A Duty Cycle Self-adaptation Algorithm for the 802.15.4 Wireless Sensor Networks

Camila H. S. Oliveira, Yacine Ghamri-Doudane, Stéphane Lohier

▶ To cite this version:

Camila H. S. Oliveira, Yacine Ghamri-Doudane, Stéphane Lohier. A Duty Cycle Self-adaptation Algorithm for the 802.15.4 Wireless Sensor Networks. Global Information Infrastructure and Networking Symposium (GIIS'13), Oct 2013, Trente, Italy. 10.1109/GIIS.2013.6684349 . hal-01378125

HAL Id: hal-01378125
https://hal.science/hal-01378125

Submitted on 14 Mar 2017

A Duty Cycle Self-Adaptation Algorithm for the 802.15.4 Wireless Sensor Networks

Camila H. S. Oliveira
Université Paris-Est, LIGM
75420 Champs sur Marne, France
Email: souzaol@univ-mlv.fr

Yacine Ghamri-Doudane
Université Paris-Est, LIGM
75420 Champs sur Marne, France
Email: yacine.ghamri@univ-mlv.fr

Stéphane Lohier
Université Paris-Est, LIGM
75420 Champs sur Marne, France
Email: lohier@univ-mlv.fr

Abstract—**The IEEE 802.15.4 protocol is widely adopted as the standard for the physical and MAC layers of wireless sensor networks. Among other mechanisms, it implements a mechanism called duty cycle that defines the node's active time during the network lifetime. This paper proposes a dynamic beacon interval and superframe adaptation algorithm (DBSAA) that adjusts the network duty cycle through two MAC layer parameters: the Beacon Order (BO) and the Superframe Order (SO). The parameters adaptation is triggered by the changes in the traffic load (i.e. increase or decrease due to modification in the environment). Using DBSAA, the network coordinator adjust the BO and SO parameters based on four parameter estimations: the superframe occupation ratio, the collision ratio, the number of packets received by the coordinator, and the number of source nodes. Performance evaluation results show that the duty cycle adaptation taking into account the BO and SO values meets the trade-off defined by the application requirements and energy consumption while compared to two other protocols: the standard 802.15.4 protocol, which does not perform duty cycle dynamic adaptation; and the DSAA (Dynamic Superframe Adjustment Algorithm), which adapts the duty cycle by adjusting only the SO parameter.**

I. Introduction

In the last few years, the rapid and growing progress in the development of modern and new devices allowed the use of Wireless Sensor Networks (WSN) in a large range of applications. According to [1], the use of WSNs in real-life applications is also due to the advent of the IEEE 802.15.4 standard, which defines the physical and medium access control (MAC) layers of the WSN's protocol stack; and the ZigBee specification, which covers the WSN's network and application layers.

WSNs have been largely employed in many application domains, such as public safety, environmental monitoring and smart homes. This type of networks is featured by its resource restriction, mainly concerned with the nodes energy. Suitable to networks with energy limitations, the 802.15.4 protocol implements a mechanism named duty cycling. The duty cycle is employed when the 802.15.4 protocol operates in the beacon enabled mode. In this operation mode, a superframe structure is used in order to manage the duty cycle mechanism. The timeline is divided in Beacon intervals (BI), which in turn, are divided in two different periods (i.e. each BI): an active period which represents the Superframe Duration (SD) and an inactive period. The SD defined in each BI and the BI itself are calculated based on two MAC protocol parameters, the Superframe Order (SO) and the Beacon Order (BO), respectively. In this beacon mode, the relation between the SD and the BI defines the network duty cycle.

However, although this protocol works well regarding the energy, memory and processing consumption; it does not take into account the dynamic nature of the sensors activity. Without considering the external stimuli, the duty cycle adopted initially can lead to a decreased application quality.

Aiming to provide the network with the ability of handling the trade-off between saving energy and meeting the application demands in terms of delay and throughput, we propose a dynamic and self-adaptive algorithm named Dynamic Beacon interval and Superframe Adaptation Algorithm (DBSAA). DBSAA is able to change the duty cycle through the adjustment of the two MAC protocol parameters, BO and SO. Its main objective is to provide more flexibility in duty cycle adaptation in order to increase the network throughput when needed and decrease the end-to-end delay while keeping saving energy.

