Strategies for Optimal MAC Parameter Setting in IEEE 802.15.4 Wireless Sensor Networks: a Performance Comparison

Simone Brienza[1], Domenico De Guglielmo[1], Giuseppe Anastasi[1], Marco Conti[2], Vincenzo Neri[3]

[1]Dept. of Information Engineering
University of Pisa, Italy
{firstname.lastname}@iet.unipi.it

[2]IIT-CNR
National Research Council, Italy
marco.conti@iit.cnr.it

[3]Ingegneria Dei Sistemi
IDS Corporation, Italy
v.neri@idscorporation.com

Abstract — **Recent studies have shown that the IEEE 802.15.4 MAC protocol may suffer from severe limitations in terms of reliability and energy efficiency if a non appropriate parameter setting is used. Hence, a number of solutions have been proposed to select the optimal parameter setting to provide reliability with minimum energy consumption. In this paper we compare, by simulation, three different algorithms that take different approaches to the problem, namely *offline computation*, *model-based adaptation*, and *measurement-based adaptation*. We show that adaptive algorithms perform well, however the model-based adaptive approach has some limitations that make it unsuitable in practical scenarios, where operating conditions may vary over time and transmission errors cannot be neglected. Instead, the measurement-based adaptive approach is flexible and effective.**

Keywords: IEEE 802.15.4, CSMA/CA, Energy Efficiency, Reliability, MAC Parameters Tuning

I. INTRODUCTION

Wireless sensor networks (WSNs) are being used for an increasing number of applications in different domains, ranging from environmental monitoring to healthcare, from logistics to industrial applications, from location/tracking to automatic building management. This trend is pushed by the adoption of two communication standards for WSNs, namely the IEEE 802.15.4 standard [1] and ZigBee specifications [2], defining the low and high layers of the protocol stack, respectively. Usually, *energy efficiency* is the major constraint in the design of WSN-based systems as sensor nodes are typically powered by batteries, with limited power budget, that cannot be replaced, or recharged, due to environmental/cost constraints. However, in many application scenarios, additional requirements need to be considered, such as *reliability* and *timeliness* [3].

A number of studies have emphasized that the IEEE 802.15.4 MAC protocol suffers from severe limitations in terms of *reliability* (i.e, *packet delivery probability*) which make it unsuitable for many application scenarios [4]. This is mainly due to the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) algorithm used by the 802.15.4 MAC protocol for channel access, which is not able to efficiently handle contention when the number of simultaneously contending nodes is relatively high. This is a common problem for all contention-based MAC protocols (a similar problem does not occur when using a time-division or polling scheme). However, in the 802.15.4 MAC it is made more severe than in other similar MAC protocols due to the CSMA/CA parameter values specified by the standard. Specifically, the default parameter setting suggested in the standard is not appropriate for WSNs, even when the number of sensor nodes is very low. Furthermore, in some scenarios a 100% reliability can be guaranteed only using high parameter values, formally not allowed by the standard [4]. In general, the optimal CSMA/CA parameter setting, i.e. the set of CSMA/CA parameters that is able to provide the reliability required by the application, with the minimum energy consumption, strongly depends on the operating conditions experienced by the sensor network (i.e., offered load, packet error rate, etc.). Hence, a number of strategies have been proposed to derive the optimal CSMA/CA parameter setting. They range from *offline computation* [4] [5], to *model-based adaptation* [6] and *measurement-based adaptation* [7].

In order to better understand pros and cons of the various approaches, and help the designer in choosing the most effective solution, in this paper we compare several algorithms that take different approaches to the optimal parameter-setting problem. We investigate, through simulation, their capability to satisfy application requirements, in terms of reliability. However, we also consider their energy efficiency and ability to adapt to time-varying operating conditions. Our simulation results highlight the effectiveness and flexibility of the *ADaptive Access Parameter Tuning (ADAPT)* algorithm proposed in [7]. Motivated by these results, we implemented *ADAPT* in a real sensor network and carried out an experimental analysis to confirm its suitability to operate in a real environment. The experimental results confirmed the simulation results.

