Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 6351623, 18 pages
https://doi.org/10.1155/2018/6351623

Research Article

A Selective-Awakening MAC Protocol for Energy-Efficient Data Forwarding in Linear Sensor Networks

Iclia Villordo-Jimenez[1,2] **Noé Torres-Cruz**[1,2]
Marcelo M. Carvalho[3] **Rolando Menchaca-Mendez**[1]
Mario E. Rivero-Angeles[1] **and Ricardo Menchaca-Mendez**[1]

[1]*Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, 07738 Ciudad de México, Mexico*
[2]*UPIITA, Instituto Politécnico Nacional, Av. IPN No. 2580, Col. Barrio la Laguna Ticomán, 07340 Ciudad de México, Mexico*
[3]*Departamento de Engenharia Elétrica, Universidade de Brasília, Campus Darcy Ribeiro, 70919-970 Brasília, DF, Brazil*

Correspondence should be addressed to Iclia Villordo-Jimenez; ivillordo@ipn.mx

Received 22 December 2017; Revised 7 March 2018; Accepted 21 March 2018; Published 10 May 2018

Academic Editor: Juan C. Cano

We introduce the *Selective-Awakening MAC* (SA-MAC) protocol which is a synchronized duty-cycled protocol with pipelined scheduling for Linear Sensor Networks (LSNs). In the proposed protocol, nodes selectively awake depending on node density and traffic load conditions and on the state of the buffers of the receiving nodes. In order to characterize the performance of the proposed protocol, we present a Discrete-Time Markov Chain-based analysis that is validated through extensive discrete-event simulations. Our results show that SA-MAC significantly outperforms previous proposals in terms of energy consumption, throughput, and packet loss probability. This is particularly true under high node density and high traffic load conditions, which are expected to be common scenarios in the context of IoT applications. We also present an analysis by grade (i.e., the number of hops to the sink, which is located at one end of the LSN) that reveals that LSNs exhibit heterogeneous performance depending on the nodes' grade. Such results can be used as a design guideline for future LSN implementations.

1. Introduction

Internet of Things (IoT) is a network paradigm that is playing a key role in the development of smart environments where everyday objects are augmented with computational capabilities and Internet connectivity [1, 2], so that they can interact with each other in order to share information about the environment, coordinate actions autonomously, and fulfill other specific purposes such as target tracking, medical supervision of patients, and animal monitoring [3–5].

IoT has been proposed as a core technology for establishing what it is known as Advanced Measurement Infrastructures (AMI) which are responsible for monitoring public distribution networks of water, electricity, or gas [6–8]. With this technology, it is possible to identify leaks along lines or pipelines and optimize the operation of these distribution networks.

IoT applications have also been identified as a mean for fostering smart mobility. In these scenarios, a set of sensors are located along the road infrastructure in order to coordinate urban transport, monitor events that affect mobility, or manage transit through dynamic coordination of traffic lights [9, 10]. The objective is to extend the lifetime of the road infrastructure, improve safety, decrease commuters' travel time, and diminish air pollution in densely populated cities.

As in the previous examples, when the sensing devices are deployed along linear structures, the resulting wireless sensor network (WSN) is usually denoted as a *Linear Sensor Network* (LSN) [11]. In particular, when the node density is high, as it is commonly the case when a wide variety of variables are being

sensed [3], the LSN is named a *thick* LSN [12]. As in any WSN, one of the main goals when designing an LSN is to provide high throughput in an energy-efficient way [13], which could be achieved by defining appropriate Medium Access Control (MAC) protocols.

Among the large body of work devoted to designing efficient MAC protocols that improve energy consumption, synchronized duty-cycling protocols [14] have been shown to be particularly effective in the context of MANETs [15], WSNs [16–18], and LSNs [3, 19, 20]. Building on the success of duty-cycling, complementary techniques have been proposed to further improve the performance of LSN; for instance, Tong et al. introduced the PRI-MAC protocol [19] which implements a *pipelined* scheduling as a complement for a duty-cycled protocol that is based on S-MAC [21].

In pipelined scheduling, nodes are classified according to their grade (distance in hops to the sink node) and nodes with the same grade share the same schedule; namely, they synchronously enter the different modes in the following sequence: one reception slot, one transmission slot, and ξ sleeping slots. Grade schedules are arranged in such a way that packets can be forwarded from one grade to another in a pipelined fashion. This way, a packet can traverse a source-to-sink path in a single cycle if there are no other packets at the queues of the nodes along its path, which is the main advantage of this strategy over other proposals.

In this paper, we present a synchronized duty-cycled protocol with pipelined scheduling where nodes selectively awake depending on node density and traffic load conditions. Unlike previous works, the proposed Selective-Awakening MAC (SA-MAC) protocol is specifically designed for IoT scenarios, where node density and traffic load can be high. To cope with these conditions, SA-MAC introduces the following two rules:

(i) During a transmission slot, nodes that are candidates to contend for the channel awake with probability $p(i)$, where i is the nodes' grade. This probability is selected in order to maximize the LSN throughput.

(ii) During a reception slot, nodes with saturated buffers do not awake.

As shown in Section 7, the first rule simultaneously reduces node energy consumption and the probability of having a packet collision, without reducing throughput or requiring extra signaling. The second rule also reduces energy consumption without affecting any of the network performance metrics, because it avoids wasting energy to receive a packet that will be dropped. To the best of our knowledge, these strategies, as well as their analysis and evaluation, have not been previously reported in the context of LSN or IoT applications.

In concordance with previous work (e.g., [16, 19, 21]), our proposed protocol is specifically designed to operate in an LSN with a single sink node at one end of the network, and the remaining nodes can both generate new packets and act as relays for other nodes without performing any data processing or aggregation. In addition, data packets are transferred between nodes located in adjacent grades by means of unicast transmissions. A detailed description of the assumptions and operation scenarios is presented in Section 3.

In order to evaluate the performance of SA-MAC, we developed a mathematical model of the nodes' queue length based on a Discrete-Time Markov Chain (DTMC). From the steady-state probabilities of this DTMC, we developed expressions for the main system performance metrics, namely, energy consumption, throughput, packet loss probability, and source-to-sink delay. This analysis is similar to those presented in [16–19] and is validated through extensive discrete-event simulations.

Our numerical results show that SA-MAC clearly outperforms previous work in terms of energy consumption, packet loss probability, and throughput, at the cost of a moderate increase in source-to-sink delay. This is particularly true under high node density and high traffic load conditions, which are expected to be a possible scenario in the context of IoT applications [1].

Lastly, it is important to mention that our analysis is intended to evaluate not only the general performance of the network, but also the performance at each grade. This detailed analysis reveals that the network performance is not homogeneous and makes it possible to identify those specific grades that may compromise the whole network lifetime and reliability. This is another major contribution of this paper because it provides a guideline for the design of pipelined LSN.

The rest of the paper is organized as follows. In Section 2 we discuss a representative sample of previous work in duty-cycled MAC protocols for multihop WSNs and LSNs. In Section 3 we present a general overview of the proposed protocol. Then, in Section 4 we derive the DTMC that models our system. In Section 5 we present expressions for the network performance metrics. In Section 6 we provide criteria to select the awakening probability. Lastly, we discuss relevant numerical results and present our conclusions in Sections 7 and 8, respectively.

2. Related Work

Synchronized duty-cycled protocols for WSNs have been extensively studied because of their energy efficiency, which is achieved by reducing idle listening and overhearing. In these schemes, nodes synchronously wake up in order to transmit/receive one or several packets during a cycle. One example of this kind of protocol is Sensor MAC (S-MAC), which was proposed by Ye et al. [21]. In S-MAC, a fully connected group of nodes share the same schedule; namely, all the nodes synchronously enter the transmission/reception mode to exchange information and, then, all of them synchronously enter sleeping mode to save energy. One disadvantage of S-MAC is that it substantially increases source-to-sink delay because nodes may wait a complete cycle in order to attempt a transmission; moreover, this disadvantage is accentuated when multiple hops are required because the delay accumulates at each hop.

Yang and Heinzelman [16] provide a model, based on a DTMC, to evaluate the performance of duty-cycled protocols.

In their model, the Markov chain states represent the nodes' queue length; and the chain solution is used to evaluate throughput, delay, and power consumption. This kind of analysis has been utilized in subsequent works (e.g., [17, 18]); however, the analysis in [16] is restricted to one-hop WSNs, whereas more recent works, including [19] and our work, have extended the analysis to study LSNs.

On the other hand, in recent years, multiple works have proposed synchronized duty-cycled protocols for multihop WSNs, including [17, 18, 22–24]. In the following paragraphs, we present a brief overview of each of them.

Guntupalli and Li [17] propose another synchronized duty-cycled protocol for multiple packet transmission per cycle. In this scheme, which is called DW-MAC, multiple packet transmission is enabled by means of competition-based reservations established during a particular time interval, called the data period. The authors present a DTMC-based analysis that characterizes the energy consumption and the lifetime of the network. It must be noticed that although their study includes multiple hops, it is limited to a scenario where there is only one relay between the source node and the sink node. In [18], Guntupalli et al. extended the analysis presented in [17] to incorporate a four-dimensional DTMC; this analysis, however, does not consider multiple hops.

Singh and Chouhan [22] present a protocol called CROP-MAC that optimizes pipelined flows with the help of staggered sleep/wake scheduling, efficient synchronization, and appropriate use of routing layer information. Based on CROP-MAC, Singh et al. [23] developed a protocol that provides multipacket transmission per cycle (JRAM protocol). In this protocol, a node has more than one opportunity to transmit, and the sink is able to receive packets from multiple nodes in the same cycle. JRAM accomplishes this by dividing the network into disjoint sets of nodes, by proposing a new cycle structure, and by reducing the size of periodical synchronization packets. Their results indicate that JRAM works best in scenarios with high-rate events occurrence.

Additionally, Khaliq and Henna [24] present a protocol called HE-PRMAC that uses cross-layer information for the transmission of packets through multiple hops in a single duty cycle. As a result, packet latency is reduced. In order to decrease the overhead when traffic load is high, HE-PRMAC introduces a new message called Ready To Receive (RTR), which is used to transmit how many packets a destination node can receive from a source node.

Even though many of the above-mentioned works analyze multihop WSNs, their protocols were not designed to take advantages of particular topologies. By contrast, in [3, 19, 20] the authors proposed synchronized duty-cycled MAC protocols specifically for LSNs.

Wang et al. [3] introduce a Utility-based Adaptive Duty-Cycle (UADC) algorithm to increase energy efficiency, reduce transmission delay, and improve network lifetime of WSNs with linear topology. The main idea of this algorithm is to select the relay node that maximizes the utility of the system, which is defined as a function of the nodes' energy consumption.

Tong et al. [19] propose PRI-MAC, a synchronized duty-cycled protocol that incorporates a pipelined scheduling for LSN. The authors evaluated the performance of the network in terms of throughput, node's radio activity per cycle, and delivered packet latency. Khanh et al. [20] introduce the RP-MAC protocol, which is based on PRI-MAC. In RP-MAC the authors define a new cycle structure and a new state, called overhearing, which eliminates the need of transmitting the Request To Send (RTS) message, hence reducing the end-to-end delay and energy consumption, in comparison with PRI-MAC.