The remainder of this paper is organized as follows. In the next section, we discuss the related works. Section III presents the 802.15.4 WSNs basic description. Section IV discusses the problem approached in this work and the Section V presents the solution proposed. In the Section VI we have the carried out simulations and the obtained results, while Section VII concludes this paper.

II. Related Work

The literature presents different approaches in adaptive duty cycle control. Nonetheless, some of them change the duty cycle without analyzing the value of the parameters that define the duty cycle itself, as can be seen in [2]. On the other hand, there are a few works that adapt the duty cycle by changing only one of the MAC layer parameters, either BO or SO, [3] and [4]. As shown in this paper, this leads to limited impacts while targeting improving the throughput usage, end-to-end delay, and energy usage.

Hence, in [2], the authors propose a congestion control mechanism that implements an adaptive duty cycle to do resource and traffic control. Although the congestion control is accomplished, as they do not consider the BO and SO values in their duty cycle adaptation and the active time is limited by thresholds, this solution may impose in certain cases delays in the packet delivery.

Also, in [3], the authors adapt the duty cycle by changing the BO value. The only goal of the adaptation is to save energy through modifying the time that the nodes remain in sleep mode. Consequently, the needs for more or less active time (defined by the SO) to handle the network traffic are not considered.

Alternatively, the work presented in [4] changes only the SO parameter intending to decrease the energy consumption and to improve the channel utilization. In this case, the application needs are taken into account as the active time is adapted. However, the BO value defines how much time the nodes wait before sending their new packets. In this case, the protocol allows the network to save energy and to deliver the data; but as the BO is not modified, the packet delivery delay is increased.

There are also theoretical approaches proposed to define the optimal duty cycle. In [5], a Markov-based theoretical analysis is proposed with the aim of saving more energy and meeting the application requirements. Looking for defining the optimal duty cycle dynamically, a solution based on reinforcement learning is presented in [6]. These solutions founded on mathematical models show good accuracy related to the optimal values defined for the BO and SO parameters. However, they are computationally complex solutions and, to the best of our knowledge, they have not been implemented in real networks to prove their practical feasibility.

Despite the fact that the works described above perform duty cycle dynamic adaptation, they present deficits incurred by the choice of just one parameter, either BO or SO, in the DC definition or by their complexity limiting their practical implementability. In order to improve the profit produced by the duty cycle dynamic adaptation, we propose in this paper a dynamic and self-adaptive algorithm to control the duty cycle in an 802.15.4 wireless sensor network that is simple and acting on both BO and SO. In the next section, the MAC layer of the 802.15.4 WSN is described to help the understanding of the duty cycle mechanism.

III. IEEE 802.15.4 WSNs

The IEEE 802.15.4 standard is developed to be used in wireless communication networks made up with low-cost wireless devices. Therefore, the standard defines the physical and MAC layers of WSNs in order to deal with the node limitations (energy, memory and processing), [7]. The standard specifies the topology, the operation mode, the network capacity and the channel access protocol (CSMA/CA). In the IEEE 802.15.4 based WSNs, the nodes can be classified as Full Function Devices (FFD) or Reduced Function Devices (RFD). The FFD node can function as a personal area network (PAN) coordinator, as a router or as a regular device. The RFD nodes are more limited computationally and can communicate only with an FFD node.

The IEEE 802.15.4 MAC layer can operate in either beacon-enabled or non-beacon-enabled modes. In this work, we are interested only to the beacon-enabled mode. As can be observed in Figure 1, in the beacon-enabled mode, the network operation timeline is divided into consecutive time intervals called Beacon Intervals (BI). The BI uses a structure called superframe to regulate the packet transmission and the nodes state. Hence, the superframe can be divided in two periods, active and inactive periods. The BI defines the entire period of a superframe, and the Superframe Duration (SD) defines the active period duration. The active period (represented by the white rectangle in the Figure 1) consists of 16 time slots with the same length which can be further divide in a Contention Access Period (CAP) and a Contention Free Period (CFP) portions. The slotted CSMA/CA protocol is used to control the node access to the channel in the CAP period.