The rest of this paper is organized as follows. Section II introduces three different algorithms for the optimal parameter setting in IEEE 802.15.4 sensor networks. Section III compares the considered algorithms both in stationary and dynamic scenarios. One of these algorithms (ADAPT) is implemented in Section IV. Finally, conclusions are drawn in Section V.

II. CSMA/CA PARAMETER SETTING

IEEE 802.15.4 [1] is a standard for low-rate, low-power, and low-cost Personal Area Networks (PANs). It defines two different channel access methods, namely a *beacon enabled* mode and a *non-beacon enabled* mode. In this paper, we will focus on the beacon-enabled mode only since it is the most common mode used in sensor networks. Also, for the sake of simplicity, we will refer to a star network topology. In the beacon-enabled mode a slotted

CSMA/CA scheme is used for channel access whose behavior is regulated by four parameters that are summarized in Table I, along with the corresponding values allowed by the standard (details about the 802.15.4 CSMA/CA algorithm can be found in [8]). The problem to address is: how to select the *optimal* parameter setting, i.e., the set of values that can provide the reliability level required by the application with the minimum energy consumption for the sensor nodes (under time-varying operating conditions). To solve this problem three solutions have been proposed [5] [6] [7], that will be investigated throughout the paper. They take different approaches, namely *model-based offline computation* [5], *model-based adaptation* [6], and *measurement-based adaptation* [7]. Although the algorithm in [7] can be used also in a multi-hop topology, to make the comparison fair, throughout we will refer to a star topology.

TABLE I. 802.15.4 CSMA/CA PARAMETERS AND ALLOWED VALUES [1].

Parameter	Values	Description
macMaxFrameRetries	Range: 0-7 Default: 3	Maximum number of retransmissions
macMaxCSMABackoffs	Range: 0-5 Default: 4	Maximum number of backoff stages -1
macMaxBE	Range: 3-8 Default: 5	Maximum backoff window exponent
macMinBE	Range: 0-7 Default: 3	Minimum backoff window exponent

A. Model-based Offline Algorithm

The solution proposed in [5] takes an offline computation approach and leverages on an analytical model of the IEEE 802.15.4 CSMA/CA algorithm, based on a Discrete Time Markov Chain (DTMC). The details of the analytical model are omitted here for the sake of space (the reader can refer to [5]). We just mention that it considers a sensor network with a star topology and assumes that the operating conditions (number of sensor nodes, packet size) are known. In addition, the wireless medium is assumed to be ideal. Under these hypothesis, the analytical model is able to provide, for any CSMA/CA parameter setting, the following performance indexes: (**i**) packet delivery ratio, (**ii**) average packet latency, and (**iii**) average energy consumed by a sensor node for correctly delivering a packet to the sink. Then, the optimal parameter setting can be easily identified as the setting capable of providing the delivery ratio (and, possibly, latency) required by the application with the minimum energy consumption.

In practice, the effectiveness of the algorithm is strongly influenced by the accuracy of the analytic model it relies on. In other words, the parameter setting computed by the algorithm may not be optimal if the underlying model is not accurate. Furthermore, we need to point out that the analytical model developed in [5] assumes that the communication channel is ideal (i.e., transmission errors never occurs). In addition, it assumes the following packet generation process: (i) at any sensor node, after a frame transmission (and at startup time) a new packet is generated with a probability equal to (1-q_0), while, after a frame transmission, the transmission queue remains empty with probability q_0; (ii) if there are no packets to transmit, the node goes to the *idle* state. While in this state, a new packet is generated with probability (1-q_0).

From assumptions (i) and (ii) it follows that q_0=0 corresponds to a saturated traffic condition, i.e., all sensor nodes always have a packet ready for transmission.

