



**HAL**  
open science

# Asynchronous blind MAC protocol for wireless sensor networks

Affoua Thérèse Aby, Alexandre Guitton, Michel Misson

► **To cite this version:**

Affoua Thérèse Aby, Alexandre Guitton, Michel Misson. Asynchronous blind MAC protocol for wireless sensor networks. IWCMC (International Wireless Communications and Mobile Computing Conference), 2014, Nicosia, Cyprus. hal-01789139

**HAL Id: hal-01789139**

**<https://hal.science/hal-01789139v1>**

Submitted on 9 May 2018

**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.

# Asynchronous blind MAC protocol for wireless sensor networks

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-63173 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**—In the past few years, energy conservation has been the main focus of researchers working on wireless sensor networks. One of the main technique to save energy is to deactivate periodically the radio module of sensor nodes: nodes alternate periods of activity and periods of inactivity, which is referred to as their duty-cycle. In this paper, we focus on asynchronous duty-cycle mechanisms, as these mechanisms are usually simple, do not require time synchronization and support network changes. We propose an asynchronous MAC protocol based on blind rendez-vous and random wake-up. Our protocol is based on a modification of the IEEE 802.15.4 standard, where activities start at a random time within each activity cycle. Our simulations show that our protocol can achieve a good performance under various scenarios, for small duty cycles (ranging from 0.1% to 5%).

**Keywords:** MAC protocol, asynchronous duty-cycle, blind protocol, IEEE 802.15.4.

## I. INTRODUCTION

Energy conservation is an important factor in wireless sensor networks (WSNs) dedicated to environmental monitoring such as volcanoes [1], fields [2], and bridges [3]. Because the environment of deployment for these types of applications has no fixed infrastructure and is not always be accessible to recharge the sensors. As the radio module is the component of a sensor node that consumes the most energy, energy-efficient MAC protocols aim at deactivating the radio module as often as possible (thus, the node can neither send nor receive during these periods). Many MAC protocols of the literature use a sequence of periodic activity and inactivity periods, called duty cycle. These protocols can be classified into two main categories: synchronous and asynchronous MAC protocols.

In the synchronous MAC protocols (such as D-MAC [4], SEA-MAC [5], and others [6], [7]), the nodes share a common vision of time, obtained through synchronization. The IEEE 802.15.4 standard [8] in beacon-enabled mode is one of the main synchronous MAC protocol for WSNs. However, synchronous MAC protocols suffer from several drawbacks: the synchronization requires to exchange a large number of messages, is hard to achieve when the topology is large or dynamic, and is not suitable to low duty cycles.

In the asynchronous MAC protocols [9], a mechanism is required to ensure that nodes can communicate. This mechanism can be initiated by the sender (using a preamble, such as in B-MAC [10] or in [11], [12]) or by the receiver (such as in RI-MAC [13], PW-MAC [14], HKMAC [15]). The nodes can

also start their activities using an on-demand wake-up (such as in DW-MAC [16] or in [17]), a scheduled wake-up (as in [18], [19]) or a random wake-up (as in RAW [20] and [21]).

In this paper, we propose an asynchronous MAC protocol, based on random wake-up for nodes. Nodes have a fixed duty cycle, but start their activities at random intervals (for a constant duration). Our protocol is blind in the sense that a node does not know the activity periods of its neighbors. When a node wakes up, it sends a beacon frame to notify its active neighbors that it can be used to forward data to the sink. Our protocol has the main following advantages: it is based on low control overhead, it is suitable to various scenarios (involving mobility or a large number of nodes) such as environmental monitoring, as nodes do not need to maintain any information about the topology (such as routing tables or neighborhood information), and does not require any synchronization between nodes. The delay required for a source to send data to the sink might be large, however, our protocol is able to operate on very low duty cycles (typically, around 1%).

The remainder of this work is organized as follows. Section II describes some MAC protocols of the literature. Section III presents our asynchronous, blind MAC protocol. Section IV gives our simulations results, performed in a large variety of scenarios. Finally, Section V summarizes and concludes our work.

