhead is reduced. For E-ADCR, the end-to-end delay of data packets is low (from 4 s to 5 s with a traffic period of 5 s, and from 11 s to 6 s with a traffic period of 60 s). This shows that sending packets to all possible neighbors allow the E-ADCR protocol to find the quickest route for packets (among many other routes). Remember that the delay is computed based on the received packets only (see Figure 10), and it takes into account the time

of the first reception of the packet by the destination.

2) *Impact of large duty-cycles*: In the second simulation scenario, we study the impact of a large duty cycle on the performance of the three routing protocols. We consider here duty cycles that are between 1% and 10%.

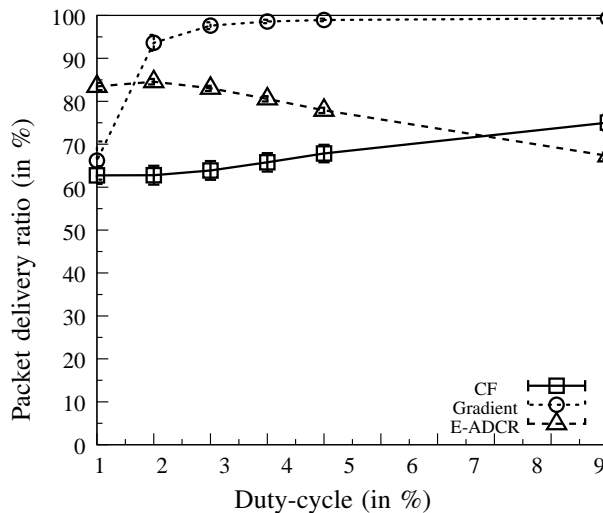


Fig. 13: Packet delivery ratio as a function of the duty-cycle for a traffic generation period of 5 s, and for the routing protocols.

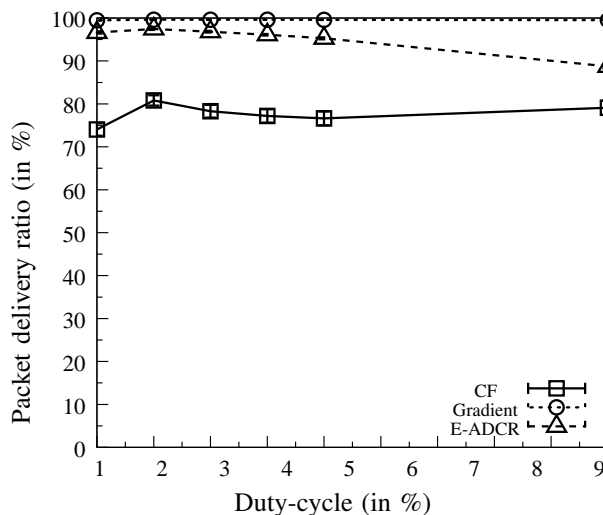


Fig. 14: Packet delivery ratio as a function of the duty-cycle for a traffic generation period of 60 s, and for the routing protocols.

Figures 13 and 14 show the packet delivery ratio as a function of the duty cycle, with respectively a traffic generation period of 5 s and of 60 s. For CF, the results show that there is no significant variation of the delivery ratio (from 63% to 75% with a traffic generation period of 5 s, and from 74% to 79% with a traffic generation period of 60 s) when the duty-cycle is large. For the Gradient protocol, the packet delivery ratio increases significantly with the duty-cycle (from 66% to 97% with a traffic generation period of 5 s). For the E-ADCR protocol, the

packet delivery ratio reaches a maximum with a duty-cycle of 2%. This is explained by the fact that when the nodes have long activities, common activities with neighbors occur more frequently. This results into an important number of data packets copies and an increase of the number of collisions, thus limiting the packet delivery ratio.