B. Model-based Adaptive Algorithm

The model-based algorithm described in the previous section works offline and, in addition, it requires to know the operating conditions of the sensor network. Hence, this solution is not suitable for real environments where operating conditions typically vary over time and cannot be predicted in advance. Finally, the algorithm requires a significant amount of time to solve the model and to derive the optimal parameter setting. To overcome these limitations a model-based adaptive algorithm is proposed in [6] by the same authors. The latter algorithm is still based on a DTMC model of the sensor network. However, the model is a simplified version of the one developed in [5] and, thus, it can be solved by sensor nodes with limited computational and memory resources. In particular, sensor nodes estimate some congestion indexes through online measurements, instead of deriving them analytically from the model. At the beginning of each Beacon Interval a generic sensor node performs the following steps: (i) estimates the probability to find the channel busy during the first and second CCA (referred to as α and β, respectively), as well as the probability τ that the node will attempt the first CCA in a generic backoff slot; to this end, it uses local measurements collected in previous Beacon Intervals, (ii) introduces α, β, and τ in the analytical model and derives delivery ratio, average packet latency and energy consumption, for any parameter set, (iii) selects the optimal parameter setting.

To improve the accuracy of the estimates derived at step (i), each index is estimated on the basis of m samples (collected in previous Beacon Intervals), using the moving average method. Since the previous actions are repeated at each Beacon Interval, the algorithm is able to react to possible changes in the operating conditions.

C. Measurement-based Adaptive Algorithm

The main limitation of a model-based approach is that (**i**) its effectiveness strongly depends on the accuracy of the related model, and (**ii**) it typically takes a long time to solve the model and provide the optimal solution. An alternative strategy is using a measurement-based approach, as in the *ADaptive Access Parameters Tuning* (ADAPT) algorithm presented in [7]. ADAPT is a heuristic algorithm that dynamically adapts the CSMA/CA parameter setting, based on local measurements of the performance index that must be guaranteed. The algorithm presented in [7] focuses on reliability and tries to guarantee a packet delivery probability d^{des}, specified by the application, with minimum energy consumption. In [7] it is shown, that the packet delivery probability increases monotonically with the minimum contention window size ($macMinBE$), the number of backoff trials ($macMaxCSMABackoffs$), and the number of retransmissions ($macMaxFrameRetries$).

However, increasing $macMaxCSMABackoffs$ results in a higher energy consumption than increasing $macMinBE$, while increasing $macMaxFrameRetries$ results in a higher energy consumption than $macMaxCSMABackoffs$ [7]. On the basis of these results, ADAPT dynamically increases and decreases the value of the above-mentioned parameters, depending on the measured delivery probability, which also depends on the time-varying operating conditions. Specifically, on the basis of d^{des}, ADAPT identifies the following two thresholds that define the reliability region within which the delivery ratio should be confined:

$$d^{low} = d^{des} \cdot (1 + \sigma)$$
$$d^{high} = d^{des} \cdot (1 + \sigma + \gamma), \qquad \sigma, \gamma \in [0,1]$$

Then, at each Beacon Interval *n*, the algorithm estimates the current delivery probability d_n^{est} experienced by the sensor node as follows

$$d_n^{est} = \delta \cdot d_{n-1}^{est} + (1 - \delta) \cdot d_n^{meas} \qquad (1)$$

where d_n^{meas} is the delivery probability measured during the *n*-th Beacon Interval (ratio between number of acknowledged packets and number of transmitted packets), and δ is a memory factor in the range [0,1]. To guarantee the reliability constraint d^{des}, ADAPT compares the estimated delivery probability d_n^{est} against the two thresholds d^{low} and d^{high}, and applies the tuning strategy described in Algorithm 1 to compensate possible variations in the delivery probability, due to congestion. ADAPT also includes an additional module (not shown here for the sake of space) to contrast the effects of packets dropped due to transmission errors. Basically, each sensor node also measures the packet loss probability l_n^{est} due to transmission errors, using an approach similar to (1). If $(1 - l_n^{est}) < d^{des}$ it is not possible to guarantee the required delivery ratio even using the maximum values for $macMinBE$ and $macMaxCSMABackoffs$. Hence, the retransmission mechanism must be enabled.