## II. RELATED WORKS ON ASYNCHRONOUS MAC PROTOCOLS

In asynchronous MAC protocols, nodes are not synchronized. When a source node has to send a frame to a destination node, the source and the destination have to be both awake. To do this, an explicit mechanism is required, and various mechanisms are described in the following. Notice that generally, asynchronous MAC protocols yield large delays, but have a low energy consumption.

In sender-initiated MAC protocols, the source sends a long preamble before each frame, and receivers wake up periodically to sense the preamble. The B-MAC protocol [10] is the main protocol of this category, and it has been improved in [11] and [12] by using short preambles. The drawbacks of these protocols are that energy consumption depends heavily on traffic production, and that the medium is often busy

because of the preamble, rather than because of actual frame transmissions, which reduces the overall performance.

In receiver-initiated MAC protocols, a node that wakes up sends a beacon frame to inform its neighbors that it is now able to receive frames. A node has to receive a beacon frame before being allowed to send its data frame. This mechanism is used by RI-MAC [13], for instance, but it introduces a wasted period as the sender has to wait for the reception of the beacon. This overhead is reduced by PW-MAC [14], by having each node use a scheduled duty cycle (based on a pseudo-random generator), and by ensuring that each node knows the schedule of its neighbors, which introduces additional overhead. HKMAC [15] uses a hybrid approach, where time is divided into random activation periods (similar as in RI-MAC) and scheduled activation periods (where activation periods are scheduled, which requires synchronization).

On-demand wake-up protocols assume that each sensor node possesses two radio modules: one is used for frame transmission, and the other is used to notify destination nodes of an incoming frame and consumes low energy. A node waits with the first radio module inactive but the second radio module active. When it receives the notification on a frame incoming on the second radio module, it activates the first radio module and is able to receive the frame. Examples of on-demand wake-up protocols are DW-MAC [16] and [17]. The main drawbacks of these protocols are the use of two radio modules (which increases the cost of the sensor nodes), as well as the energy consumption of the second radio module.

Scheduled wake-up protocols divide activity periods of neighboring nodes using repetitive patterns, such that the activity of neighboring nodes overlap eventually. The existing protocols [18], [19] differ on the way the patterns are built. These protocols have the advantage to be simple to implement, but the produced patterns might depend on the number of neighbors of each node.

Random wake-up protocols are protocols where the activation periods are chosen randomly. These protocols are suited for dense deployments, as they assume that when a node wakes up, it has another active neighbor to communicate with. Such protocols include RAW [20] and [21].

### III. PROPOSITION

In this section, we describe our proposition of an asynchronous, blind MAC protocol for WSNs. The protocol is receiver-initiated, and is based on a random wake-up mechanism (although it does not assume a dense deployment of nodes). It also operates according to a fixed duty cycle, that is, node activity does not depend on traffic.

#### A. Principle of operation

Each node knows the duration  $c$  of a cycle (although this duration might vary slightly from nodes to nodes, due to clock skews), and its activity duration  $a$  per cycle. During each cycle, the node decides randomly when the activity starts by waiting for a random delay in  $[0; c - a]$  at the beginning of each cycle. With this mechanism, two nodes that are in range

will eventually share a common activity period (even if their cycles do not start at the same time, or if the duration of the cycle is slightly different for each node).

Figure 1 shows this mechanism for three nodes  $n_1$ ,  $n_2$  and  $n_3$ , that are supposed to be in range, and with a duty cycle of 10% (that is,  $a = 0.1 \times c$ ). Note that to simplify the presentation, the cycles of each node are shown as if they were starting at the same time, which is not the case in practice. It can be seen that, on this example, any pair of node can eventually share a common activity period, even though this does not happen at each cycle. On this example, there is even a period where all three nodes share a common activity period.