Similar to [19, 20], we also analyze a pipelined scheduling for LSN; however, we introduce a novel MAC protocol, where nodes selectively awake, depending on node density and traffic load in the network.

From this review, we would like to highlight the main contributions of our work: (i) none of previous proposals are designed for dense LSNs, which can be widely applicable in the context of IoT; (ii) unlike previous protocols, our proposed protocol does not aggregate new control messages or node states during a cycle; (iii) even though the performance experienced by a node in an LSN is highly dependent on its distance to the sink, previous grade analyses are limited and fail to provide detailed performance models. In contrast, our model allows the analysis of both the complete LSN and each grade in the system, as it is described in the next sections.

3. Selective-Awakening MAC Protocol

In this section, we provide a general overview of the proposed protocol. We first present a brief description of one of its main components: the pipeline scheduling (Section 3.1). Then, in Section 3.2, we provide the basis of traditional synchronized duty-cycled MAC protocols ([16, 19, 21]). Later, in Section 3.3, we describe the main features of our proposed MAC protocol as well as its advantages over previous work.

3.1. Pipelined Scheduling. As in [19], we assume that the LSN has a unique sink node which is located at one end of the network. The remaining nodes generate packets with information provided by their respective sensors and can act as relays. Nodes are classified according to their *grade*, namely, their distance in hops to the sink node. Figure 1 illustrates this type of network. The grade of a node is denoted by i, and the total number of grades in the LSN is denoted by I. By definition, the sink node has grade zero; hence $i \in [0, I]$. As in [19, 25], we also assume that the process of determining the node's grade and the next node on the way to the sink was previously performed during an initialization stage, where a different layer of the protocol stack fills a *Next Hop* table; in other words, we are assuming unicast transmissions.

Multihop transmissions are accomplished through a pipelined scheduling where newly generated packets are relayed from a node at grade i to a node at grade $i - 1$ until they reach the sink. During the pipelined scheduling, nodes operate according to the frame depicted in Figure 2. In this frame, the nodes at grade $i + 1$ with packets to transmit synchronously awake to contend for the channel, following the MAC protocol described in Section 3.2. During this same time slot, nodes at grade i also synchronously awake to be able to receive a packet from grade $i + 1$. In the next slot,

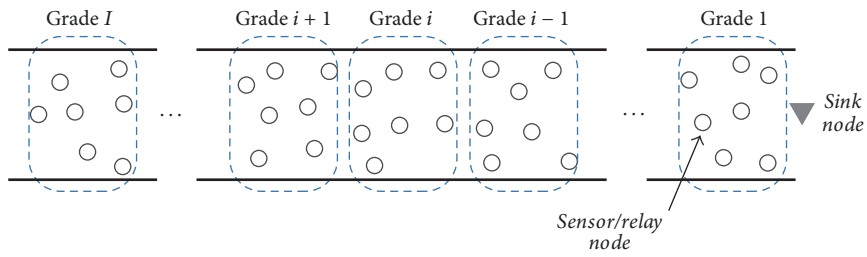

Figure 1: LSN with nodes classified by their grades.

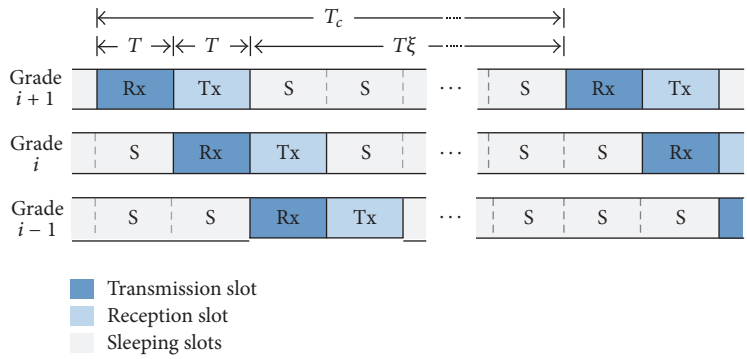

Figure 2: Pipelined scheduling cycle.

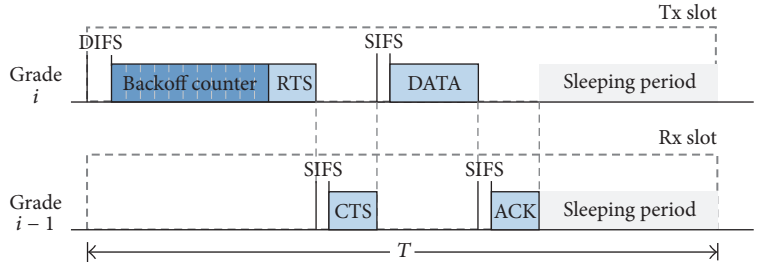

Figure 3: Transmission/reception example.

nodes at grade i act as transmitters, while nodes at grade $i-1$ act as receivers. This process continues until nodes at grade 1 become transmitters and the sink node becomes the receiver.

After the reception and transmission slots, all the nodes at a particular grade go to sleep for ξ slots (see Figure 2), in order to save energy. According to this description, the cycle duration, denoted by T_c, is equal to $(\xi+2)T$, where T is the duration of every slot. Also note that since the schedules between adjacent grades are staggered, it is possible for a packet to travel through the whole LSN in only one cycle.

3.2. Synchronized Duty-Cycled MAC Protocol. Similar to PRI-MAC [19], we assume that, at the beginning of every transmission slot, awake nodes at grade i with packets to transmit synchronously initiate a backoff counter (see Figure 3), whose value w is a uniform random variable in $[0, W-1]$. This backoff counter generates a contention window whose duration is equal to σw, where σ is the length of one minislot.

Contending nodes keep listening to the channel during their backoff time and if they detect a transmission from other nodes, they go to the sleeping mode. Otherwise, they initiate an RTS/CTS/DATA/ACK handshake with a receiver at grade $i-1$ (recall that we assume unicast transmissions). Figure 3 illustrates the handshake process for the case where a data packet is successfully transmitted.

When there is a tie in the smallest backoff counter generated by the contending nodes, an RTS collision occurs, and none of the winners receive the corresponding CTS message. When these nodes detect the lack of this message, they go to sleep; consequently, no data packet is transmitted from grade i during that particular cycle. From now on, we will refer to the smallest backoff time generated by a contending node as the *winner* backoff time.

We also assume that time is slotted with a slot duration T, that is given by

$$T = \tau_{\text{difs}} + \sigma W + \tau_{\text{rts}} + \tau_{\text{cts}} + \tau_{\text{data}} + \tau_{\text{ack}} + 3\tau_{\text{sifs}}, \quad (1)$$

where τ_{difs} is the duration of a DCF Interframe Space (DIFS), σW is the maximum allowed backoff duration, τ_{sifs} is the duration of a Short Interframe Space (SIFS), and τ_{rts}, τ_{cts},

τ_{data}, and τ_{ack} are the duration of the messages defined by the protocol.

Note that it is possible to have a sleeping period after the exchange of frames within a transmission slot because the winner backoff time can be smaller than W (see Figure 3). However, the slot duration is always equal to T in order to maintain synchronization. Also, note that, in this protocol, at most one packet can be transmitted per cycle from one grade to the following one. Thus, since this is also true for the communication between grade 1 and the sink node, the maximum throughput in the LSN is $1/T_c$ packets/sec.

3.3. Selective Awakening. Unlike previous proposals (e.g., PRI-MAC) where all the nodes at grade i, with packets to transmit, awake to contend for the channel during their transmission slot, in the proposed protocol, nodes at grade i wake up with probability $p(i)$. The rationale behind this proposal is that if there are many awake nodes at each grade, a lot of collisions may occur, degrading the throughput performance. Moreover, even if the collision probability is kept small (e.g., by utilizing a large contention window), a lot of energy is unnecessarily wasted, since, among the awake nodes, at most one of them will successfully transmit a packet. It is important to point out that the values of the probabilities $p(i)$ have to be carefully selected because if they are excessively small, it will be common to have situations where none of the nodes wake up at the transmission slots, reducing the network throughput. Considering this, we propose to set the values of the probabilities $p(i)$ in such a way as to maximize the network throughput as a function of the traffic load.

In Section 6, we describe the criteria to select these probabilities. The main intuition behind our proposal is that, due to the nature of an LSN, the traffic load at lower grades is higher because nodes at those grades receive the accumulated traffic generated at higher grades. Our numerical results show that, by following these criteria, the energy consumption is significantly reduced in comparison with PRI-MAC, without degrading network throughput.

We also propose to consider the state of the data queues when deciding whether to awake a node during a reception slot. More specifically, in SA-MAC nodes with full data queues remain in sleeping state during reception slots. This simple functionality substantially increases energy efficiency without affecting the system throughput or any other performance metric because it avoids wasting energy to receive a packet that will be dropped.

In the following sections, we develop a mathematical analysis of the performance of the proposed protocol based on a Discrete-Time Markov Chain (DTMC). In this analysis, we assume that the only reason for transmission errors is packet collisions, similar to previous works, such as [16, 19]. Therefore, the channel is assumed to be ideal, without effects such as multipath fading, shadowing, and signal capture. In order to make the following sections more readable, we summarize the main variables used throughout our analyses in the Notations section.

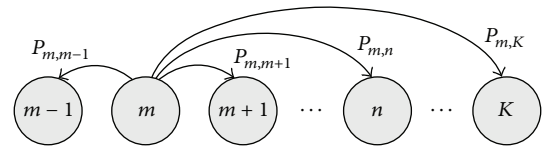

Figure 4: DTMC that represents the queue's length at the beginning of the transmission slot.

4. Analytical Model

As it has been proposed in several works (e.g., [16–19]), we model the behavior of the nodes by means of a unidimensional DTMC whose states represent the length of a node's queue at the beginning of the transmission slot. Let K be the maximum queues' length (in packets) and m any of the possible states for this chain; hence, $m \in [0, K]$. In Figure 4, we illustrate the possible transitions from one of these states.

In addition, we denote by $P^i_{m,n}$ the transition probabilities from state m to state n in one cycle, for a node in grade i. As we show in Figure 4, the chain cannot transit from state m to a state $n \leq m - 2$ because the nodes can transmit (eliminate from its queue) at most one packet during one cycle. By contrast, transitions to any higher state are possible as a result of the new-packet generation process in each node.

As it was mentioned before, a node at grade i has to relay packets from nodes belonging to higher grades (besides transmitting the locally generated packets); hence, the DTMC steady-state probabilities at grade i depend on the behavior of the higher grades.

In order to obtain these steady-state probabilities at each grade, we adapted the method described in [19] to our proposal. This method consists in solving the DTMC for the grade i and then using these results to calculate the probability of having a successful transmission at this grade, which in turn is used to solve the DTMC at grade $i-1$, and so on. Notice that the steady-state probabilities at grade I do not depend on other grades (since nodes at this grade do not relay packets), and hence, the method proceeds by first solving the Markov chain for this latter case.

The presentation of the analysis is organized as follows. In Section 4.1 we calculate the probabilities of transmitting or receiving a packet in each cycle. These probabilities are required to establish the transition probabilities for the DTMC. Since the DTMC sampling points are the beginnings of the transmission slots, all the probabilities in Section 4.1 refer to such points. In Section 4.2, we establish the transition probabilities for the DTMC and we provide a detailed description of the method mentioned in the previous paragraph. Considering that the evaluation of some performance parameters depends on the probability of having a saturated queue at the beginning of the reception slot, we obtain such probability in Section 4.3.