Algorithm 1: ADAPT

```
If $d_n^{est} < d^{low}$ then
    If $macMinBE < macMinBE^{max}$ then
        min ($macMinBE + 2, macMinBE^{max}$)
    Else If $macMaxCSMABackoffs < macMaxCSMABackoffs^{max}$ then
        min ($macMaxCSMABackoffs + 2$ , $macMaxCSMABackoffs^{max}$)
Else If $d_n^{est} > d^{high}$ then
    If $macMaxCSMABackoffs > macMaxCSMABackoffs^{min}$ then
        $macMaxCSMABackoffs - 1$
    Else If $macMinBE > macMinBE^{min}$ then
        $macMinBE - 1$
```

III. PERFORMANCE COMPARISON

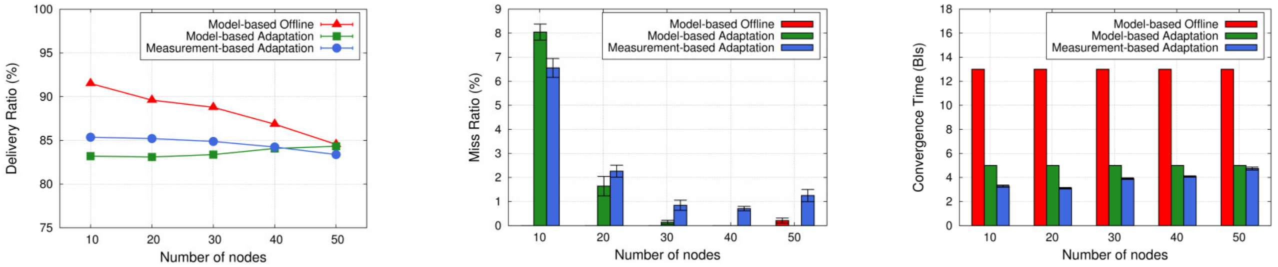
To compare the performance of the three algorithms described in the previous section we used the ns2 simulation tool [9], which includes the 802.15.4 module. To implement the analytical model leveraged by the offline algorithm we also used MATLAB [10]. In our analysis we considered a star network scenario where the sink node acts as the network coordinator, and sensor nodes are placed in a circle centered at the sink node, 10 m far from it. The transmission range was set to 15 m while the carrier sensing range was set to 30 m (according to the model in [11]). We considered a periodic reporting application where sensed data have to be reported to the sink periodically, i.e., at each Beacon Interval.

To evaluate the performance of the three considered algorithms we derived the following performance indexes.

- *Packet delivery ratio*, defined as the ratio between the number of packets correctly received by the sink, and the total number of packets generated by all sensor nodes. It measures the *long-term reliability.*
- *Miss ratio,* defined as the fraction of times the packet delivery probability – calculated over the current Beacon Interval – drops *below* the threshold required by the application. It measures the inability to provide *short-term reliability* (ideally should be zero).
- *Average energy per packet*, defined as the total energy consumed by each sensor node divided by the number of data packets correctly delivered to the sink. It measures the *energy efficiency.*
- *Average latency*, defined as the average time from when the packet transmission starts at the source node to when the packet is correctly received by the sink. It characterizes the *timeliness.*
- *Convergence time,* defined as the time when the packet delivery probability – calculated over the current Beacon Interval – reaches the threshold required by the application for the first time (from the network startup time). It measures the *ability to adapt* to changing conditions.

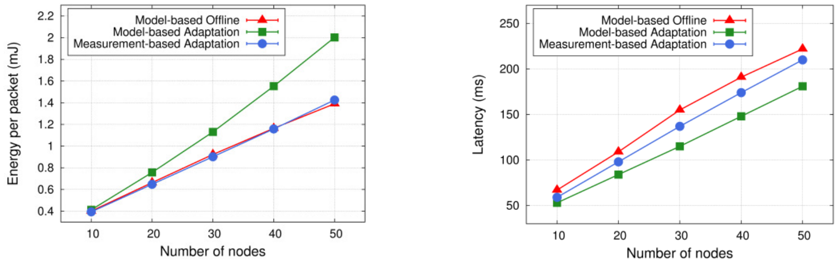
The energy consumed by a sensor node was calculated using the model presented in [12], which is based on the Chipcon CC2420 radio transceiver [13]. The model assumes four radio states, namely *transmit*, *receive*, *idle* and *sleep*. In addition, the model also accounts for the energy spent during the transition from a state to another.