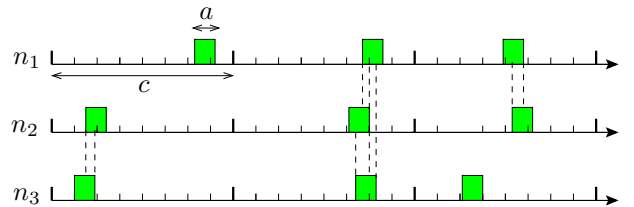


Figure 1. An example of the activity of three nodes, with a duty cycle of 10%, and with nodes starting their activities randomly in each cycle.

The delay in our mechanism can be high, because a source has to wait for a common period with the destination before sending a frame. This delay depends on parameters  $a$  and  $c$ . However, it is possible to reduce this delay by fragmenting the activity into  $f$  parts: instead of being active once for a duration of  $a$  per cycle, a node can become active  $f$  times per cycle, for a duration of  $a/f$ . In this way, the duty cycle remains  $a/c$ , but nodes share common activities more often (but for less time). Finding the optimal number of fragmentations  $f$  such that the delay is low but the duration of common activities is long enough to communicate is an issue of this mechanism, which will be discussed in Section IV.

Figure 2 shows this mechanism with a fragmentation of  $f = 2$ , again with a duty cycle of 10%. It can be noticed that each node has now two short activities per cycle, which results into more, but shorter, common activities.

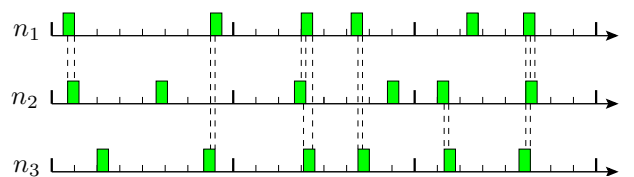


Figure 2. An example of the activity of three nodes, when the activity is fragmented into  $f = 2$ , and with a duty cycle of 10%.

The energy savings of our protocol come from several features. First, as it is completely asynchronous, it is robust against clock drifts, which become relevant when nodes are inactive for large durations. For instance, if a clock drift of 30 ppm is assumed, the clock drift between two nodes can exceed 320  $\mu s$  after only 4 s (which is the duration of a backoff slot in the slotted CSMA/CA mechanism of IEEE 802.15.4 in beacon-enabled mode). This means that

a global synchronization has to be performed every 4 s, which generates a large overhead if nodes are active only 1% of these 4 s. Second, nodes are able to communicate with any neighbor, which improves the energy efficiency of the approach. For instance, if nodes have a duty cycle of 1%, the expected number of cycles for two nodes to share an activity is 100. However, if a node has more neighbors, the expected number of cycles for a node to communicate with a neighbor greatly decreases: it is 50 for two neighbors and 33 for three neighbors. This overhead becomes small as the number of neighbors increase, as shown in results given later in Section IV (see for instance the delivery rate as a function of the duty cycle for a varying number of next hops, as it is related to the number of common active periods).

### B. Description of the MAC protocol

Our MAC protocol uses the unslotted CSMA/CA mechanism of the non beacon-enabled mode of the IEEE 802.15.4 standard, but uses a different activation mechanism (based on random wake-up). When a node wakes up, it transmits a beacon frame. To reduce the probability of a collision with this beacon frame, the beacon is sent using unslotted CSMA/CA, as shown on Figure 3. Because of the random backoffs of unslotted CSMA/CA, the beacon frame might be delayed, which reduces the duration of the active period that can be used to transmit frames. All data frames (which are sent in unicast) are required to be acknowledged. A data frame which is not acknowledged is retransmitted at most four times.

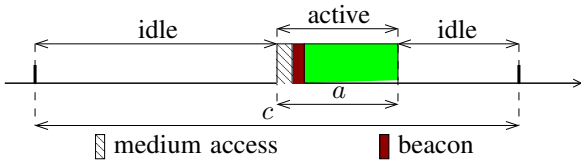


Figure 3. Zoom on the activity of a node during a cycle.