4.1. Probabilities of Transmitting and Receiving a Packet. For our analysis, let $p(i)$ be the awakening probability of nodes at grade i given nonempty queues, and let $p_e(i)$ be the probability of having an empty queue at grade i; therefore,

by using the Law of Total Probability, the (unconditional) awakening probability $p_a(i)$ for each node is defined as

$$p_a(i) = p(i)\left(1 - p_e(i)\right) + p^*(i)\, p_e(i), \quad (2)$$

where $p^*(i)$ is the awakening probability given that the queue is empty. Since nodes with empty queue are not allowed to wake up, $p^*(i) = 0$; hence, (2) becomes

$$p_a(i) = p(i)\left(1 - p_e(i)\right). \quad (3)$$

This means that nodes at grade i will wake up only if they have packets to transmit.

For the sake of simplicity, we further assume that the LSN is homogeneous and, hence, that the number of nodes at each grade, denoted by N, is constant. Under these conditions, and since each node wakes up independently of the other nodes in the same grade, then, the probability of having n contenders besides the node of interest at grade i is given by

$$C_n(i) = \binom{N-1}{n} p_a(i)^n \left(1 - p_a(i)\right)^{N-1-n}. \quad (4)$$

In the proposed protocol, nodes contend for the channel by generating a backoff counter w, which is a uniform random variable in $[0, W-1]$. Given a value of the backoff counter w, the probability of winning the contention is the same as the probability of having the smallest backoff counter among the contending nodes. Since the probability of having a backoff timer smaller than another node is $(W-w)/W$, the probability of having the smallest backoff counter in a group of n contenders equals $((W-w)/W)^n$. This probability includes the case when more than one contender obtains the smallest backoff timer.

Now, by using the Law of Total Probability, we obtain that the probability of winning the contention given n additional contenders, t_n, can be written as

$$t_n = \frac{1}{W}\sum_{w=0}^{W-1}\left(\frac{W-w}{W}\right)^n. \quad (5)$$

Analogously, the probability of having a successful packet transmission (i.e., winning the contention without collisions) given n additional contenders, s_n, can be expressed as

$$s_n = \frac{1}{W}\sum_{w=0}^{W-1}\left(\frac{W-w-1}{W}\right)^n. \quad (6)$$

Recalling that $C_n(i)$ is the probability of having n contenders besides the node of interest, we can use (4) and (5) to obtain the probability $p_t(i)$ of winning the contention as

$$p_t(i) = \sum_{n=0}^{N-1} C_n(i)\, t_n. \quad (7)$$

Similarly, by combining (4) and (6), we obtain the probability $p_s(i)$ of having a successful packet transmission as

$$p_s(i) = \sum_{n=0}^{N-1} C_n(i)\, s_n. \quad (8)$$

Notice that we calculate $p_s(i)$ from the point of view of a contending node at grade i; hence, for an external observer (e.g., a receiver in the next grade), the probability of having a successful packet transmission from a node at grade i is $p_a(i)p_s(i)$.

Now, since we are assuming that the number of nodes in each grade is constant and that nodes communicate in unicast mode, the number of receivers at grade i equals the number of transmitters at grade $i+1$. Consequently, the packet reception probability $p_r(i)$ for a node at grade i is the same as the probability of having a successful packet transmission from a node at grade $i+1$. Equation (9) computes the packet reception probability $p_r(i)$, for $0 \le i \le I-1$; and $p_r(I) = 0$ (since grade I has no packets to relay).

$$p_r(i) = p_a(i+1)\, p_s(i+1). \quad (9)$$

On the other hand, since contending nodes can detect a busy channel or win the contention, we can establish that $p_t(i) + p_b(i) = 1$, and the probability of detecting a busy channel (for an awake node of interest) can be written as

$$p_b(i) = 1 - p_t(i). \quad (10)$$

In the same way, once a node wins the contention, two possible events may occur: a successful transmission or a collision; therefore $p_t(i) = p_s(i) + p_c(i)$. According to this, the probability of collision (for an awake node of interest) can be defined as

$$p_c(i) = p_t(i) - p_s(i). \quad (11)$$

4.2. Markov Chain Model. The proposed DTMC has $K+1$ states that represent a node's queue length, where K is the queue size. The sampling points of this Discrete-Time Markov Chain are placed at the beginning of the transmission slots, which also implies that the chain's time step is equal to the cycle duration (T_c).

We further assume that a node generates packets related to the monitored phenomena, at rate λ packets/sec, according to a Poisson process; that is, packet interarrival times follow an exponential distribution. By considering this, the probability of generating k packets during a cycle can be expressed as

$$a_k = e^{-\lambda T_c}\frac{(\lambda T_c)^k}{k!}. \quad (12)$$

Because the chain transition probabilities depend on $p(i)$, $p_t(i)$, and $p_r(i)$, as well as their complements, here, we define $q(i) = 1 - p(i)$, $q_t(i) = 1 - p_t(i)$, and $q_r(i) = 1 - p_r(i)$ in order to simplify some of the subsequent expressions.

Now, let $P^i_{m,n}$ be the probability of having a transition from state m to state n, for a node at grade i; then, according to all the previous considerations, $P^i_{m,n}$ can be expressed as follows (for the sake of space and clarity, in these equations the dependence on i is omitted):

$$P^i_{0,0} = a_0 q_r, \quad (13)$$

$$P^i_{0,n} = a_{n-1}p_r + a_n q_r; \quad n \in [1, K-1], \quad (14)$$

$$P^i_{m,n} = 0; \quad m \in [2, K],\ n \in [0, m-2], \quad (15)$$

$$P^i_{m,m-1} = p a_0 q_r p_t; \quad m \in [1, K], \quad (16)$$

$$P^i_{m,m} = p\left[a_0 p_r p_t + a_0 q_r q_t + a_1 q_r p_t\right] + q a_0 q_r; \quad m \in [1, K-1], \tag{17}$$

$$P^i_{m,n} = p\left[a_{n-m-1} p_r q_t + a_{n-m}\left(p_r p_t + q_r q_t\right) + a_{n-m+1} q_r p_t\right] + q\left[a_{n-m-1} p_r + a_{n-m} q_r\right]; \quad m \in [1, K-1],\ n \in [m+1, K-1], \tag{18}$$

$$P^i_{m,K} = 1 - \sum_{n=0}^{K-1} P_{m,n}; \quad m \in [0, K]. \tag{19}$$

It is important to point out that the chain transition probabilities depend on the probability $p_t(i)$ of winning the contention (with or without collision). This is because, as in previous works, we assume that the packet at the head of the queue is removed when the node wins the contention, even if a collision occurs. This strategy implies that every packet has only one chance to be transmitted from one grade to the next one.

In order to provide a deeper insight into the rationale behind (13)–(19), we additionally highlight the following:

(i) Because a node with an empty queue has 0 probability of awaking or transmitting, $P^i_{0,n}$ does not depend on $p(i)$ or $p_t(i)$, as can be seen in (13) and (14).

(ii) Since a node can transmit one packet, at most, during a cycle, its queue length cannot be reduced by two or more units during this period of time; therefore, $P^i_{m,n}$ for $n \leq m-2$ is zero, as can be seen in (15).

(iii) In general, $P^i_{m,n}$ is expressed as the sum of two main terms: (a) one term that represents the transitions when the node is awake during the transmission slot, and, hence, it depends on both $p_t(i)$ and $p_r(i)$; (b) one term that represents the transitions when the node is asleep during such slot; therefore, it only depends on $p_r(i)$. This was made explicit in (17) and (18).

Now, in order to evaluate the performance of the whole LSN, the previous DTMC has to be solved for each grade to find the steady-state probabilities, π^i_m for $0 \leq m \leq K$, which represent a unique solution, because the DTMC is irreducible and aperiodic. Given that the solution at grade i depends on the solutions of higher grades, firstly, we proceed by solving the DTMC for grade I because the probability of receiving a packet from a higher grade equals 0 ($p_r(I) = 0$).

In general, π^i_m can be found by substituting $p(i)$, $p_r(i)$, $p_t(i)$, and a_k in (13)–(19) and then numerically solving the resulting system of linear equations. Except for $p_t(i)$, all the probabilities that are involved in (13)–(19) are known at this point: $p(i)$ can be previously selected according to the criteria of Section 6, $p_r(i)$ is equal to zero (for $i = I$) or can be calculated according to (9) (for $i < I$), and a_k is defined in (12).

Unfortunately, $p_t(i)$ is expressed in terms of the probability $p_e(i)$ of having an empty queue, which in turn is part of the solution of the DTMC because $p_e(i)$ is equal to π^i_0. In order to overcome this problem, we propose the following numerical solution at each grade i:

(1) Propose an initial value for $p_e(i)$ and use it to calculate $p_t(i)$, according to (3)–(5) and (7).

(2) Substitute $p_t(i)$, as well as $p(i)$, $p_r(i)$, a_k, and so on, in (13)–(19), and solve the DTMC, for example, through the Gauss-Seidel method [26].

(3) Make $p_e(i) \leftarrow \pi^i_0$.

(4) Repeat Steps (2) and (3) until $p_e(i)$ has converged according to certain criteria (e.g., mean squared error within a certain threshold value).

(5) Compute $p_r(i-1)$ according to (9), because this value will be required when computing the solution for the next grade.

In the Appendix, we present a convergence analysis of this iterative method.

4.3. Probability of Having a Full Queue at the Beginning of the Reception Slot. Let ρ^i_K be the probability of having a full queue at the beginning of the reception slot. We can compute ρ^i_K using the solution of the previous DTMC, instead of establishing a new chain. Now, let T^*_c be the period of time between the beginning of a transmission slot and the beginning of the next reception slot; we have that $T^*_c = (\xi + 1)T$, as can be seen in Figure 2. Then, the probability of generating l or more packets during T^*_c seconds can be computed as

$$A_l = 1 - \sum_{k=0}^{l-1} e^{-\lambda T^*_c} \frac{\left(\lambda T^*_c\right)^k}{k!}. \tag{20}$$

On the other hand, recall that π^i_m, for $m \in [0, K]$, represents the probability that a node has m packets at the beginning of the transmission slot. A node in this state accumulates K packets by the beginning of the following reception slot, if one of the following events occurs:

(i) The node removes one packet from its queue and generates $K - m + 1$ or more new packets.

(ii) The node generates $K - m$ or more new packets, without removing packets from its queue.

The first of these events occurs with probability $p(i)p_t(i)A_{K-m+1}$ for $m > 0$, whereas the second one occurs with probability $(p(i)q_t(i) + q(i))A_{K-m}$ for $m > 0$ and with probability A_K for $m = 0$. Due to the fact that these statements are true for $m \in [0, K]$, we can obtain ρ^i_K as

$$\rho^i_K = \pi^i_0 A_K + \sum_{m=1}^{K} \pi^i_m \left[p(i)\, p_t(i)\, A_{K-m+1} + \left(p(i)\, q_t(i) + q(i)\right) A_{K-m}\right]. \tag{21}$$

Notice that ρ^i_K can also be interpreted as relayed-packet drop probability at grade i, given a successful transmission at grade $i+1$.