In our analysis we assumed that the application is critical in terms of reliability and, hence, it requires a packet delivery ratio d^{des} >= 80%. In addition, to avoid concentrated packet losses, the application also requires a miss ratio lower than 10%. This means that the application can temporarily tolerate a delivery probability lower than 80%. However, this should occurs in a limited number of cases (less than 10% of the beacon intervals). Hence, the optimal CSMA/CA parameter setting must guarantee the requested reliability (in terms of delivery ratio and miss ratio), while minimizing the average energy per packet and average latency. We need to point out that the considered thresholds for delivery ratio and miss ratio (i.e., 80% and 10%) are somewhat arbitrary as, in practice, they depend on the specific application. However, we performed other experiments, not shown here, with different thresholds and we obtained results similar to those presented below.



(a) (b) (c)

Figure 1. Delivery ratio (a), Miss Ratio (b), and Convergence Time (c) for the different algorithms.



(a) (b)

Figure 2. Average Energy per Packet (a), and Average Latency (b) for the different algorithms

TABLE II. 802.15.4 MAC PROTOCOL PARAMETERS.

Parameter	Value
Bit Rate	250 Kbps
Data Frame (Payload) Size	109 (100) bytes
ACK Frame Size	11 bytes
Beacon Order (BO), Superframe Order (SO)	11, 8
macMinBE^{min}, *macMinBE*^{max}	1, 7
macMaxCSMABackoffs^{min}, *macMaxCSMABackoffs*^{max}	1, 10
macMaxFrameRetries^{min}, *macMaxFrameRetries*^{max}	0, 9
Power Consumption in RX mode (P_{rx})	35.46 mW
Power Consumption in TX mode (P_{tx})	31.32 mW
Power Consumption in Idle mode (P_{idle})	0.77 mW
Power Consumption in Sleep mode (P_{sleep})	0.036 µW

TABLE III. SPECIFIC PARAMETERS.

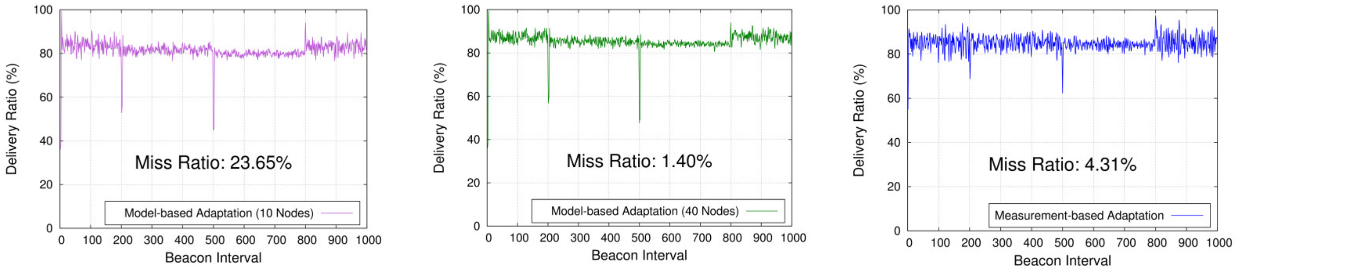
Parameter	Value
δ(ADAPT)	0.6
σ (ADAPT)	0.03
γ(ADAPT)	0.03
m (Model-based Adaptation)	4

Table II summarizes the 802.15.4 MAC protocol parameter values used for all the algorithms, while Table III refers to specific parameters of individual algorithms. For ADAPT, the considered values of δ, σ, and γ are taken from [7]. For the model-based adaptive algorithm, a window size *m*=16 was used in [6] to estimate α, β, and τ. We used a lower value (i.e., *m*=4), as it reduces the convergence time of the algorithm and it still provides accurate estimates (see below). As anticipated, we considered a periodic reporting application, which is typical for beacon-enabled sensor networks. We also assumed that each sensor node generates 10 data packets at every Beacon Interval. This means that all sensor nodes have a packet to transmit for some time. Hence, the operating conditions are very close to saturated traffic conditions. Accordingly, we set q_0=0 in the model used by both model-based algorithms.