Our protocol also differs from IEEE 802.15.4 by the fact that our protocol uses a receiver-initiated, gradient-based approach in order to identify active next-hops towards the sink. Algorithm 1 presents this approach. Each node stores a hop count (used as a gradient), which is the distance to the sink. When a node  $n$  wakes up, it sends a beacon frame to inform the neighbors that are further away from the sink of its potential availability as a next hop. A neighbor  $r$  that is further away from  $n$  can use  $n$  as a next hop if their common active duration is greater than a threshold (defined as twice the expected time to transmit a frame using unslotted CSMA/CA). Then,  $n$  waits to receive beacon frames from neighbors that are closer to the sink. When it receives such a beacon, it considers the neighbor as a potential next hop (for this active period only). Nodes having queues which are nearly filled are not available (but the node can still store frames produced locally).

The main advantages of our proposition can be summarized as: it does not require synchronization (and is robust to clock skews and drifts), nodes do not need to maintain information about the topology (which makes the protocol suitable in

---

**Algorithm 1** Our receiver-initiated, gradient-based approach to identify the next-hops of a node  $n$ .

---

```

if there is enough room in the queue size of  $n$  for five frames
then
     $n$  is available
else
     $n$  is not available
end if
 $n$  sends a beacon frame (with its current hop count, availability, and activity duration  $a$ )
while node  $n$  is active do
    if a beacon frame is received from neighbor  $r$  then
        if hop count of  $r <$  current hop count of  $n$  then
            if node  $r$  is available then
                neighbor  $r$  becomes a potential next hop (for this active period)
            end if
        else
            if  $n$  is available and the remaining common active duration is greater than a threshold then
                node  $n$  sends a beacon message to inform  $r$  of its availability (with its current hop count, availability, and remaining active time)
            end if
        end if
    end if
    if there is another frame to send and there is enough remaining time to send the frame and receive the acknowledgment then
         $n$  sends the frame to any potential next hop
    end if
end while

```

---

the case of mobility or varying propagation conditions), the protocol has a low control overhead, and the protocol can operate with low duty cycles (typically, around 1%).

## IV. SIMULATION RESULTS

In order to evaluate the performance of our protocol, we performed simulations using the network simulator NS-2 [22], in three different scenarios (each scenario corresponding to a subsection). The transmit power is set to 0 dBm, and the propagation model is the shadowing model with a path loss of 3.0 and a standard deviation of 2 dBm. In the following, results are averaged over 100 repetitions, and each repetition lasts for 5000 seconds.

Unless specified otherwise, the activity cycle is 5 s, the duty cycle is 5%, the number of fragmentations  $f$  is 1, and one packet is generated at the source every 8 s.

### A. Scenario 1: Evaluation of a single link

In this first scenario, we evaluate the performance of the MAC protocol between a source and a destination in range. We use this scenario to compute the delivery rate and the delay between the generation of packets at the source and

their reception at the destination. Our objective is also to find the optimal value for the number of fragmentations  $f$ , which is an important parameter of our protocol.

The delivery rate is equal to 100% when  $f \in [2; 20]$ , and is about 99.9% when  $f = 1$  or when  $f \in [21; 25]$ . This can be explained as follows. When the number of fragmentation is high ( $f \geq 21$ ), the delivery rate decreases slightly. Indeed, in this case, a node is active during very short periods (of 12,5 ms for  $f = 20$  fragmentations), which yields common activities that, sometimes, do not last long enough for the source to detect the destination, perform the required backoffs, and actually send the frames to the neighbor. This shows that the value of  $f$  has to be limited. When the number of fragmentation is low ( $f = 1$ ), the time required for the source to have a common activity with the destination is very large (it is larger than 25 s, that is 5 cycles). Frames accumulate in the queues of nodes, and some are eventually dropped.

Figure 4 shows the delay as a function of the number of fragmentations  $f$ . When the number of fragmentations is high ( $f \geq 20$ ), the delay is high due to the fact that nodes often meet for a duration which is too short to send frames. When the number of fragmentations is low ( $f \leq 5$ ), the delay is high due to the time required for the source to meet the destination.