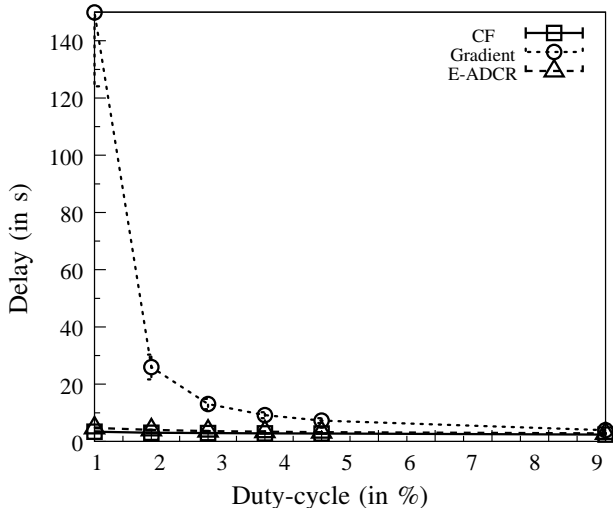


Fig. 15: End-to-end delay of data packets as a function of the duty-cycle for a traffic generation period of 5 s, and for the routing protocols.

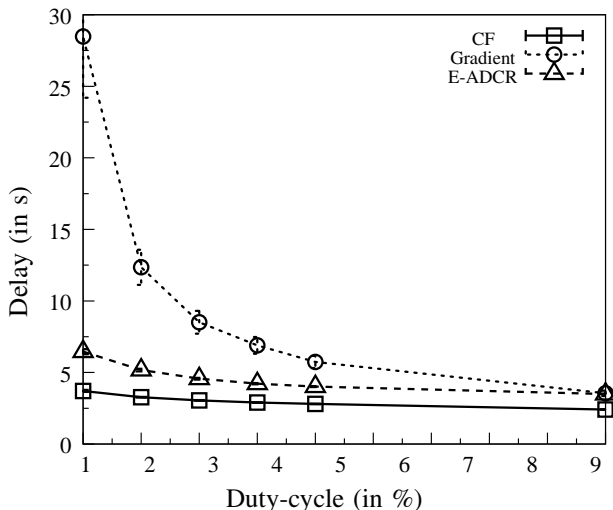


Fig. 16: End-to-end delay of data packets as a function of the duty-cycle for a traffic generation period of 60 s and for the routing protocols.

Figures 15 and 16 show the end-to-end delay of data packets as a function of the duty cycle, with respectively a traffic generation period of 5 s and a period of 60 s. For the CF protocol, the end-to-end delay varies between 2 s and 3 s for both traffic generation periods, when the duty-cycle increases from 1% to 10%. For the Gradient protocol, the end-to-end delay decreases significantly when the duty-cycle increases (from 150 s down to 3 s with

a traffic generation period of 5 s, and from 28 s down to 3 s for a traffic generation period of 60s), when the duty-cycle increases from 1% to 10%. This increase of performance is due to the fact that with a large duty cycle, the nodes meet often for large durations, and do not have to wait for a long time before meeting a node closer to the sink. For the E-ADCR protocol, the end-to-end delay remains low (between 3 s and 6 s for both traffic generation periods), even with a high packet delivery ratio (see Figures 9 and 10).

In summary, we can notice that the performance of the CF protocol in terms of packet delivery ratio remains below the performance of both E-ADCR and Gradient protocols. When the duty cycle is above 1%, the Gradient protocol shows better performance than the E-ADCR protocol. This result is generally admitted, as flooding-based routing protocols are often less efficient than gradient-based routing protocols. However, it is interesting to notice that when the duty cycle is below 1%, the performance of the Gradient protocol is much lower than the performance of the E-ADCR protocol. The E-ADCR protocol takes up the challenge of effectively operating at very low duty-cycles. It guarantees a packet delivery ratio over 92% and an end-to-end delay of about 11 s (in our simulations), even when the duty cycle is fixed at 0.25% and a traffic generation period of 1 packet per minute.

Figure 17 shows the energy consumption in joule during 1 hour as a function of the duty-cycle (from 0.25% to 1%). As expected, the energy consumption increases (from 0.62 J to 1.94 J) with the duty-cycle (from 0.25% to 1%). The energy consumption is similar for all three protocols because of the same MAC protocol that operates with a fixed duty-cycle for all nodes. The duty-cycle is therefore a better indicator to estimate the network lifetime.