In our experiments, for each scenario, we performed 10 independent replications, where each replication consists of 1000 Beacon Periods. The results are averaged over all the different replications. We also derived confidence intervals by using the independent replication method.

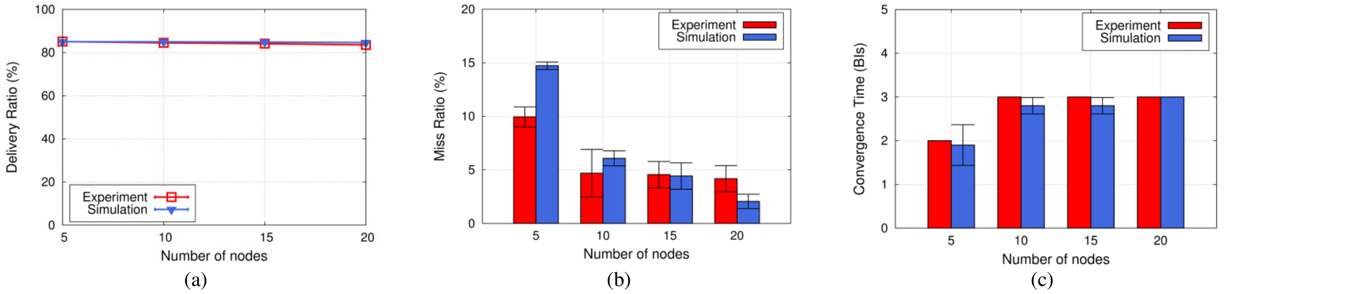
A. *Analysis in stationary conditions*

In this section we analyze the performance of the three algorithms in stationary conditions. Specifically, we assume that, for each experiment, the number of nodes is fixed. Figure 1-a and Figure 1-b show the delivery ratio and miss ratio of the considered algorithms, for an increasing number of sensor nodes. We can see that all the three algorithms are able to fulfill the application requirements. The two adaptive algorithms have similar performance in terms of delivery ratio, but the measurements-based algorithm typically exhibits a larger miss ratio. This is because it tends to oscillate between adjacent parameter sets, while the model-based adaptive algorithm is more stable. We can also observe that both the adaptive algorithms provide a delivery ratio slightly higher than the threshold. Instead, the delivery ratio provided by the offline algorithm is significantly above the threshold. This is because the adaptive algorithms can switch between different parameter sets over time, while the offline algorithm always uses the same set during the whole experiment. Also, we need to point out that, since the offline algorithm relies on the analytical model and the latter includes some simplifying assumptions, the parameter set derived by the algorithm may not be the *optimal* set. We observed that in some specific scenarios a different set would have guaranteed the reliability constraints with a lower energy consumption. In terms of energy efficiency (Figure 2-a), the offline algorithm and the



(a) (b) (c)

Figure 3. Delivery ratio, as a function of time, provided by the model-based adaptation algorithm with input equal to 10 nodes (a) and 40 nodes (b). Delivery ratio, as a function of time, provided by the measurement-based adaptation algorithm.



(a) (b) (c)

Figure 4. Delivery ratio, Miss ratio, and Convergence Time for the measurement-based adaptation algorithm: Experimental vs. Simulation Results