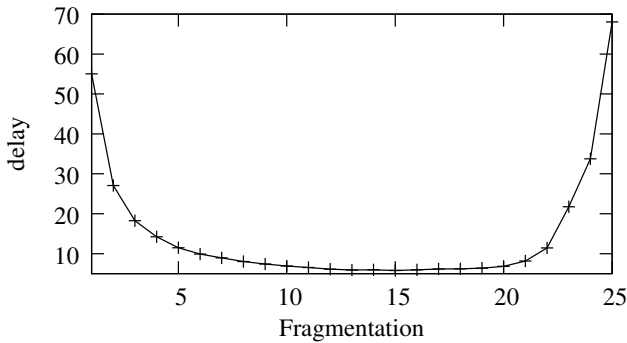


Figure 4. Delay of data frames as a function of the number of fragmentations, on a link and with a duty cycle of 5%.

With these settings, the number of fragmentations  $f$  can be set 15, which correspond to both a large delivery rate and a small delay. Each node has an activity of about 17 ms, and 15 activities every 5 s.

### B. Scenario 2: Evaluation of several possible next hops

In this second scenario, we evaluate the performance of our receiver-initiated, gradient-based approach. To do so, we consider a source in range of  $k$  intermediate nodes. Each of these intermediate nodes is in range of the destination. Thus, the source has  $k$  potential next hops in order to send the frame to the destination. The topology is shown on Figure 5, where  $s$  is the source and  $d$  is the destination. We refer to this topology as the diamond topology (with  $k$  intermediate nodes). Both delivery rates and delay are computed from source to destination (that is, they take into account the two links each frame has to follow).

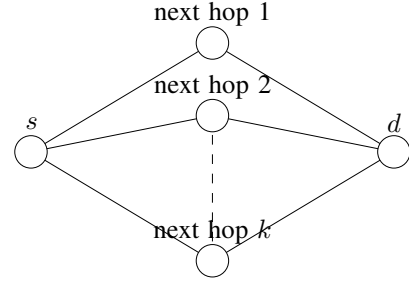


Figure 5. Network topology (called the diamond topology) for the second and third scenarios, with  $k$  possible next hops for the source  $s$ .

The delivery rate is equal to 100% when the number of next hops  $k$  varies between 2 and 6. When  $k = 1$ , the delivery rate is equal to about 99.8%. Indeed, when the number of next hops in the topology is low ( $k = 1$ ), the delivery rate decreases. This is due to the fact that the source has few opportunities to send frames: frames fill node queues quickly, and get eventually dropped. Note that smaller delivery rates are achieved with periods smaller than 8 s. The small difference between the delivery rate for  $k = 1$  and in the first scenario is due to the fact that in the second scenario with  $k = 1$ , there are two links, while there is only one link in the first scenario.

Figure 6 shows the delay as a function of the number of next hops  $k$ . As the number of possible next hops increases, the source has a larger probability to have a common activity with one of these next hops, which decreases the delay to send frames. This illustrates that when  $k$  is large, delivery rate is high and delay is low. In the following, we focus on small numbers of next hops. More specifically, we study the cases where  $k$  varies between 1 and 4, as we believe that these numbers of next hops are realistic for many WSNs.

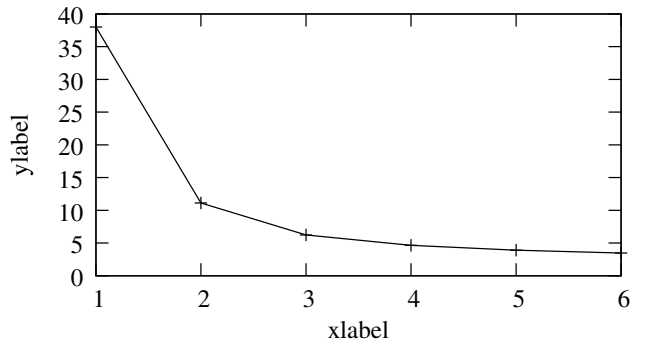


Figure 6. Delay of data frames as a function of the number of next hops, on a diamond topology.