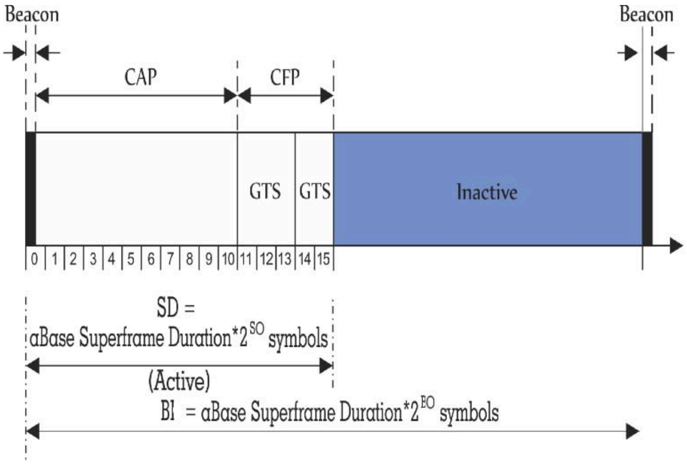

Beacon
Beacon
CAP
CFP
GTS
GTS
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
SD = aBase Superframe Duration*2^SO symbols
(Active)
BI = aBase Superframe Duration*2^BO symbols

Fig. 1: Super frame structure.

The BI starts with the dispatch of a beacon message, called Beacon Frame (BF), by the coordinator. The BF mark the beginning of the BI as can be observed in Figure 1. It is sent in the slot 0 of the CAP period and contains information related to the superframe, such as the Beacon Interval (BI) and the Superframe Duration (SD). The BI, SD and DC values are calculated according to the equations below, where BO (*macBeaconOrder*) and SO (*macSuperframeOrder*) are MAC layer parameters.

$$BI = 2^{BO} \times aBaseSuperFrameDuration, \quad 0 \leq B0 \leq 14 \tag{1}$$

$$SD = 2^{SO} \times aBaseSuperFrameDuration, \quad 0 \leq SO \leq BO \leq 14 \tag{2}$$

$$aBaseSuperFrameDuration = aBaseSlotDuration \times aNumSuperframeSlot \tag{3}$$

$$DC = \frac{SD}{BI} \tag{4}$$

The *aBaseSuperFrameDuration* is calculated using equation 3, where *aBaseSlotDuration* is equal to 60 symbols and *aNumSuperframeSlot* is equal to 16, according to the standard. The standard defines the slot size in symbol number but it can be easily converted to time in seconds.

IV. Problem Description

The 802.15.4 standard was developed to take into account the energy, memory and processing limitations of WSNs. If we consider the energy problem, the protocol saves energy putting the network into sleep mode during some periods through the duty cycle mechanism. Nevertheless, the standard does not offer means to adapt his duty cycle which is statically set up.

Choosing a duty cycle in runtime is a complicated task and depends on the network objectives. Superficially, we know that setting a low duty cycle means a low energy consumption whereas a high duty cycle means that a bigger number of packets can be transmitted but with greater overall energy consumption. It is thus clear that there is a need for self-adapting the duty cycle to better accommodate the throughput needs and the energy constraints. As the duty cycle is defined in function of the BO and SO values, these two parameters play a significant role in this self-adaptation.

Considering the BO and SO values, we can easily compute the maximum energy quantity which will be consumed, the maximum number of packets that can be served as well as the achievable end-to-end delay. Hence, for instance, if a duty cycle of 50% with BO = 2 and SO = 1 is set, as the difference between BO and SO is small, the nodes are able to save energy whereas avoiding important delays in the packet delivery as BO is small. However, if the duty cycle of 50% with a BO = 10 and SO = 9 is defined, we have a long sleep time and, even if all packets are delivered during the active time, there will be a significant delay between packet delivery which can be problematic for the intended application. Further, it can still bring on high contention between the nodes at the beginning of the following beacon interval causing energy waste.

Aiming to avoid the problem of adapting the DC only thinking about the percentage which represents the nodes activity time, we consider in this work the duty cycle adjustment through finding appropriate values for the BO and SO parameters such that the energy consumed by the nodes is minimized while the application requirements are fulfilled.

V. DBSAA Algorithm

This section describes the solution proposed called DBSAA (Dynamic Beacon interval and Superframe Adaptation Algorithm). Before that, let us first define the considered network assumptions. Then, the methodology used to estimate the variables that characterize the network traffic are expound. Finally, we describe DBSAA operation.

A. Assumptions

In order to develop our algorithm, we made some assumptions on the network operation mode and configuration. At first, the algorithm is proposed for 802.15.4 network using the beacon-enabled operation mode in a star topology. After that, we decide to consider only the CAP portion in the active period. This choice is based on the same justifications as those presented in [8], such as, the limitation in the number of GTS slots, which can be allocated in each beacon interval (at most seven), and the fact that the GTS slots are optional in an 802.15.4-based network.