measurement-based algorithm exhibit similar performance, while the model-based adaptive algorithm is more energy consuming. This is mainly because the latter algorithm tends to increase the delivery probability by increasing the number of retransmissions (i.e. $macMaxFrameRetries$), while the former two algorithms achieve the same result by increasing the number of backoff trials (i.e. $macMaxCSMABackoffs$), which is less expensive in terms of energy consumption as highlighted in [7]. On the other side, the strategy used by the model-based adaptive algorithm results in a lower average latency experienced by packets, as highlighted by Figure 2-b. This is because, when a packet is re-transmitted the contention window size is re-initialized to its minimum value. Instead, when a new backoff trial is started, the contention window size is doubled (unless it has reached its maximum value). Finally, Figure 1-c shows the *convergence time* of the three algorithms. For the offline algorithm we assume as convergence time the time needed to solve the analytical model at the coordinator/sink node, derive the optimal parameter setting, and communicate it to the sensor nodes. Obviously, this time strongly depends on the machine used to solve the model. We found that, with a quad-core PC, this time is approximately 400 s (i.e., around 7 minutes). Assuming a Beacon Interval of about 30 s (31.45s with BO=11), this results in a convergence time of 13 Beacon Intervals, as shown in Figure 1-c (obviously it depends on the size of the Beacon Interval). The two adaptive algorithms have significantly lower convergence times. The model-based algorithms converges in about 5 Beacon Intervals as the optimal setting is derived using the samples measured in the previous *m* Beacon Intervals (and we used *m*=4). The measurement-based algorithm is the fastest one as, in the considered scenarios, it converges in 3-5 Beacon Intervals. Unlike the model-based algorithms, in ADAPT the convergence time is not constant and, generally, tends to increase with the number of sensor nodes. This is because it starts using the default parameter set (see Table I) and, then, increases the parameter values as shown in Algorithm 1, until an appropriate parameter configuration is reached. Obviously, the number of steps increases with the number of nodes as a higher number of contending nodes increases congestion and, thus, requires higher parameter values to guarantee the same reliability level.

A. Analysis in dynamic conditions

In this section we analyze the behavior of the three algorithms in dynamic conditions. To this end, we vary over time the number of active sensor nodes. We assume that, initially, there are 10 nodes. Then, this number increases to 20 at Beacon Interval 200, and to 40 at Beacon Interval 500. Finally, it reduces again to 10 at Beacon Interval 800. All the sensor nodes, when active, generate 10 packets per period and, hence, the offered load increases accordingly with the number of nodes. We restricted the following analysis to adaptive algorithms only, as the offline algorithm is not able to adapt itself to changing conditions. Also, we need to point out that the model-based adaptive algorithm requires the number of nodes as an input parameter. This may be problematic when the number of active nodes cannot be known in advance. In our analysis we ran two different set of experiments for the model-based algorithm, with two different input values for the number of nodes. In the first case the algorithm uses the initial number of nodes (i.e., 10), while in the second case it uses the maximum number of nodes (i.e., 40). Figure 3-a and Figure 3-b show the delivery ratio, as a function of the Beacon Interval, experienced in the two cases (we show here a single run for each case, however we observed similar results in all the runs we performed). As expected, in the first case the algorithm exhibits very bad performance when the number of sensor nodes is higher than 10. The miss ratio is 15.6% and 53.3% with 20 and 40 nodes, respectively. In the second case the algorithm performs well in all the considered conditions and the observed miss ratio is always less than 2%.

ADAPT does not require to know in advance the number of sensor nodes. The results for it are shown in Figure 3-c. Apart from spikes due to sudden changes in the number of nodes, the algorithm performs well in all the considered conditions. The miss ratio is, on average, equal to 4.3% and always less than 10%.

D. Learned Lesson

A number of consideration can be drawn from the previous analysis. First, the model-based offline algorithm is not able to adapt to changing operating conditions and, hence, it is not suitable for dynamic scenarios. Also, since the related analytical model makes some assumptions (e.g., on the packet generation process), it may not always provide the optimal setting in practical scenarios (e.g., periodic reporting applications). The latter consideration also applies to the model-based adaptive algorithm.

Both the adaptive algorithms perform similarly in terms of delivery ratio. The measurement-based algorithm tends to oscillate more, but it takes a lower time to converge. The model-based algorithm provides a lower average latency, while the measurement-based algorithm is more energy efficient. This is an important point as sensor nodes are typically energy constrained. Finally, it should be emphasized that the model-based adaptive algorithm requires to know in advance the number of sensor nodes in the network and assumes that the wireless channel is ideal. This makes it unsuitable for practical scenarios where operating conditions (including the number of active nodes) may vary over time and transmission errors cannot be neglected. The measurement-based algorithm does not suffer from these limitations.