5. Performance Metrics

Based on the steady-state probabilities of the previously described Markov chain, in this section, we determine expressions for the following performance metrics: energy consumption, packet loss probability, throughput, and source-to-sink delay. Initially, all the expressions are formulated in terms of the nodes' grade and, subsequently, the general performance of the network is obtained.

5.1. Energy Consumption. In order to evaluate the power consumption, we first calculate a related parameter: the average power consumption of a node at grade i. This parameter can be obtained as

$$P(i) = \frac{1}{T_c}\left[P_{tx}T_{tx}(i) + P_{rx}T_{rx}(i) + P_{\text{sp}}T_{\text{sp}}(i)\right], \quad (22)$$

where P_{tx}, P_{rx}, and P_{sp} are the average power consumption for a node in transmission, reception, and sleeping modes, respectively (which are usually specified by hardware manufacturers). In addition, $T_{tx}(i)$, $T_{rx}(i)$, and $T_{\text{sp}}(i)$ are the average time per cycle that a node spends in transmission, reception, and sleeping modes, respectively.

In order to estimate $T_{tx}(i)$, we analyze the node's activity during its transmission slot. As we mentioned before, a node at grade i awakes in this slot (i.e., it enters transmission mode) with probability $p_a(i)$. Given that the node is awake, the time that it spends in transmission mode is different for each of the following situations: (1) the channel is sensed busy; (2) the node transmits, but a collision occurs; or (3) the node successfully transmits. Let T_b, T_c, and T_s be the average time that a node is awake in each case, respectively; therefore $T_{tx}(i)$ can be expressed as

$$T_{tx}(i) = p_a(i)\left[p_b(i)T_b + p_c(i)T_c + p_s(i)T_s\right]. \quad (23)$$

For the sake of finding expressions for T_b, T_c, and T_s, recall that, in the proposed MAC protocol, a contending node uses a backoff counter whose value w is a uniform random variable in $[0, W-1]$. Hence, the backoff time is equal to σw, where σ is the duration of one minislot. In addition, recall that τ_{difs}, τ_{rts}, τ_{cts}, τ_{data}, τ_{ack}, and τ_{sifs} are the duration of the messages and interframe spaces involved in the protocol.

In the contention process, when the channel is sensed to be busy, the node goes immediately to sleep; hence, T_b can be obtained as

$$T_b = \tau_{\text{difs}} + \sigma W^{(t)}, \quad (24)$$

where $W^{(t)}$ is the average value of the backoff counter for a node that wins the contention (with or without collision). Notice that $\sigma W^{(t)}$ is not the average backoff time of the observed node, but the average backoff time sensed by this node.

On the other hand, when two or more nodes win the contention, that is, a collision occurs, they go to sleep as soon as they notice the absence of the CTS message (see Figure 3); hence, T_c can be written as

$$T_c = \tau_{\text{difs}} + \sigma W^{(c)} + \tau_{\text{rts}} + \tau_{\text{sifs}} + \tau_{\text{cts}}, \quad (25)$$

where $W^{(c)}$ is the average backoff counter for a node that experiences a collision.

When a node successfully transmits a packet, it goes to sleep after receiving the ACK message (see Figure 3). Hence, we obtain

$$T_s = T_{\text{difs}} + \sigma W^{(s)} + \tau_{\text{rts}} + \tau_{\text{cts}} + \tau_{\text{data}} + \tau_{\text{ack}} + 3\tau_{\text{sifs}}, \quad (26)$$

where $W^{(s)}$ is the average backoff counter for a node that successfully transmits a packet.

According to [16], the average backoff counter for a node that wins the contention, given that there are n additional contenders, can be written as

$$W_n^{(t)} = \frac{\sum_{w=0}^{W-1} w\left[((W-w)/W)^{n+1} - ((W-w-1)/W)^{n+1}\right]}{\sum_{w=0}^{W-1}\left[((W-w)/W)^{n+1} - ((W-w-1)/W)^{n+1}\right]}. \quad (27)$$

Similarly, if a node experiences a collision, given that there are n additional contenders, its average backoff counter can be expressed as

$$W_n^{(c)} = \frac{\sum_{w=0}^{W-1} w\left[((W-w)/W)^{n+1} - ((W-w-1)/W)^{n+1} - ((n+1)/W)((W-w-1)/W)^{n}\right]}{\sum_{w=0}^{W-1}\left[((W-w)/W)^{n+1} - ((W-w-1)/W)^{n+1} - (n/W)((W-w-1)/W)^{n}\right]}, \quad (28)$$

while the average backoff counter for a node that successfully transmits a packet, given that there are n additional contenders, can be obtained as

$$W_n^{(s)} = \frac{\sum_{w=0}^{W-1} w((W-w-1)/W)^{n}}{\sum_{w=0}^{W-1}((W-w-1)/W)^{n}}. \quad (29)$$

Lastly, if we consider the average value across all possible numbers n of contending nodes, we have $W^{(t)}$, $W^{(c)}$, and $W^{(s)}$ as

$$W^{(t)} = \sum_{n=0}^{N-1} C_n(i) W_n^{(t)},$$

$$W^{(c)} = \sum_{n=0}^{N-1} C_n(i) W_n^{(c)}, \quad (30)$$

$$W^{(s)} = \sum_{n=0}^{N-1} C_n(i) W_n^{(t)}.$$

Recall that $C_n(i)$ is the probability of n contenders besides the node of interest at grade i, and we defined it in (4). Notice that we conclude the estimation of $T_{tx}(i)$ after evaluating (30).

Now, we proceed to analyze the node's activity during its reception slot to estimate $T_{rx}(i)$. We establish that in SA-MAC a node awakes during its reception slot (i.e., it enters reception mode) only if its queue is not saturated. Given that it has awakened, the time that the node spends in reception mode depends on whether it receives a packet or not. If we denote by T_r and T_{nr} the average awake time in each case, we can express $T_{rx}(i)$ as

$$T_{rx}(i) = \left(1 - \rho_K^i\right)\left[p_r(i)\,T_r + \left(1 - p_r(i)\right)T_{nr}\right], \tag{31}$$

where ρ_K^i is the steady-state probability of saturated queue at the beginning of the reception slot, as we defined it in (21).

To calculate T_r, we simply observe in Figure 3 that it is equal to the average awake time for a node that successfully transmits a packet, T_s, which is defined in (26). Additionally, we assume that an awake node that does not receive a packet keeps waiting for an RTS message until the whole contention window has finished; hence

$$T_{nr} = \tau_{\text{difs}} + \sigma W + \tau_{\text{rts}}. \tag{32}$$

As a final step, we proceed to evaluate the average sleep time per cycle T_{sp}, which can be expressed as follows:

$$T_{\text{sp}}(i) = T_c - T_{tx}(i) - T_{rx}(i). \tag{33}$$

By substituting (23), (31), and (33) into (22), we finish the estimation of the average power consumption per cycle, for a node in grade i. Since we consider a constant number of nodes in each grade, the average power consumption per cycle, for any node in the LSN, takes the following form:

$$P_{\text{av}} = \frac{1}{I}\sum_{i=1}^{I} P(i). \tag{34}$$

5.2. Throughput and Packet Loss Probability. In an LSN, a packet that was generated at grade i has to accomplish i successful hops in order to reach the sink node; hence, the packet loss probability depends on this index. As it is expected, this probability is closely related to the network throughput; hence, in this subsection we analyze both of these parameters.

As we have explained in Section 4.1, $p_a(i)$ is the awakening probability for a node at grade i at the beginning of the transmission slot and $p_t(i)$ is the probability of winning the contention given that the node is awake (see (3) and (7), resp.). According to this, the product $p_a(i)p_t(i)$ represents the probability of removing one packet from the node's queue, in one cycle (recall that when a node wins the contention, it removes the packet from the head of its queue, even if a collision occurs). Notice that this product can also be interpreted as the rate at which packets leave a node at grade i, in packets per cycle.

Additionally, we define $p_r(i)$ as the packet reception probability for a node at grade i and ρ_K^i as the probability of having a saturated queue at the beginning of the reception slot (see (9) and (21), resp.); therefore, we can interpret the product $(1-\rho_K^i)p_r(i)$ as the rate at which packets from higher grades enter the queue of a node at grade i, in packets per cycle.

We assume that the generation packet rate per node in packets/cycle is λT_c. However, because of the finite queue length, the new-packet admission rate per node, $r_q(i)$, can be smaller than λT_c; that is, some packets are dropped in its source node.

In order to calculate $r_q(i)$, we take into consideration that, at any queue, the input and output rates must be equal. In a node's queue, the former is given by $(1-\rho_K^i)p_r(i)+r_q(i)$, while the latter is $p_a(i)p_t(i)$. According to this, we obtain that

$$r_q(i) = \begin{cases} p_a(i)\,p_t(i) - \left(1 - \rho_K^i\right)p_r(i), & 1 \le i \le I-1, \\ p_a(i)\,p_t(i), & i = I. \end{cases} \tag{35}$$

This rate allows us to calculate the new-packet drop probability at grade i, as

$$p_{\text{ND}}(i) = 1 - \frac{r_q(i)}{\lambda T_c}. \tag{36}$$

On the other hand, given that a packet has been admitted in the queue of a node at grade j, it will reach the next grade, only if a collision does not occur during its transmission and if the queue of the receiver node is not saturated. The probability of no collision during a transmission is $p_s(j)/p_t(j)$, while the probability of nonsaturated queue in the next grade is $(1 - \rho_K^{j-1})$; therefore, the probability of accomplishing a successful hop is $(1-\rho_K^{j-1})p_s(j)/p_t(j)$. According to this, and considering that no packets are dropped at the sink node, a packet admitted at grade i is delivered at the sink node with probability $p_{\text{DS}}(i)$, which can be written as

$$p_{\text{DS}}(i) = \frac{p_s(i)}{p_t(i)}\prod_{j=1}^{i-1}\left[\left(1 - \rho_K^j\right)\frac{p_s(j)}{p_t(j)}\right]. \tag{37}$$

By multiplying (37) with $Nr_q(i)$, we find the rate $r_s(i)$ at which the generated packets at grade i reach the sink node, which is given by

$$r_s(i) = Nr_q(i)\,\frac{p_s(i)}{p_t(i)}\prod_{j=1}^{i-1}\left[\left(1 - \rho_K^j\right)\frac{p_s(j)}{p_t(j)}\right]. \tag{38}$$

By using (38), and observing that the packet generation rate per grade is $N\lambda T_c$, we obtain the packet loss probability $p_{\text{PL}}(i)$ for a packet generated at grade i as

$$p_{\text{PL}}(i) = 1 - \frac{r_s(i)}{N\lambda T_c}. \tag{39}$$

Now, let Th_i be the throughput at grade i, that is, the rate at which packets are transferred from grade i to grade $i-1$. It is important to clarify that our definition of Th_i includes all the transferred packets between two adjacent grades, without regarding the source grade of these packets. Additionally, similar to the previous analyses in this subsection, we

interpret the product $p_a(i)p_s(i)$ as a rate in packets/cycle. Based on these considerations, Th_i can be given by

$$\mathrm{Th}_i = Np_a(i)\,p_s(i)\left(1-\rho_K^{i-1}\right), \tag{40}$$

for $2 \le i \le I$, and $\mathrm{Th}_1 = Np_a(1)p_s(1)$ because we assume that no packet drop occurs at the sink node.