B. Methodology

The next paragraphs describe the three steps defined in our methodology to estimate the information used by the DBSAA algorithm, and also the details of the algorithm operation.

The methodology chosen in this work is based on two premises. The first one assumes that the packet collision is directly related to the number of sensor nodes sending data. The second one specifies the relation between the number of packets received by the coordinator and the superframe occupation. The DBSAA algorithm deliberates upon theses two premises to decide when and how to change the BO and SO values. Besides, the algorithm is executed only by the coordinator and utilizes the information estimated in it. Therefore, it is not necessary to create new messages and modify the 802.15.4 standard.

In order to get the needed information to make a decision, DBSAA follows the three steps described below:

- Step 1: DBSAA estimates network load based on the number of packets received by the coordinator and on the number of nodes that sent packets during a measurement window enclosing the N last beacon intervals.
- Step 2: DBSAA determines if there was changes in the network load during the N last BI and delineates the type of the variation arisen.
- Step 3: DBSAA calculates a factor α which assigns the number of beacon intervals that the coordinator should wait before triggering the DBSAA algorithm. A factor α is defined to avoid either a too long or a too short waiting time to perform DBSAA.

The step one is easily performed through simple measurement made by the coordinator over the last N intervals. After, in the step two, the coordinator computes the superframe Occupation Ratio (OR) and the Collision Ratio (CR) using the information collected in the first step. In DBSAA, the OR and CR, which estimate the current network load, are computed by the equations (5), (6) and (7):

$$OR = \frac{numPkt \times Tmp_{trans}}{2 \times SD} \tag{5}$$

$$Tmp_{trans} = \frac{tamPkt}{250} + T_{ack} + ACK + IFS + (2 \times CCA) \tag{6}$$

$$CR = \frac{(numN \times NumPSD \times 2) - numPkt}{numN \times NumPSD \times 2} \tag{7}$$

In equation (5), the $numPkt$ is the sum of packets received by the coordinator during the last N beacon intervals and Tmp_{trans} is the time necessary to transmit a data packet. In Tmp_{trans} we do not consider the backoff time because it has no influence in the overall occupation ratio, since when the nodes are performing a backoff, they stay in idle mode which allows the other nodes to send their packets. In equation (6), the *tamPkt* is the packet size in bits, T_{ack} is a turn around time waited by a node before sending an ACK, the *IFS* is the time that separates two successive frames transmitted from a device and the *CCA* is the time taken to realize the clear channel assessment procedure that verifies if the channel is available. In equation (7), the $numN$ is the number of nodes who sent packets in the last N intervals and $NumPSD$ is the number of packets that the source nodes generate during a SD. The $NumPSD$ is calculated based on the data rate used in the network.

In addition, in step two, two more variables have their values set. These are: $change\_numPkt$ and $change\_numNodes$. These variables represent the direction of changes related to the number of nodes and packets. At the end of step two, these variables can assume two different values: 1 when the number of packets/nodes increases and 0, otherwise. According to these values, the coordinator can detect the traffic load changes.

In the last step, the factor α is computed as stated by equation (8). This factor is responsible of adapting the number of waited intervals ($num\_interval$) by the coordinator to alter the BO and SO values. α has a different impact depending on the BO value. If we have a large BO, the BI also assumes a large value. Considering the computed value of α, it may take a long time until the coordinator make the duty cycle adaptation, which can result in delays in the packet delivery. On the other hand, if BO assumes a small value, the BI becomes very short and the frequent checking of the network load may result in unnecessary consumption of energy. The α adaptation is thus an important step in DBSAA performance. In addition, equation (8) considers an OR factor, β, which represents the occupation ratio and helps decreasing the α value when there is an intensive traffic in the network. See Table I.

TABLE I: β values.

OR	0% - 25%	26% - 50%	51% - 75%	76% - 100%
β	4	3	2	1

$$\alpha = \frac{(15 - BO) + \beta}{BO} \tag{8}$$

C. DBSAA Algorithm

After accomplishing the steps described above, the coordinator can call the DBSAA algorithm to select the BO and SO adjustments that are the more adequate to the network state. In order to take into consideration the application requirements, DBSAA uses two thresholds: the THoccupation and the THcollision. These thresholds are used in the algorithm to characterize the traffic behavior and are defined by the application according to its needs. They represent the application sensitivity to delay and packet losses. The detailed operations of DBSAA are presented in Algorithm 1.