IV. EXPERIMENTAL ANALYSIS

Motivated by the previous remarks, we implemented the measurement-based adaptive algorithm (ADAPT) in real sensor nodes and performed an experimental analysis aimed at (i) showing that ADAPT is a viable solution for a real environment, and (ii) validating the previous simulation results for it. Our testbed consists of Tmote Sky sensor nodes [14] with TinyOS 2.x operating system [15]. Tmote Sky sensor nodes use the Chipcon CC2420 radio transceiver [12] that is compliant to the 802.15.4 physical layer. In our experimental analysis, we referred to the same star network scenario and parameters setting considered in the simulation analysis. We measured all the performance indexes considered in the simulation analysis but energy consumption, as it is extremely difficult to accurately measure very small energy consumptions. The results obtained are shown in Figure 4. It can be observed that delivery ratio, miss ratio and convergence time obtained in simulations and experiments are quite similar. We also measured packet latency (not shown here for the sake of space). As expected, we observed an experimental latency slightly higher than that measured in simulations. This is due to additional re-transmissions triggered by collisions caused by non-perfectly synchronized clocks and possible transmission errors.

V. CONCLUSIONS

In this paper we have compared three different algorithms for deriving the optimal settings of 802.15.4 sensor networks so as to guarantee the reliability requirements of the application with minimum energy consumption. The considered algorithms take different approaches, namely *offline computation*, *model-based adaptation* and *measurement-based adaptation*. We have found that both the adaptive algorithms perform well, however the model-based approach has some limitations that make it unsuitable for practical scenarios, where operating conditions typically vary over time and transmission errors cannot be neglected. Instead, the heuristic measurement-based algorithm does not suffer from such limitations. Hence, we have implemented it in a real sensor network and validated the simulation results for it through experimental measurements.

REFERENCES

[1] IEEE Standard for Information technology, Part 15.4; Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEE Computer Society, 2006.

[2] ZigBee Alliance, The ZigBee Specification version 1.0 (Q4/2007).

[3] R. Zurawski, "Networked Embedded Systems: An Overview" Chapter 1 in Networked Embedded Systems (R. Zurawski, Editor), pp. 1.11-1.16, CRC Press, 2009

[4] G. Anastasi, M. Conti, M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in 802.15.4 Wireless Sensor Networks", *IEEE Transactions on Industrial Informatics*, Vol.7, N.1, Feb. 2011.

[5] P. Park, P. Di Marco, P. Soldati, C. Fischione, K. H. Johansson, "A Generalized Markov Chain Model for Effective Analysis of Slotted IEEE 802.15.4", *IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS* 2009), 2009.

[6] P. Park, P. Di Marco, C. Fischione, K. Johansson, "Modeling and Optimization of the IEEE 802.15.4 Protocol for Reliable and Timely Communications", *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[7] M. Di Francesco, G. Anastasi, M. Conti, S. Das, V. Neri, "Reliability and Energy Efficiency in IEEE 802.15.4/ZigBee Sensor Networks: A Cross-layer Approach", *IEEE Journal on Selected Areas in Communications,* Vol. 29, N. 8, September 2011.

[8] S. Brienza, D. De Guglielmo, G. Anastasi, M. Conti, V. Neri, "Strategies for Optimal MAC Parameter Setting in IEEE 802.15.4 Wireless Sensor Networks: a Performance Comparison (Extended Version)", Technical Report DII-2012-13, Univ. Pisa, Italy, 2013. http://www.iet.unipi.it/~anastasi/papers/tr-13-4.pdf.

[9] Network Simulator Ns2, http://www.isu.edu/nsnam/ns.

[10] MATLAB software, *http://www.mathworks.it/products/matlab/*.

[11] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella, "Understanding the Real Behavior of 802.11 and Mote Ad hoc Networks," *Pervasive and Mobile Computing*, Vol. 1, N. 2, 2005.

[12] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, W. Dehaene, "Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives", Proc. Conference on Design, Automation and Test in Europe (DATE), Volume 1, pp. 196-201, March 7-11, 2005.

[13] Chipcon CC2420 Website: http://www.ti.com/product/cc2420.

[14] Tmote Sky Platform, MoteIV Corporation. [Online]. Available: http://www.moteiv.com/products/tmotesky.php.

[15] TinyOS Website [Online]. Available: http://www.tinyos.net/.