Figure 7 shows the delivery rate as a function of the duty cycle, for several number of possible next hops. When the duty cycle increases, the delivery rate increases. This is because the nodes are active more often and for longer durations. When the duty cycle is low (below 4%), nodes do not share common activities that are sufficiently long to exchange frames, which reduces dramatically the delivery rate.

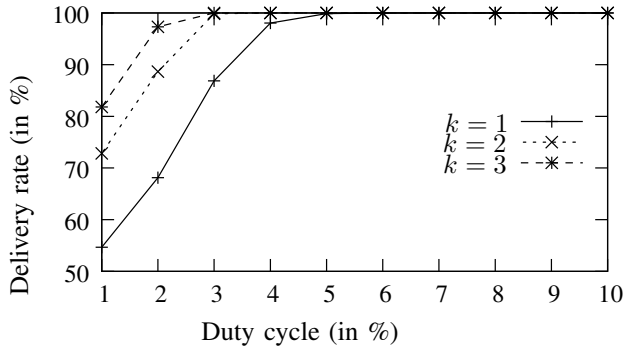


Figure 7. Delivery rate as a function of the duty cycle, for a varying number of possible next hops on a diamond topology.

Figure 8 shows the delay as a function of the duty cycle, for several number of possible next hops. When the duty cycle is high (above 5%), the delay is low as nodes are active often and for long durations. When the duty cycle is low (below 4%), the delay is high. The sudden drop of delay for a duty cycle of 1% with  $k=1$  is due to the fact that many frames are dropped with these parameters, and frames experiencing long delays are more likely to be dropped.

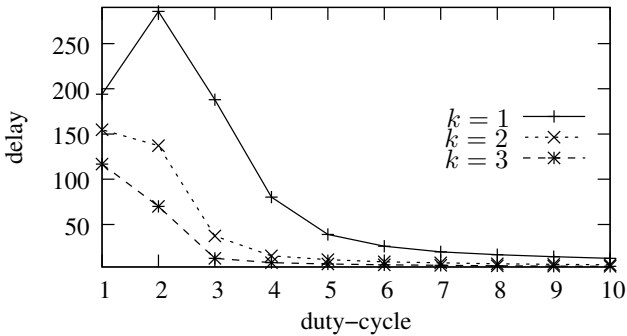


Figure 8. Delay of data frames as a function of the duty cycle, for a varying number of possible next hops on a diamond topology.

### C. Scenario 3: Evaluation of several possible next hops with very low duty cycles

In this third scenario, we evaluate the performance of our protocol when it is applied on very low duty cycles (between 0.1% and 1%). We consider larger periods, as WSNs with very low duty cycles are adapted to low traffic production.

Figure 9 shows the delivery rate as a function of the sending period, on a diamond topology with  $k=3$  possible next hops, and with a duty cycle of 1%. We varied the sending period from 8 s to 300 s. The results show that when the sending period is about 50s, the delivery rate reaches 100% for all number of fragmentations. Thus, it seems that the delivery rate is more impacted by the sending period than by the number of fragmentations.

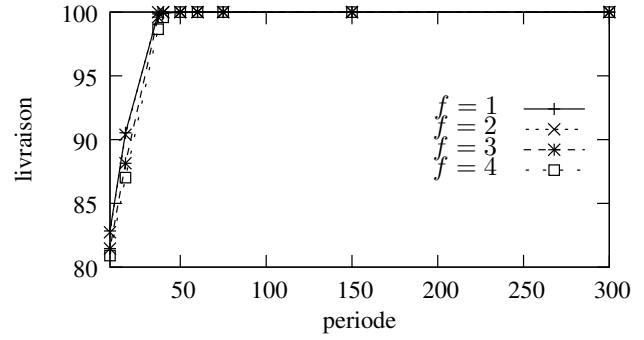


Figure 9. Delivery rate as a function of the sending period, for three possible next hops on a diamond topology, with a duty cycle of 1%, and for a varying number of fragmentations.