Notice that Th_1 is particularly relevant, because it represents the rate at which the packets from all the LSN are transferred to the sink node. In other words, Th_1 represents the whole network throughput Th. Hence,

$$\mathrm{Th} = Np_a(1)\,p_s(1), \tag{41}$$

in packets per cycle.

In order to give a deeper insight on the relation between Th and $p_{\mathrm{PL}}(i)$, in the following, we propose an alternative way to calculate $r_s(i)$.

Let $R_s(i)$ be the rate at which the packets generated at grades i to I reach the sink node. This rate can be found as the product of Th_i (the packet admission rate at grade $i-1$) and $p_{\mathrm{DS}}(i-1)$ (the probability of traveling from grade $i-1$ to the sink node); as a result, we obtain

$$R_s(i) = Np_a(i)\,p_s(i)\prod_{j=1}^{i-1}\left[\left(1-\rho_K^j\right)\frac{p_s(j)}{p_t(j)}\right]. \tag{42}$$

According to the definitions of $R_s(i)$ and $r_s(i)$, it is clear that $r_s(i) = R_s(i) - R_s(i+1)$, for $1 \le i \le I-1$, and $r_s(I) = R_s(I)$. Hence,

$$\sum_{i=1}^{I} r_s(i) = R_s(1) = Np_a(1)\,p_s(1) = \mathrm{Th}. \tag{43}$$

In other words, the sum of the rates at which the packets from every grade i ($1 \le i \le I$) reach the sink node is equal to the network throughput, as it is expected.

5.3. Delay. Based on the methods proposed in [16, 19], in this subsection we show how to estimate the average source-to-sink delay D_i for a packet that is generated at grade i. This parameter can be calculated as the sum of the delays experienced by the packet at each of the grades that it has to travel to reach the sink node. Let d_h be the average delay that a packet experienced at grade h. Hence,

$$D_i = \sum_{h=1}^{i} d_h. \tag{44}$$

Now, d_h can be calculated as the sum of three parts; that is, $d_h = d_h(1) + d_h(2) + d_h(3)$, where

(i) $d_h(1)$ is the average delay from the instant when the packet arrives at grade h to the beginning of the next cycle;

(ii) $d_h(2)$ is the average delay from the beginning of the first cycle after the arrival, to the instant when the packet reaches the head of the queue.

(iii) $d_h(3)$ is the average delay from the instant when the packet reaches the head of the queue to the instant it is transmitted to the next grade.

In order to compute $d_h(1)$, we recall that packet generation at each node occurs according to a Poisson process. As a consequence, the new-packets arrival instants are uniformly distributed along a cycle, which means that the average delay from the packet arrival instant to the beginning of the next cycle is $d_h(1) = T_c/2$ in the source node. Otherwise, the packet arrives at the beginning of a cycle; hence, $d_h(1) = 0$, if $h \ne i$. To summarize, $d_h(1)$ can be expressed as

$$d_h(1) = \begin{cases} \dfrac{T_c}{2}, & h = i, \\ 0, & \text{otherwise.} \end{cases} \tag{45}$$

On the other hand, $d_h(2)$ depends on how many packets are in the queue when the observed packet arrives and on the average time to advance one position in the queue (δ). The latter of these parameters depends on the average number of failed attempts to transmit, which in turn is a function of the probability of transmitting in one cycle, $p(i)p_t(i)$. According to this, δ can be expressed as

$$\begin{aligned}\delta &= T_c\sum_{j=0}^{\infty} j\left[1-p(i)\,p_t(i)\right]^{j-1}p(i)\,p_t(i)\\ &= \frac{T_c}{p(i)\,p_t(i)}.\end{aligned} \tag{46}$$

Then, by considering all possible values of the queue's length (except K, because an admitted packet cannot find a saturated queue), we obtain

$$d_h(2) = \frac{\delta}{1-\pi_K}\sum_{k=0}^{K-1} k\pi_k. \tag{47}$$

From a similar analysis, we can express the average time that a packet spends at the head of the queue as

$$\begin{aligned}d_h(3) &= T_c\sum_{j=0}^{\infty} j\left[1-p(i)\,p_t(i)\right]^{j}p(i)\,p_s(i)\\ &= T_c\frac{p(i)\,p_s(i)\left[1-p(i)\,p_t(i)\right]}{\left[p(i)\,p_t(i)\right]^2}.\end{aligned} \tag{48}$$

Notice that $d_h(3)$ depends not only on $p(i)p_t(i)$, but also on the probability of successful transmission in one cycle, $p(i)p_s(i)$.

6. Criteria for Selecting the Awakening Probabilities

As we mentioned in Section 3, if an excessively small value of $p(i)$ is selected, it is probable that none of the nodes wakes up in some of the transmission slots, hence reducing the network throughput. On the other hand, if we simply select $p(i) = 1$

(or select a large value), possibly, a lot of collisions will occur and the network throughput will be also diminished. On the basis of these considerations, we propose to select $p(i)$ in such a way that the throughput in the corresponding grade is maximized.

In (40), we established the throughput of grade i. Notice that, by substituting (3), (4), and (8) into (40), we get an expression where $\mathrm{Th}(i)$ is explicitly given in terms of $p(i)$. By taking the first derivative of this expression and equating it to zero, we obtain

$$\sum_{n=0}^{N-1} \binom{N-1}{n} s_n (q_e p)^n (1 - q_e p)^{N-n-2} (n + 1 - N q_e p) = 0, \tag{49}$$

where $q_e(i) = 1 - p_e(i)$ and, for the sake of space, we have omitted the dependence on i in $p(i)$ and $q_e(i)$.

By solving (49) in terms of $p(i)$, we can obtain the awakening probability that maximizes $\mathrm{Th}(i)$. However, this cannot be directly calculated, since $p_e(i)$ depends on $p(i)$ (see Section 4.2). In order to overcome this problem, we propose to calculate an approximation for $p_e(i)$, which will only be used to solve (49). This approximation is denoted by $p_e^*(i)$ and must be substituted into (49) instead of $p_e(i)$.

As it was mentioned above, $p(i)$ must be selected in order to avoid collisions. Hence, our approximation considers each node at grade i as a K-length queue with one server, where packets arrive at rate $\lambda Tc + p_r(i)$ packets per cycle, and the mean service time is N cycles. Notice that the latter of these assumptions is a direct consequence of considering that the probability of collision is very close to zero.

Additionally, numerical evaluations show that the selection of $p(i)$ does not need to be very strict, because $d\mathrm{Th}/dp(i)$ is very close to zero in a wide range of values around the optimum one. Based on this observation, and for the sake of simplifying our analysis, we model a node's queue as an $M/M/1/K$ system, only for the purpose of estimating $p_e(i)$. According to this, the approximation of $p_e(i)$ can be calculated as

$$p_e^*(i) = \frac{1 - \alpha_i}{1 - \alpha_i^{K+1}}, \tag{50}$$

where $\alpha_i = N[\lambda T_c + p_r(i)]$ is the offered traffic to nodes at grade i.

Lastly, it is important to notice that $p_e^*(i)$ depends on the behavior of nodes in higher grades, since $p_r(i) = p_a(i+1)p(i+1)$. Therefore, for the purpose of calculating $p_e^*(i)$ and consequently $p(i)$, the higher grades must be previously solved, according to the method described in the end of Section 4.2.

7. Numerical Results and Discussion

The DTMC evaluation is carried out through the numerical solution explained in Section 4.2. As we have mentioned before, this solution involves the application of the Gauss-Seidel method; we have evaluated it with 5×10^4 iterations, with an approximation error of 5×10^{-4}.

Table 1: Network parameters values.

Parameter	Value
ξ	18
σ	1 ms
τ_{ack}	11 ms
τ_{cts}	11 ms
τ_{data}	43 ms
τ_{difs}	10 ms
τ_{rts}	11 ms
τ_{sifs}	5 ms
P_{rx}	59.9 mW
P_{sp}	0 mW
P_{tx}	52.2 mW
W	64

In addition, we have developed a discrete-event simulation for SA-MAC to validate our analyses, which was implemented in Matlab and executed over 1×10^5 cycles. In every cycle, we simulated all the events that are included in the staggered schedules of the grades, and we provide a brief description of them in the following paragraph.

At the beginning of a transmission slot, we identify all nodes that have packets to transmit, and these nodes wake up with probability $p(i)$. Subsequently, we generate a backoff counter for every awake node and, then, we carry on the contention process. Once this process is over, we update statistics related to the packet transmission. Simultaneously, we simulate the reception slot for the next grade. In this case, we identify which nodes have nonsaturated queues and we execute for them the reception events, and we also update statistics about the packet reception. We simulate the transmission/reception between every couple of adjacent nodes, including the process between grade 1 and the sink node. It is important to mention that, along these processes, nodes are generating new packets at a rate λ. In all the transmission/reception events, we consider an ideal channel.

In this section, we first present numerical evaluations for the awakening probability selection: in terms of traffic load, from low to high load with $\lambda \in \{0.001, 0.003, 0.01, 0.03\}$, in a dense network ($N = 20$), and in terms of network density, from low to high node density with $N \in \{10, 20, 30\}$, and $\lambda = 0.003$.

Second, we evaluate SA-MAC performance in terms of throughput, power consumption, average source-to-sink delay, and new-packet drop probability. In order to highlight the advantages of SA-MAC over previous proposals, we compare its performance with PRI-MAC under different scenarios. This helps us to recognize how the number of nodes per grade and traffic load affect these performance parameters.

Finally, we present results on the grade analysis, where we can highlight that packet loss probability and source-to-sink delay are not the same for every grade, whereas average power consumption is different for remote grades.

In all our numerical evaluations, we select $I = 7$ (number of grades in the network) and $K = 15$ (queue's length), and we take from [16, 19] the rest of the network parameter values, which are shown in Table 1.

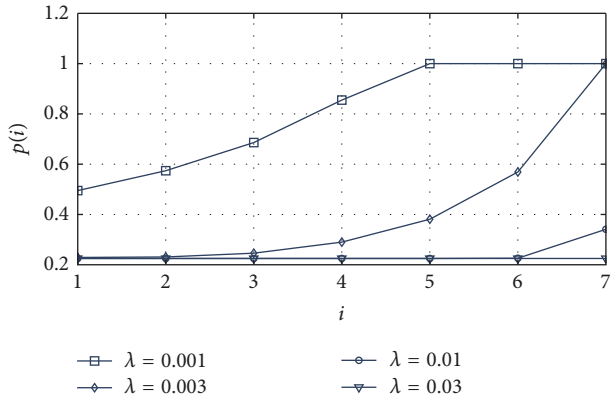

Figure 5: Awakening probability, $p(i)$, as a function of grade (i) and packet rate generation (λ), for $N = 20$.