Initially, the $num\_interval$ value is checked (line 2). If the $num\_interval$ reaches α, the network load is verified through examining the $change\_numPkt$ variable (line 3). When $change\_numPkt$ value is 0, it may signify two different situations: either the nodes had no packets to send, or they could not send any because of collisions. At first, we try to eliminate the possibility of collision. For that, the channel state is checked out (line 4). If the channel is busy, it means that collision probability is high. In this case (lines 5-6), if the $BO - SO = 1$ the BO is decremented to avoid a high contention after a large interval. Otherwise (lines 8-10), if the $BO - SO > 1$, DBSAA not only decrease the BO value but also increase the SO value to give more time for packet transmissions. On the other hand, if the channel is not busy, the OR is verified to assess the traffic conditions (lines 14-16). If the OR is smaller than THoccupation, the BO and SO values are not modified. Indeed, in this case the DBSAA assumes that we do not expect more packets to arrive as the traffic load is tending to become smaller. In case OR is larger than THoccupation, and if SO is smaller than BO, SO is increased. Indeed, increasing SO and thus SD, gives more time for the nodes to send their packets, which improve the delivery rate.

Considering the case where $change\_numPkt$ is 1 (line 18), the next step is to monitor the $change\_numNodes$ value (line 19). If $change\_numNodes$ is 1, it might indicate an increase in the traffic intensity due to a growth in the number of nodes sending data. In this case (lines 20-24), the CR is compared to THcollision. If THcollision is not exceeded, BO and SO remain unchanged. Otherwise, SO is updated according to the difference between BO and SO (since SO is limited by the BO value). If BO is larger than SO, SO is increased in order to decrease the contention among the nodes to access the channel. Therewith, the nodes will have more time to send their packets and, as the active time becomes longer, they will need a fewer number of slots to send their packets. As result, the collisions will decrease. In the case where the $change\_numNodes$ is 0 (line 25), the DBSAA assumes that the number of packets increased due to the rise in the rate of sent packets by the sources. In this case, the active time in the beacon interval has to be increased to meet the application's requirements. For that, the algorithm ascertains the CR and OR with relation to the THcolision and THoccupation to have an evaluation of the network load (lines 26). If one of the thresholds is exceed, the DBSAA checks if SO is smaller than BO, and in that case SO is increased by 1. In this way, the nodes stay active for more time whereas the BO remains the same to prevent an increase of delay in the packet delivery. If not, as SO is limited by BO, BO and SO are both increased, giving more active time to the nodes to transmit their data.

VI. Performance Evaluation

This section describes the DBSAA performance evaluation using NS-2 (Network Simulator 2). We simulate the IEEE 802.15.4 standard and the DSAA algorithm [4] to compare with DBSAA. In our implementation, DSAA underwent certain improvements as some details are not described in the original paper where it is proposed. These improvements are restricted to the definition of its collision and occupation ratio. Apart from these, the algorithm operation remains exactly the same as presented in [4]. The DSAA ratios simulated are the same than those used in our work, except that the DSAA ratios are based on the information of every interval while, in our algorithm, they lie on the feedback of the N last intervals. In DBSAA simulation, N is set to 2. We chose to keep the history of the two last BI because our simulations showed that