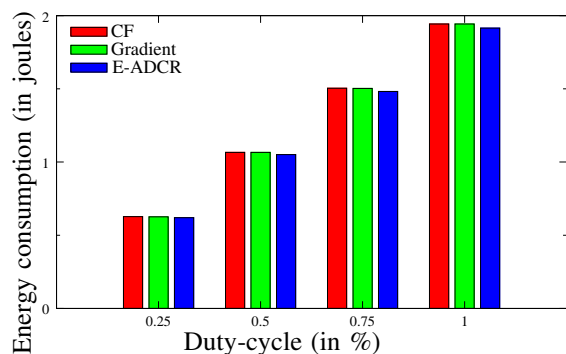


Fig. 17: Energy consumption as a function of the duty-cycle for a traffic generation period of 30 s, and for the three routing protocols.

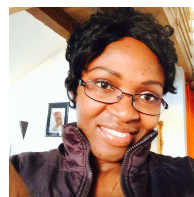
## V. CONCLUSION

Most routing protocols for WSNs require control messages to maintain their neighborhood and take routing decisions. These protocols usually behave poorly when the duty cycle is very low (that is, below 1%), as the control overhead becomes significantly greater than

the data traffic. In this paper, we proposed the routing protocol E-ADCR based on a flooding mechanism, and operating over a blind, opportunistic MAC protocol. In E-ADCR, once a node is active, it keeps broadcasting as many packets as possible. When a neighbor eventually receives one of this transmissions, the neighbor starts broadcasting this packet in turn. The E-ADCR protocol require limited control messages, and we show that it is suitable for networks operating on very low duty cycles, where active links between neighbor nodes are scarce. We compared several protocols by simulation. The results show that E-ADCR yields good performance in terms of end-to-end delay and packet delivery rate for duty cycles below 1%. As future work, we plan to combine the flooding mechanism of E-ADCR with a gradient-based mechanism, and to use the mechanism according to the packet priority or to the actual duty cycle.

## REFERENCES

- [1] A. T. Aby, A. Guitton, and M. Misson, "Asynchronous blind MAC protocol for wireless sensor networks," in *IWCMC (International Wireless Communications and Mobile Computing Conference)*, 2014.
- [2] 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.
- [3] K. Pister and D. Lance, "TSMP: Time synchronized mesh protocol," *IASTED Distributed Sensor Networks*, pp. 391–398, November 2008.
- [4] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks," in *Wireless Communications and Mobile Computing*, vol. 7, September 2007, pp. 863–875.
- [5] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks," in *ACM MobiHoc*, 2008, pp. 53–62.
- [6] D. I. Ozlem, V. H. Lodewijk, J. Pierre, and H. Paul, "MC-LMAC: A multi-channel MAC protocol for wireless sensor networks," in *Ad Hoc Networks*, vol. 9. Elsevier, 2011, pp. 73–94.
- [7] C. Tengfei, W. Thomas, P. Kris, and W. Qin, "Adaptive synchronization in multi-hop TSCH networks," *Computer Networks*, vol. 76, pp. 165–176, 2015.
- [8] Y. Z. Zhao, C. Y. Miao, and M. Ma, "An energy-efficient self-adaptive duty cycle MAC protocol for traffic-dynamic wireless sensor networks," in *Wireless Personal Communications*, 2012, pp. 1287–1315.
- [9] S. David, V. Xavier, W. Qin, W. Thomas, and P. S. J. Kristofer, "Adaptive synchronization in IEEE 802.15.4e networks," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 795–802, 2014.
- [10] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *ACM Sensys*, November 2004.
- [11] A. El-Hoiydil, J.-D. Decotigniel, and J. Hernandez, "WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks*, ser. LNCS, vol. 3121. Springer Berlin / Heidelberg, 2004, pp. 18–31.
- [12] M. Buettner, Y. Gary, V. E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *ACM Sensys*, Boulder, Colorado, USA, November 2006.
- [13] P. Tom, H. Gertjan, B. Maarten, and L. Koen, "The MAC framework: redefining MAC protocols for wireless sensor networks," in *Wireless Networks*, 2010, pp. 2013–2029.
- [14] Z. Li, M. Li, and Y. Lui, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *ACM Transactions on Sensor Networks (TOSN)*, vol. 10, April 2014.
- [15] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *ACM Sensys*, 2008.
- [16] Y. Sun, O. Gurewitz, and D. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *IEEE INFOCOM*, April 2011, pp. 1305–1313.
- [17] D. E. Knuth, *The art of computer programming: sorting and searching*. Pearson Education, 1998, vol. 3.
- [18] X. Wang, X. Zhang, G. Chen, and Q. Zhang, "Opportunistic cooperation in low duty cycle wireless sensor networks," in *IEEE ICC'10*, May 2010, pp. 1–5.
- [19] H. Tang, C. Sun, Y. Liu, and B. Fan, "Low-latency asynchronous duty-cycle MAC protocol for burst traffic in wireless sensor networks," in *IWCMC'13*, July 2013, pp. 412–417.
- [20] O. Landsiedel, E. Ghadimi, S. Duquenooy, and M. Johansson, "Low power, low delay: Opportunistic routing meets duty cycling," in *Proc. ACM/IEEE IPSN'12*, Beijing, China, April 2012.
- [21] S. Lai and B. Ravindran, "On distributed time-dependent shortest paths over duty-cycled wireless sensor networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.
- [22] J. Hao, Z. Yao, K. Huang, B. Zhang, and C. Li, "A gradient-based multiple-path routing protocol for low duty-cycled wireless sensor networks," *Wireless Communications and Mobile Computing*, 2014.
- [23] X. Zhang, F. Yan, L. Tao, and D. K. Sung, "Optimal candidate set for opportunistic routing in asynchronous wireless sensor networks," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*. IEEE, 2014, pp. 1–8.
- [24] C. Hsu, M. S. Kuo, S. C. Wang, and C. F. Chou, "Joint design of asynchronous sleep-wake scheduling and opportunistic routing in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 63, no. 7, pp. 1840–1846, 2014.
- [25] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," *Proc. ACM MobiCom'09*, pp. 133–144, 2009.
- [26] S. Guo, S. K. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *Proc. IEEE ICNP'11*, October 2011, pp. 383–392.
- [27] L. Cheng, Y. Gu, T. He, and J. Niu, "Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1393–1401.
- [28] F. Wang and J. Liu, "On reliable broadcast in low duty-cycle wireless sensor networks," in *Mobile Computing, IEEE Transactions*, vol. 11, no. 5, May 2012, pp. 767–779.
- [29] Y. Zhang and M. P. J. Fromherz, "A robust and efficient flooding-based routing for wireless sensor networks," *Journal of Interconnection Networks*, vol. 7, pp. 549–568, December 2006.
- [30] T. Instruments, "CC2420: 2.4 GHz IEEE 802.15. 4/ZigBee-ready RF transceiver," Available at Available at <http://www.ti.com/lit/gpn/cc2420>, p. 53, 2006.
- [31] "Network simulator 2," 2002, <http://www.isi.edu.nsnam/ns>.