Figure 10 shows the delay as a function of the sending period, on a diamond topology with three next hops, and with a duty cycle of 1%. The delay is high for small sending periods (below 50 s), as the network becomes congested quickly with such small periods and duty cycles. The sudden drop in delay for a sending period of 8 s is due to the low delivery rate.

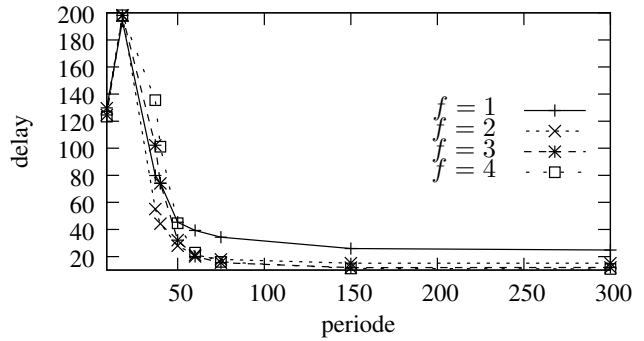


Figure 10. Delay of data frames as a function of the sending period, for three possible next hops on a diamond topology, with a duty cycle of 1%, and for a varying number of fragmentations.

Figure 11 shows the delivery rate as a function of the duty cycle, for very low duty cycles (between 0.1% and 1%). The results are obtained for a sending period of 300 s (that is, one packet is sent every 5 minutes), and for  $f=1$  and  $f=2$ . This large period of traffic generation is intended to model a realistic application running over a WSN that has a very low duty cycle. It can be noticed that when the duty cycle exceeds 0.2%, the delivery ratio is high. Increasing the number of fragmentations for this small duty cycle has a detrimental impact on the delivery ratio, as the nodes share common activities that are too short to be further reduced by increasing  $f$ . From this figure, it can be seen that our protocol achieves a throughput of about 0.8 bps (with a payload of 30 bytes every 300 s, and with a duty cycle of 0.2%), for a two-hop network. To the best of our knowledge, there is no study on the performance of unsynchronized MAC protocols for very

low duty cycles.

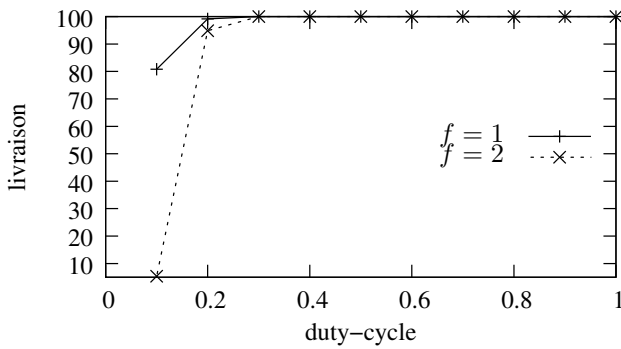


Figure 11. Delivery rate as a function of the duty cycle, for three next hops on a diamond topology, for a sending period of 300 s, and for a varying number of fragmentations.

Figure 12 shows the delay as a function of the duty cycle, for very low duty cycles (between 0.1% and 1%). The results are obtained for  $f = 1$  and  $f = 2$ , and a sending period of 300 s. The delay is small for duty cycles above 0.3%, and the delay is high when the duration of the node activities is too small, as it forbids communications. The sudden drop of the delay for  $f = 2$  and a duty cycle of 0.1% can be explained by the very important loss rate (as shown on Figure 11).

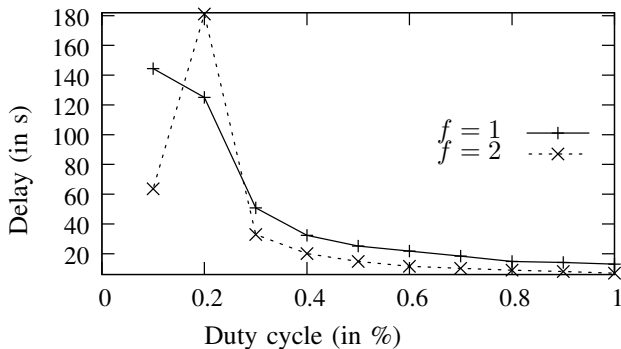


Figure 12. Delay of data frames as a function of the duty cycle, for three possible next hops on a diamond topology, for a sending period of 300 s, and for a varying number of fragmentations.