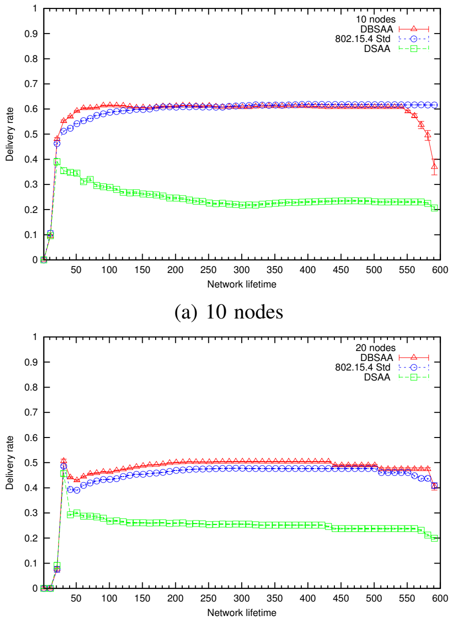
Algorithm 1 DBSAA Algorithm.

```
At each beacon interval
if $num\_interval$ = $\alpha$ then
  if $change\_numPkt$ == 0 then
    if $channel\_status = BUSY$ then
      if $BO - SO = 1$ then
        $BO \leftarrow BO - 1$
      else
        if $BO - SO > 1$ then
          $BO \leftarrow BO - 1$
          $BO \leftarrow SO + 1$
        end if
      end if
    else
      if $OR >= THoccupation$ and $BO <= 14$ and $SO < BO$ then
        $SO \leftarrow SO + 1$
      end if
    end if
  else
    if $change\_numNodes == 1$ then
      if $CR > THcollision$ then
        if $BO - SO >= 1$ then
          $SO \leftarrow SO + 1$
        end if
      end if
    else
      if $OR > THoccupation$ and $CR > THcollision$ then
        if $SO < BO$ then
          $SO \leftarrow SO + 1$
        else
          $SO \leftarrow SO + 1$
          $SO \leftarrow BO + 1$
        end if
      end if
    end if
  end if
end if
```

this duration was sufficient to detect a significant change in the network load.

The performed simulations consider a network with a maximum link throughput of 250Kb/s. The network is configured in a star topology. Our traffic model tries to model the dynamism and the unpredictability present in the traffic generated by WSN applications in the context of smart-* (home, building, etc.). In order to achieve that, we simulate networks with 4 different number of nodes (10, 15, 20 and 25) where the number of sources nodes sending packets changes dynamically all over the simulation time (the start and stop time of each traffic is chosen randomly which results in different number of source sending packets during the network lifetime) and the simulation of each network is performed 33 times. The value 33 is chosen because, according to [9], 30 is the minimum amount of samples required to find a reliable confidence interval. are maintained and the power consumption in each mode operation utilized by the NS-2 simulator is showed in the Table I.

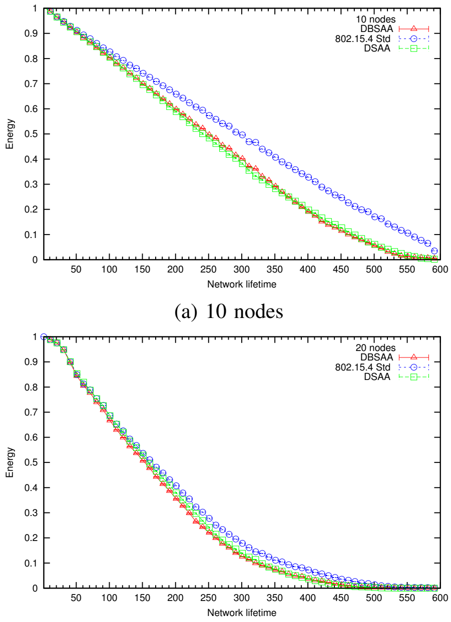
Aiming to demonstrate the benefits of adapting the BO and SO parameters to achieve more appropriate duty cycles with regard to the application requirements and the power saving, we compare the performance of the three algorithm regarding to three metrics: the average energy consumption, the data delivery ratio and the end-to-end delay. In order to consider a general scenario, not so restrictive, the simulations are realized using a THoccupation value of 75% and a THcollision value of 30%.



(a) 10 nodes

(b) 20 nodes

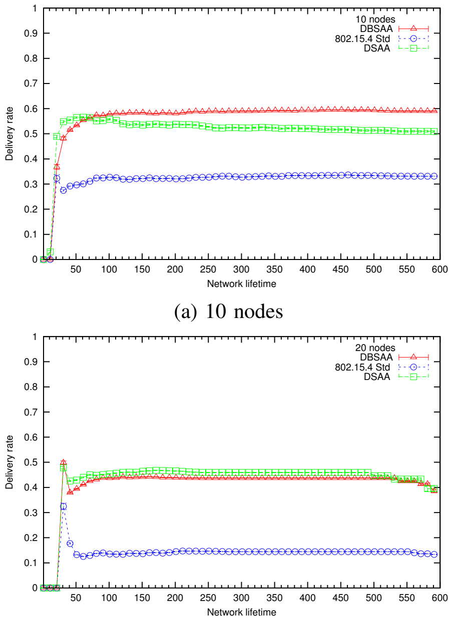
Fig. 2: Delivery rate DC 50% (BO = 2, SO = 1).