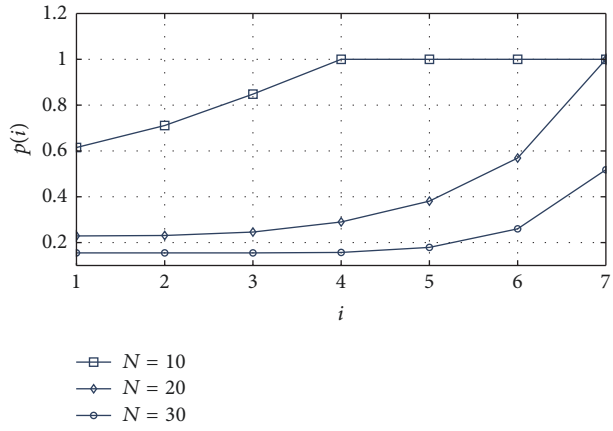

Figure 6: Awakening probability as a function of the grade (i) and the number of nodes per grade (N), for $\lambda = 0.003$.

7.1. Awakening Probability Evaluation. In Figure 5, we show the selected awakening probability for a node with packets to transmit, $p(i)$, as a function of the grade number i and for different values of the packet generation rate (λ) when $N = 20$. In Figure 6, we also show this probability, but in terms of the number of nodes per grade (N) for the case when $\lambda = 0.003$. These results were obtained through the approximation method described in Section 6.

From Figure 5, we can observe that if λ increases, the selected $p(i)$ must decrease because the higher the value of λ, the higher the probability that multiple nodes have packets to transmit; therefore, it is necessary to avoid collisions by reducing the number of contenders. Similarly, large values of N require the selection of small values of $p(i)$, as it can be observed in Figure 6.

Additionally, it is interesting to point out that, in both Figures, $p(i)$ is smaller for nodes closer to the sink (small values of i). This is a consequence of the fact that these nodes accumulate packets from higher grades. As a result, their probability of having packets to transmit is higher, and collisions must be avoided by awaking a small proportion of them in each cycle, that is, selecting a small value of $p(i)$.

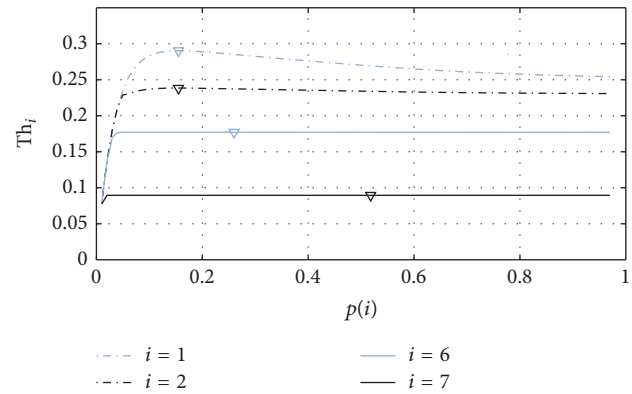

Figure 7: Throughput per grade, in packets/s, as a function of $p(i)$ in comparison with the throughput obtained by approximating $p(i)$, for $\lambda = 0.003$ and $N = 30$. Triangles refer to the approximated maximum value of Th_i.

We also analyze the effect of using the proposed approximation method for determining $p(i)$ on the system throughput by exhaustively evaluating Th_i over the whole domain of $p(i)$. These results are shown in Figure 7, where the triangles indicate the points ($p(i)$, Th_i) computed by the approximation method. In this scenario ($\lambda = 0.003$ and $N = 30$), Th_i exhibits the same performance for $i \in [3, 6]$ and, hence, we only show the results for Th_6.

In this scenario, the throughput in remote grades (e.g., $i = 7$) reaches its maximum over a wide range of values of $p(i)$. This is because the arrival packet rate is so low that small values of $p(i)$ do not create bottlenecks and large values do not generate collisions. By contrast, in grades where the traffic load is much higher, the throughput reaches its maximum only if $p(i)$ is adequately selected.

These results show that, independently of the traffic load per grade, the proposed method is capable of finding the values of $p(i)$ that maximize the throughput. On the basis of this observation, in the remaining numerical evaluations, we use the approximation method for selecting the values of $p(i)$.

Additionally, it is important to point out that the wide range of values of $p(i)$ that provide the maximum throughput will allow, in future work, to select this probability in such a way as to optimize not only the throughput, but also other performance parameters. For example, further energy savings could be achieved, but they must be constrained by target levels of source-to-sink delay or packet loss probability. Another approximation may include the selection of the awakening probabilities as a function of the queue's length of nodes, so that the packet loss probability or the source-to-sink delay can also be optimized, while satisfying energy consumption constraints.

These new proposals may require redefining equations in Section 6; however, the DTMC and performance parameters definitions would suffer small changes.

7.2. SA-MAC Protocol Evaluation. In Figure 8 we show the network throughput attained by SA-MAC and PRI-MAC. In SA-MAC, the throughput is almost constant in spite of

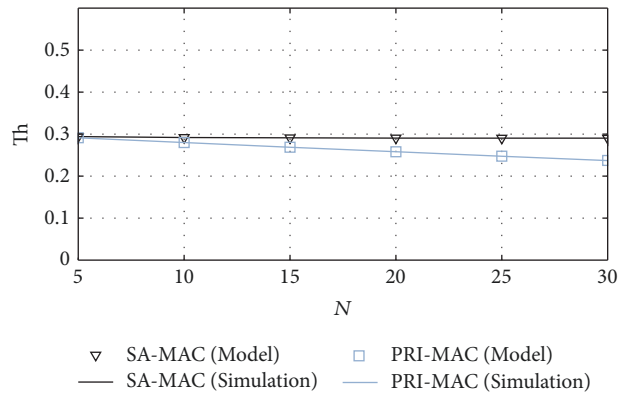

Figure 8: Throughput, in packets/s, as a function of the number of nodes per grade (N), for $\lambda = 0.03$.

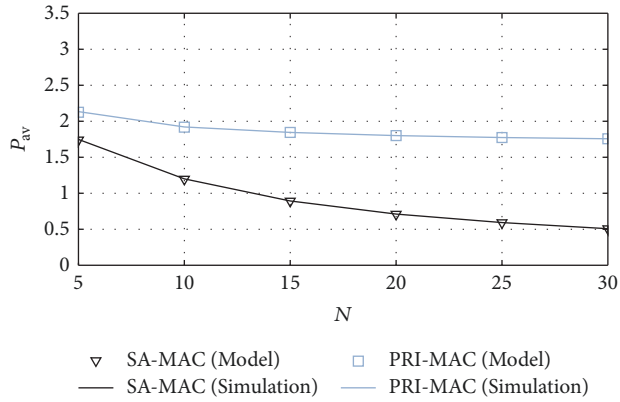

Figure 9: Average power consumption per node, in mW, as a function of the number of nodes per grade (N), for $\lambda = 0.03$.

the increase in N. The explanation for this performance is that $p(i)$ is selected to maximize the throughput (it must be noticed that, given the network parameters in Table 1, the maximum throughput, $1/T_c$, is 0.31 packets/sec). By contrast, since PRI-MAC does not consider a selective awakening, that is, $p(i) = 1$ regardless of the network parameters, a lot of collisions occur for large values of N in the grades which are closer to the sink, and, as a result, the throughput performance is degraded. Consequently, it can be concluded that SA-MAC is better suited for high-density networks than PRI-MAC.

In Figure 9 we show the average power consumption for both protocols. It can be observed that a significant reduction of power consumption can be achieved with SA-MAC, in comparison with that of PRI-MAC; and it must be highlighted that this reduction is quite notorious for large values of N. This performance is achieved in SA-MAC because the energy wasted in collisions is very small; moreover, further savings are achieved by avoiding the awakening of multiple nodes in a network that is able to transmit only one packet per contention. Besides, in SA-MAC, the wasted energy is also reduced because the nodes with a saturated queue do not awake in the reception mode.

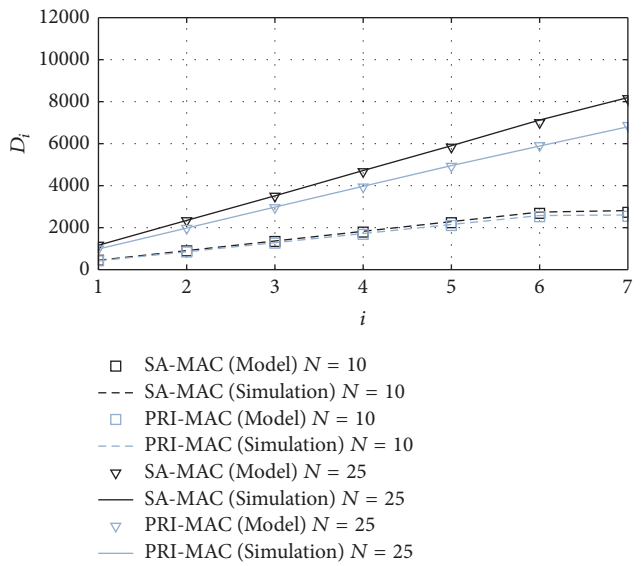

Figure 10: Average source-to-sink delay, in seconds, as a function of grade (i) and number of nodes per grade (N), for $\lambda = 0.03$.

Notice that the joint analysis of throughput and power consumption reveals the main advantage of SA-MAC: its adaptation to different levels of traffic load allows it to save a lot of energy (hence, extending the network lifetime), without sacrificing the throughput performance.

It is also important to notice that, in some scenarios, this advantage comes with a moderate degradation of the source-to-sink delay, as it is shown in Figure 10. This was expected because our proposal reduces collisions by probabilistically turning off nodes with packets to transmit, which increments the waiting time experienced by the packets in the sleeping nodes. This degradation is more apparent for large values of N, as we can observe in Figure 10. This is because the larger the value of N, the smaller the value of $p(i)$. Therefore, on average, nodes will have to wait for longer periods of time to attempt a transmission.

This increase in delay also affects the new-packet drop probability, $p_{\mathrm{ND}}(i)$, because large delays reduce the space in the buffers to admit new generated packets. Since delay is larger for SA-MAC in comparison with PRI-MAC, $p_{\mathrm{ND}}(i)$ is also slightly larger for the former, as it is shown in Figure 11. A similar situation occurs in terms of N: when this parameter increases, $p_{\mathrm{ND}}(i)$ also increases in both schemes.

When $p_{\mathrm{ND}}(i)$ is analyzed in terms of the grade i, a special case arises for the last grade ($i = 7$ in our experiments), where buffers are dedicated only to admit the locally generated packets; therefore, $p_{\mathrm{ND}}(7)$ is lower in comparison with other grades.

From this analysis, it is possible to identify network conditions where both protocols exhibit similar performance for throughput, source-to-sink delay, and drop probability, but where SA-MAC significantly outperforms PRI-MAC in terms of energy consumption; for instance, when $N = 10$ the power consumption incurred by SA-MAC is around 63% of that of PRI-MAC.