**Affoua Thérèse ABY** is PhD student in Computer Science at University of Blaise Pascal Clermont-Ferrand, France. She is doing his research at LIMOS-CNRS. His thesis concerns a scale wireless sensor networks for environmental monitoring. His current research interests in wireless communications, sensor networks, energy-efficient MAC and Routing protocols.



**Alexandre Guitton** is an associate professor at Clermont Université, Université Blaise Pascal, France. He is doing his research at LIMOS-CNRS. He received his PhD in 2005 and his MSc in 2002 at University of Rennes I, in the field of computer networks. He has been working at Clermont Université as an assistant professor since 2007, and as an associate professor since 2014. His research interests include wireless communications, sensor networks, MAC protocols, routing protocols

(including loop detection and removal), and energy-efficiency.



**Michel Misson** obtained his PhD thesis in nuclear physics and his "Habilitation à Diriger des Recherches" (HDR) degree respectively in 1979 and 2001 both at Université Blaise Pascal, Clermont-Ferrand in France. In 1983, he became a lecturer, teaching networks and system architecture in the Computer Science Department of the institute of Technology in Clermont-Ferrand. He is now Professor in the Networks and Telecommunications Department and he manages the research team

"Réseaux et Protocoles" of the Computer Science Laboratory of Université Blaise Pascal: LIMOS-CNRS. He served on many conference committees and journals reviewing processes. He has also held a visiting position in the research Laboratory Télébec in Underground Communications of Quebec University in Abitibi-Temiscamingue (LRTCS-UQAT). His current research interests are Wireless Local Area Networks, Low Power Wireless Personal Networks, Wireless Sensor Networks real-time systems and protocol engineering.