(a) 10 nodes

(b) 20 nodes

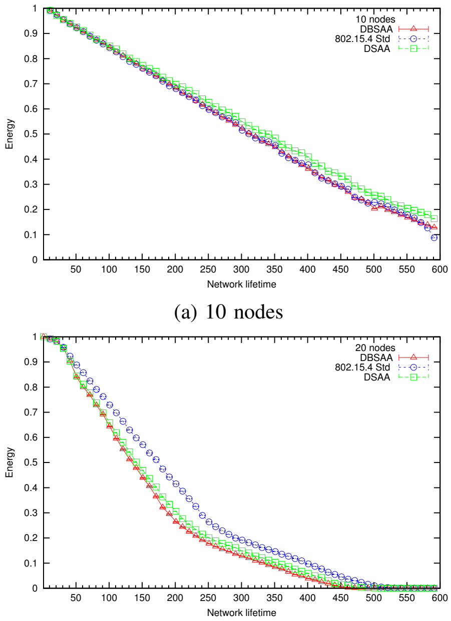
Fig. 3: Energy average DC 50% (BO = 2, SO = 1).



(a) 10 nodes

(b) 20 nodes

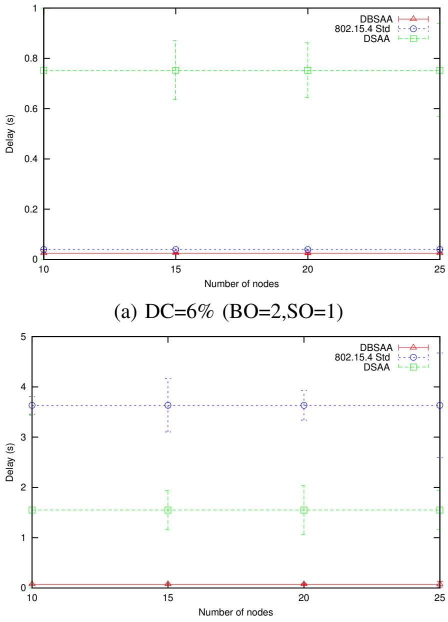
Fig. 4: Delivery rate DC 6% (BO = 6, SO = 2).



(a) 10 nodes

(b) 20 nodes

Fig. 5: Energy average DC 6% (BO = 6, SO = 2).



(a) DC=6% (BO=2,SO=1)

(b) DC=50% (BO=6,SO=2)

Fig. 6: End-to-end delay.

Figures 2 - 5 show the simulation results for the scenario with 10 and 20 nodes experiencing different duty cycles. Figure 2 represents more precisely the delivery rate result of WSNs having a DC of 50%. In this case, we have BO=2 and SO=1. This figure shows that the 802.15.4 Std and DBSAA perform better than DSAA. This is explained by the fact that DSAA operations are based on the SO adjustment. As DSAA cannot adapt the BO value to save energy, it tries to adapt the SO. However, the SO is responsible to define the active time and, for that, some changes in this parameter result in packet losses. Besides, in this scenario the difference between BO and SO is too small to allow good improvements by only adjusting the SO value. In addition, we remark that in a situation where the network has a high duty cycle and the gap between the BO and SO values is small, DBSAA performs well keeping the same delivery ratio presented by the 802.15.4 Std. The drop in the curve of delivery rate presented by the DBSAA in the Figure 2b is explained by the drop in the energy curve in the Figure 3b. As can be seen in this curve, the nodes energy in the DBSAA run out their energy few seconds earlier than in the 802.15.4 std.

In the Figures 4 and 5, which represent a scenario where the DC is reduced and the gap between the BO and SO increases, the dynamic adaptation of BO and SO leads to important improvements related to the application demand. In Figure 4 we can see that DBSAA and DSAA achieve the delivery of two times more packets than the 802.15.4 Std. This outcome is explained by the fact that the delivery ratio is directly related to the SO value, and both, DBSAA and DSAA, perform SO adaptation. So, the SO adjustment gives more capacity to handle the requested data load. Regarding the energy consumption, Figure 5 confirms that the dynamic adjustment of the MAC parameters does not imply much more energy waste. In Figures 3 and 5, we can see that DBSAA and DSAA, in some cases, waste a little more energy than the 802.15.4 Std. However, DBSAA compensates this energy waste by the accomplished low end-to-end delay presented in Figure 6.