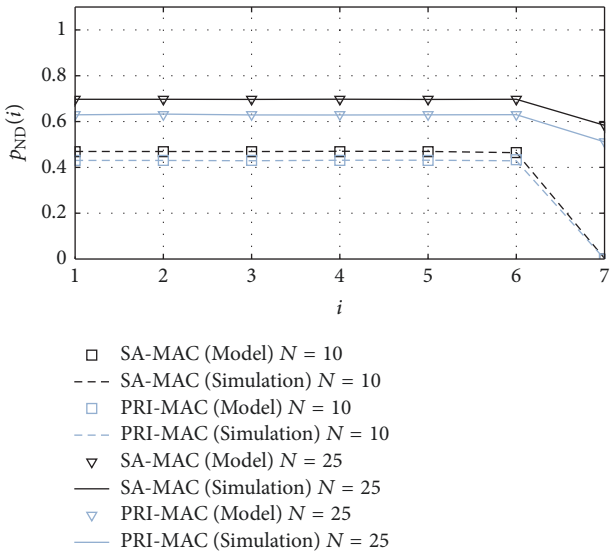

FIGURE 11: New-packet drop probability as a function of grade (i) and number of nodes per grade (N), for $\lambda = 0.03$.

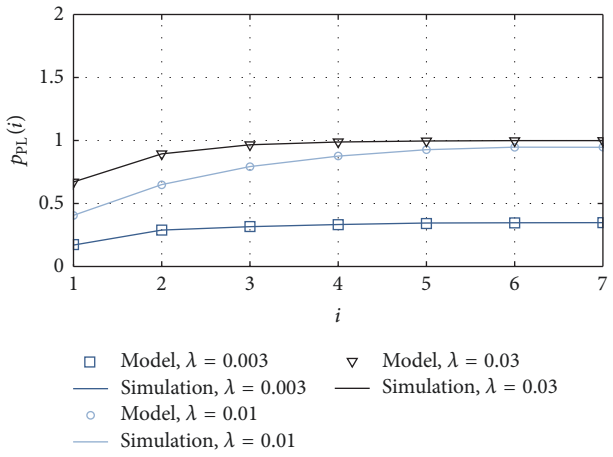

FIGURE 12: Packet loss probability as a function of grade (i) and generated-packet rate (λ), for $N = 20$.

7.3. Per-Grade Network Performance. In Figures 12, 13, and 14 we show the analysis by grade for packet loss probability $p_{PL}(i)$, average power consumption $P(i)$, and source-to-sink delay D_i, respectively, under different scenarios.

As it was expected, the packet loss probability increases as the distance from the sink also increases, because a packet generated in a remote node needs to overcome a large number of hops. This observation leads to the conclusion that prioritization schemes must be proposed in future works for packets generated in remote nodes. In addition, it is interesting to observe that this probability exhibits smooth variations in terms of i for small traffic loads (e.g., $\lambda = 0.003$).

In Figure 13 we can observe that, in general, power consumption increases as the traffic load decreases. This effect is a direct consequence of the strategy used to turn off nodes during the reception slot: since a node goes to sleep when its queue is full, under low traffic loads, nodes stay

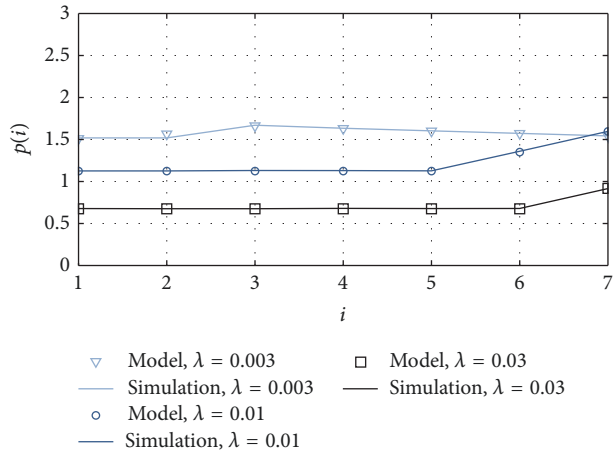

FIGURE 13: Average power consumption per node, in mW, as a function of grade (i) and generated-packet rate (λ), for $N = 20$.

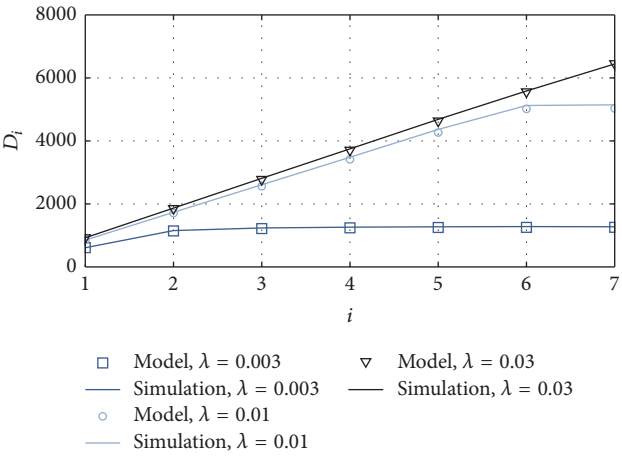

FIGURE 14: Average source-to-sink delay, in seconds, as a function of grade (i) and generated-packet rate (λ), for $N = 20$.

awake most of the time. On account of these observations, it is clear that in future works a node must go to sleep (during the transmission slot) not only when its queue is saturated, but also when the traffic load is very low.

In order to interpret the power consumption in each grade, we highlight that as i increases, the transmission power consumption decreases (since the nodes have less packets to relay), while the reception power consumption increases (since the nodes keep awake because their queues are not saturated). For $\lambda = 0.003$, the decreasing component (i.e., the transmission power consumption) is dominant when $i \geq 3$, whereas the opposite occurs otherwise; as a result, grade 3 is the least power efficient in the whole network (see Figure 13). Notice that the less efficient grade will depend on traffic load; it can even be grade I when the traffic load is high (e.g., $\lambda = 0.01$), or grade 1 when the traffic load is very low.

The previous observations also lead us to identify that the network lifetime is not determined by the average power consumption in the whole network, but by the average power consumption in the less efficient grade.

Lastly, in Figure 14, it is possible to appreciate that, under low traffic loads, the source-to-sink delay is almost the same, regardless of the packet source (e.g., $\lambda = 0.003$). This phenomenon is a consequence of the pipelined scheduling, which makes possible for a packet to go through the whole network in less than one cycle, when such packet finds an empty queue in every node (which is only possible under very low traffic loads). Under moderate traffic loads (e.g., $\lambda = 0.01$), the previously described phenomenon occurs only in the most remote grades (where multiple hops are possible during one cycle), but in the closest grades to the sink, a significant extra delay is accumulated for each accomplished hop. When the traffic load is large, this significant extra delay is accumulated in all the hops; therefore, the average delay is directly proportional to the number of hops to reach the sink.

8. Conclusions and Future Work

In this paper, we have introduced SA-MAC, an energy-efficient MAC protocol for dense LSN with applications to IoT. SA-MAC is a synchronized duty-cycle MAC protocol with a pipelined scheduling where nodes selectively awake based on the node density per grade and on the state of their data queues.

As part of SA-MAC, we introduced a set of awakening probabilities that maximizes the throughput per grade by reducing the number of packet collisions. Our numerical evaluations showed that, in order to select appropriate values for these probabilities, the network designer should also consider the grade, the data packet generation rate, and the number of nodes per grade.

Additionally, we have developed a DTMC-based analytical model for the performance of SA-MAC that was validated by means of extensive discrete-event simulations. The results obtained from simulations and mathematical analysis show that SA-MAC outperforms previous work in terms of energy efficiency, packet loss probability, and throughput. The latter is particularly true under certain IoT-like conditions, where both high node density and high traffic load are possible scenarios.

Another contribution of this work is the first *per-grade* detailed analysis of the performance of an LSN. This deeper analysis revealed, among other things, that there is a set of specific grades that have a larger impact over the network lifetime and reliability.

We believe that the analysis per grade deserves further investigation, for instance, to include other objective functions besides throughput, such as packet drop probability, source-to-sink delay, or energy consumption. It is even possible to formulate the problem as a multiobjective optimization problem where two or more performance metrics are jointly optimized. Other extensions of this work also include prioritization schemes for QoS provisioning and new schemes for selecting the awakening probabilities as a function of the queue's length that also optimize the main network performance parameters. This new scheme may lead to further reductions in the number of contending nodes and to the design of new mechanisms for assigning priorities to data packets that can also help reducing the packet loss probability and the source-to-sink delay, all this while preserving energy in order to increase the network lifetime.

Appendix

Convergence Analysis of the Iterative Method to Compute $p_e(i)$ and $p_t(i)$

In Section 4.2 we propose a method to iteratively find $p_e(i)$ and $p_t(i)$. In this appendix, we analyze the convergence of such a method. In order to simplify our nomenclature, we omit the dependence on i from π_0^i, $p_e(i)$, and $p_t(i)$.

Firstly, we define $p_t = f(p_e)$ and $p_e = g(p_t)$, where $f(p_e)$ represents (3)–(5) and (7), while $g(p_t)$ is implicit in the DTMC solution (recall that p_e is equal to π_0, the steady-state probability of an empty queue). The problem that our method solves is to find a pair (p_t^*, p_e^*) that simultaneously satisfy these functions.

To analyze this problem in terms of only one variable, instead of $p_e = g(p_t)$, we use the equivalent expression $p_t = g^{(-1)}(p_e)$, where $g^{(-1)}(x)$ is the reciprocal function of $g(x)$. Therefore, the problem can be restated as finding a value p_e^* that satisfies $f(p_e^*) = g^{(-1)}(p_e^*)$. In the following, we provide some arguments to show that a unique solution exists for this problem.

By analyzing (3)–(5) and (7), we can establish the following:

(i) Even if the contenders (besides the node of interest) always have packets to transmit ($p_e = 0$), there exists a probability $p_t^{\min} > 0$ that the node of interest wins the contention; hence

$$f(0) = p_t^{\min} > 0. \tag{A.1}$$

(ii) Only if the contenders (besides the node of interest) never have packets to transmit ($p_e = 1$), the node of interest wins the contention with certainty ($f(1) = 1$); hence,

$$f(p_e) < 1, \quad \text{if } p_e < 1. \tag{A.2}$$

Then, according to the DTMC, we can also establish the following:

(i) If an awaken node never wins the contention ($p_t = 0$), the probability of an empty queue has to be zero; hence,

$$g(0) = 0. \tag{A.3}$$

(ii) Even if a node transmits every time it awakens ($p_t = 1$), this node may accumulate packets while sleeping; hence,

$$g(1) = p_e^{\max} < 1. \tag{A.4}$$

Notice that (A.3) and (A.4) can be rewritten as $g^{-1}(0) = 0$ and $g^{-1}(p_e^{\max}) = 1$ (where $p_e^{\max} < 1$), and, when they are combined with (A.1) and (A.2), we obtain

$$\begin{aligned} f(0) &> g^{-1}(0) = 0, \\ f(p_e^{\max}) &< g^{-1}(p_e^{\max}) = 1, \end{aligned} \tag{A.5}$$

which means that these curves cross each other. In other words, (A.5) imply that, for any set of network parameter values, at least, one solution exists. Furthermore, the possible solutions satisfy that $p_e^{\min} \leq p_e^* \leq p_e^{\max}$, where $p_e^{\min} = g(p_t^{\min})$.