## V. CONCLUSION

Asynchronous MAC protocols are suitable for WSN applications which require low duty cycles. In this paper, we propose an asynchronous, blind MAC protocol based on random wake-up, and using a gradient-based approach to identify possible next-hops. Our protocol randomizes the activities of each node within each global cycle. It can achieve reasonable delay by further fragmenting the activities, without impacting too much the delivery ratio. Our protocol has many advantages, as it supports very low duty cycles (of less than 1%) and is adapted to dynamic network (as it does not need to maintain information about the topology). Extensive simulations carried

on NS-2 under various scenarios show that our protocol can achieve good performance in terms of delivery ratio and delay.

## REFERENCES

- [1] W. Geoffrey, J. Jeff, R. Mario, L. Jonathan, and W. Matt, "Monitoring volcanic eruptions with a wireless sensor network," in *European Workshop on Wireless Sensor Networks (EWSN'05)*, 2005.
- [2] J. Hart and K. Martinez, "Environmental sensor networks: A revolution in the Earth system science?" *Earth Science Reviews*, vol. 78, no. 3, pp. 177–191, 2006.
- [3] T. Nagayama, M. Ushitab, and Y. Fujino, "Suspension bridge vibration measurement using multihop wireless sensor networks," in *East Asia-Pacific Conference on Structural Engineering and Construction (EASEC)*. Elsevier, 2011, pp. 761–768.
- [4] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks," in *Ad Hoc and Sensor Networks*, April 2004.
- [5] 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.
- [6] M. I. V. Gallego and F. Rousseau, "Energy efficient neighborhood maintenance and medium access with wake on idle," in *IEEE INFOCOM*, April 2013, pp. 2931–2939.
- [7] N. Nordin and F. Dressler, "Effects and implications of beacon collisions in co-located IEEE 802.15.4 networks," in *IEEE Vehicular Technology Conference (VTC)*, Quebec City, Canada, September 2012.
- [8] 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.
- [9] R. Carrano, D. Passos, L. Magalhaes, and C. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," in *IEEE Communications Surveys & Tutorials*, June 2013, pp. 1–14.
- [10] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *ACM Sensys*, November 2004.
- [11] H. C. Kenneth, W. Parker, and A. Arora, "O-MAC: a receiver centric power management protocol," in *ICNP'06*, 2006, pp. 311–320.
- [12] P. Tom, H. Gertjan, B. Maarten, and L. Koen, "The MAC framework: redefining MAC protocols for wireless sensor networks," in *Wireless Networks*, February 2010, pp. 2013–2029.
- [13] 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.
- [14] 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.
- [15] 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 2013*, July 2013, pp. 412–417.
- [16] 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.
- [17] P. le Huy and S. Roy, "Low-power wake-up radio for wireless sensor networks," in *Mobile Networks and Applications*, 2010, pp. 226–236.
- [18] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *ACM MobiHoc*, 2003, pp. 35–45.
- [19] R. Carrano, D. Passos, L. Magalhaes, and C. Albuquerque, "Nested block designs: Flexible and efficient schedule-based asynchronous duty cycling," in *Computer Networks*, vol. 57. Elsevier, December 2013, pp. 3316–3326.
- [20] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar, "Random asynchronous wakeup protocol for sensor networks," in *International Conference on Broadband Networks (BroadNets 2004)*, October 2004, pp. 710–717.
- [21] B. P. and C.-K. Chau, "Opportunistic forwarding in wireless networks with duty cycling," in *CHANTS*, October 2008, pp. 19–26.
- [22] "Network simulator 2," 2002, <http://www.isi.edu/nsnam/ns>.