Indeed, Figure 6 makes clear that adapting just the SO can produce a higher end-to-end delay than the result obtained through the adaptation of the two parameters. In brief, the graphics show that even in a situation where the BO and SO can not vary widely (Figure 6a), DBSAA achieves good results in terms of energy consumption and end-to-end delay whereas DSAA suffers the consequences of taking into account only the SO parameter in the duty cycle adaptation. This fact is even more evident in Figure 6b, which shows the end-to-end delay result in a scenario where the BO and SO can vary in a larger range of values. DBSAA shows the relevance of looking for the balance between the BO and SO to fulfill the application needs by depicting the smaller end-to-end delay among the three protocols. To sum up, we can say that DBSAA achieve a very good cost-benefit by performing duty cycle adaptation (i.e. in terms of the trade-off between the energy consumption and end-to-end delay).

VII. Conclusion

This paper proposes a solution to the duty cycle adaptation in WSNs through the dynamic adjustment of the BO and SO values. Our algorithm, called DBSAA, takes into account the trade-off among the application requirements in terms of bandwidth and energy consumption in the DC definition. DBSAA does not imply any modification to the 802.15.4 standard and the dynamic adaptation at the coordinator level is performed based on traffic load estimation.

Performance evaluation were realized considering the WSN energy consumption as well as two other metrics that determine the application quality, the data delivery ratio and the end-to-end delay. Simulation results showed that our algorithm, DBSAA, performs better than the legacy 802.15.4 Std as well as another algorithm from the literature, DSAA, considering different scenarios. Hence that, we show in the situation where the initial DC is really bad, DBSAA basically keeps the same delivery ratio presented by DSAA and doubles the delivery ratio presented by 802.15.4 Std while shows the lowest delay among the three protocols. On the other hand, in the situation where a good DC is set, DBSAA keeps the delivery ratio as in the 802.15.4 Std while it performs expressively better when compared to DSAA. Regarding to the end-to-end delay, DBSAA keeps the good performance presented in the first scenario achieving a significantly small delay compared to the other two protocols. With such an approach, the network accomplishes to find a duty cycle that balances the energy consumption and the application requisite, such as the delay and throughput. Besides, our solution achieves the goal initially proposed without changing the 802.15.4 standard and in an entirely dynamic way. The next step in this work will be to implement DBSAA in a real wireless sensor network testbed and enhance the solution to work in a peer-to-peer topology (i.e. topologies that does not use coordinator nodes).

References

[1] G. Anastasi, M. Conti, M. Di Francesco, and V. Neri, "Reliability and energy efficiency in multi-hop ieee 802.15.4/zigbee wireless sensor networks," in *IEEE Symposium on Computers and Communications (ISCC)*, June 2010, pp. 336–341.

[2] D. Lee and K. Chung, "Adaptive duty-cycle based congestion control for home automation networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 1, pp. 42–47, February 2010.

[3] M. Neugebauer, J. Plonnigs, and K. Kabitzsch, "A new beacon order adaptation algorithm for ieee 802.15.4 networks," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, January 2005, pp. 302–311.

[4] B.-H. Lee and H.-K. Wu, "Study on a dynamic superframe adjustment algorithm for ieee 802.15.4 lr-wpan," in *IEEE Vehicular Technology Conference (VTC)*, May 2010, pp. 1–5.

[5] M. Khanafer, M. Guennoun, and H. Mouftah, "Adaptive sleeping periods in ieee 802.15.4 for efficient energy savings: Markov-based theoretical analysis," in *IEEE International Conference on Communications (ICC)*, June 2011, pp. 1–6.

[6] R. de Paz Alberola and D. Pesch, "Duty cycle learning algorithm (dcla) for ieee 802.15.4 beacon-enabled wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 664–679, June 2012.

[7] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs) (Amendment of IEEE Std 802.15.4-2003)*, IEEE Std. 802.15.4, 2006.

[8] J.-H. Lim and B. T. Jang, "Dynamic duty cycle adaptation to real-time data in ieee 802.15.4 based wsn," in *IEEE Consumer Communications and Networking Conference*, January 2008, pp. 353–357.

[9] R. Jain, *The Art of Computer Systems Performance Analysis : Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley, New York, 1991.