One way to demonstrate that there exists only one solution consists in proving that both functions are monotonically increasing and convex for $p_e^{\min} \leq p_e \leq p_e^{\max}$. The first of these properties is verifiable for both functions by analyzing the network operation:

(i) According to (3)–(5) and (7), if the probability of empty queue increases (p_e), then, the probability that a node of interest wins the contention also increases; that is, $f(p_e)$ is monotonically increasing.

(ii) According to the DTMC, if the probability that a node transmits a packet (p_t) increases, then, the probability of empty queue also increases; that is, $g(p_t)$ is monotonically increasing, which also implies that $g^{(-1)}(p_e)$ is monotonically increasing.

On the other hand, the convexity of $f(p_e)$ is also demonstrable because the second derivative of (7) is positive. Unfortunately, we have not yet developed an analytical expression for $g^{(-1)}(p_e)$; however, according to extensive numerical evaluations, we identify that this function is also convex in $p_e^{\min} \leq p_e \leq p_e^{\max}$.

In the following, we analyze our iterative method under the assumption that $f(p_e)$ and $g^{(-1)}(p_e)$ satisfy all the characteristics discussed above. However, it is clear that a deeper study on the features of $g^{(-1)}$ must be accomplished in future work in order to analytically demonstrate its convexity.

Let (p_t^k, p_e^k) be the approximated solution that the iterative method provides at the kth iteration and let p_e^0 be an arbitrary initial value for p_e; then, according to our previous descriptions, the kth iteration includes the following two steps:

(1) Substituting p_e^k in (3)–(5) and (7) to obtain $p_t^k = f(p_e^k)$

(2) Using p_t^k to solve the DTMC and obtain $p_e^{k+1} = g(p_t^k)$.

To show that the method always converges, firstly, we point out that, as a consequence of the characteristics of $f(p_e)$ and $g^{(-1)}(p_e)$ discussed above, it is true that

$$g^{-1}(p_e) < f(p_e), \quad \text{for } p_e < p_e^*, \tag{A.6}$$

$$g^{-1}(p_e) > f(p_e), \quad \text{for } p_e > p_e^*. \tag{A.7}$$

Now, let us assume that we select an initial value $p_e^0 < p_e^*$; then, at step (1), the method obtains $p_t^0 = f(p_e^0) < p_t^* = f(p_e^*)$, because $f(x)$ is monotonically increasing. At the second step, the method finds a new value p_e^1 that satisfies $p_t^0 = g^{(-1)}(p_e^1)$; since $g^{(-1)}(x)$ is also monotonically increasing and $p_t^0 < p_t^*$, we can establish that $p_t^0 = g^{(-1)}(p_e^1) < p_t^* = g^{(-1)}(p_e^*)$; hence $p_e^1 < p_e^*$. Additionally, to go from step (1) to step (2), the method establishes that $p_t^0 = f(p_e^0) = g^{(-1)}(p_e^1)$; consequently, $p_e^1 > p_e^0$, in order to satisfy (A.6). These arguments can be summarized as

$$p_e^0 < p_e^1 < p_e^*, \quad \text{if } p_e^0 < p_e^*. \tag{A.8}$$

Notice that the previous series of arguments are true at any iteration, so we can generalize (A.8) as follows:

$$p_e^k < p_e^{k+1} < p_e^*, \quad \text{if } p_e^0 < p_e^*. \tag{A.9}$$

If we repeat the previous analysis, but assuming that $p_e^0 > p_e^*$, we find that

$$p_e^* < p_e^{k+1} < p_e^k, \quad \text{if } p_e^0 > p_e^*. \tag{A.10}$$

Inequalities (A.9) and (A.10) indicate that the method always converges toward p_e^*, for any initial value p_e^0.

In Figure 15, we illustrate three iterations of this method for a specific scenario ($i = 7$, $\lambda = 0.01$, and $N = 20$).

Lastly, we estimate the rate of convergence of this method. According to [27], the rate of convergence of a sequence $\{x^k\}$ to x^* is of order r with asymptotic error constant C, if

$$\lim_{k\to\infty} \frac{\left|x^{k+1} - x^*\right|}{\left|x^k - x^*\right|^r} = C. \tag{A.11}$$

Considering this definition, we execute our method for an extremely small target error (e.g., 1×10^{-15}) and we assume that $p_e^L = p_e^*$, where L is the number of iterations that provide this target error. Then, we estimate r and C, for $k \gg 1$ and $k \ll L$, that is, for values of k that satisfy the limit in (A.11) but provide large errors in comparison with L.

We find that our method always converges linearly ($r = 1$); however, the asymptotic error constant can vary for different sets of parameters values, as we summarize in Table 2.

Table 2: Asymptotic error constant, for $i = 7$ and $\lambda = 0.01$.

N	p	Error	L	p_e^*	C
10	1.000	1×10^{-15}	20	0.961	0.167
20	0.341	1×10^{-12}	31	0.848	0.416
30	0.165	1×10^{-9}	107	0.357	0.843

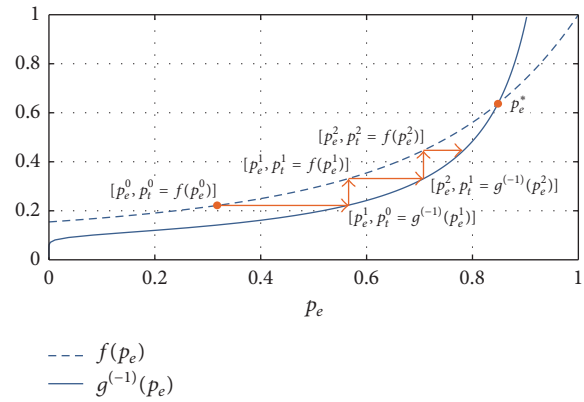

Figure 15: Iterative method to calculate p_e^*, for $i = 7$, $\lambda = 0.01$, and $N = 20$.

Notations

Main Variables Used in the Analysis

a_k: Probability that a node generates k packets during a cycle
A_l: Probability that a node generates l or more packets during a period of time T_c^*
$C_n(i)$: Probability of n contenders besides the node of interest in grade i
D_i: Average source-to-sink delay for a packet that is generated at grade i
I: Number of grades in the LSN
K: Maximum number of packets that a node can buffer
N: Number of nodes in each grade
$p(i)$: Awaking probability given a nonempty queue
$p_a(i)$: Unconditional awaking probability
$p_b(i)$: Probability of detecting a busy channel (for an awake node of interest)
$p_c(i)$: Probability of collision (for an awake node of interest)
$p_e(i)$: Probability of empty queue
$P_{m,n}^i$: Probability of transition from state m to state n
$p_{\mathrm{ND}}(i)$: New-packet drop probability
$p_{\mathrm{pl}}(i)$: Packet loss probability
$p_r(i)$: Packet reception probability
$p_s(i)$: Probability of successful packet transmission (for an awake node of interest)
$p_t(i)$: Probability of winning the contention (for an awake node of interest)
$P(i)$: Average power consumption of a node in grade i
P_{av}: Average power consumption of a node
P_{rx}: Average power consumption for a node in reception mode
P_{sp}: Average power consumption for a node in sleeping mode
P_{tx}: Average power consumption for a node in transmission mode
$r_s(i)$: Rate at which the generated packets at grade i reach the sink node
$R_s(i)$: Rate at which the generated packets at grades i to I reach the sink node
s_n: Probability of successful packet transmission given n additional contenders
T: Slot duration
T_c: Cycle duration
T_c^*: Period of time between the beginning of a transmission slot and the beginning of the next reception slot
t_n: Probability of winning the contention (with or without collision) given n additional contenders
Th: Network throughput
Th_i: Throughput at grade i
W: Maximum backoff counter
λ: Packet generation rate per node
ξ: Number of slots that a node spends in sleeping mode per cycle
π_m^i: DTMC steady-state probabilities
ρ_k^i: Relayed-packet drop probability at grade i
σ: Duration of one minislot.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] F. K. Shaikh, S. Zeadally, and E. Exposito, "Enabling technologies for green internet of things," *IEEE Systems Journal*, vol. 11, no. 2, pp. 983–994, 2017.

[2] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 9324035, 25 pages, 2017.

[3] J. Wang, C. Hu, and A. Liu, "Comprehensive Optimization of Energy Consumption and Delay Performance for Green Communication in Internet of Things," *Mobile Information Systems*, vol. 2017, Article ID 3206160, 17 pages, 2017.

[4] M. Kocakulak and I. Butun, "An overview of Wireless Sensor Networks towards internet of things," in *Proceedings of the 7th IEEE Annual Computing and Communication Workshop and Conference, CCWC 2017*, January 2017.

[5] A. P. Abidoye and I. C. Obagbuwa, "Models for integrating wireless sensor networks into the Internet of Things," *IET Wireless Sensor Systems*, vol. 7, no. 3, pp. 65–72, 2017.

[6] S. Kartakis, W. Yu, R. Akhavan, and J. A. McCann, "Adaptive edge analytics for distributed networked control of water systems," in *Proceedings of the 1st IEEE International Conference on Internet-of-Things Design and Implementation, IoTDI 2016*, pp. 72–82, April 2016.

[7] J. Lloret, J. Tomas, A. Canovas, and L. Parra, "An Integrated IoT Architecture for Smart Metering," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 50–57, 2016.

[8] R. Du, L. Gkatzikis, C. Fischione, and M. Xiao, "Energy Efficient Sensor Activation for Water Distribution Networks Based on Compressive Sensing," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2997–3010, 2015.

[9] O. Jo, Y.-K. Kim, and J. Kim, "Internet of Things for Smart Railway: Feasibility and Applications," *IEEE Internet of Things Journal*, 2017.

[10] A. Freitas, L. Brito, K. Baras, and J. Silva, "Smart mobility: A survey," in *Proceedings of the 2017 Internet of Things for the Global Community, IoTGC 2017*, July 2017.

[11] I. Jawhar, N. Mohamed, and D. P. Agrawal, "Linear wireless sensor networks: classification and applications," *Journal of Network and Computer Applications*, vol. 34, no. 5, pp. 1671–1682, 2011.

[12] S. Varshney, C. Kumar, and A. Swaroop, "Linear sensor networks: Applications, issues and major research trends," in *Proceedings of the 2015 International Conference on Computing, Communication and Automation, ICCCA 2015*, pp. 446–451, May 2015.

[13] P. Huang, L. Xiao, S. Soltani et al., "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[14] R. C. Carrano, D. Passos, L. C. S. Magalhaes, and C. V. N. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 181–194, 2014.

[15] J. J. Garcia-Luna-Aceves and R. Menchaca-Mendez, "STORM: A framework for integrated routing, scheduling, and traffic management in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 8, pp. 1345–1357, 2012.

[16] O. Yang and W. Heinzelman, "Modeling and performance analysis for duty-cycled MAC protocols with applications to S-MAC and X-MAC," *IEEE Transactions on Mobile Computing*, vol. 11, no. 6, pp. 905–921, 2012.

[17] L. Guntupalli and F. Y. Li, "DTMC modeling for performance evaluation of DW-MAC in wireless sensor networks," in *Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, WCNC 2016*, April 2016.

[18] L. Guntupalli, J. Martinez-Bauset, F. Y. Li, and M. A. Weitnauer, "Aggregated packet transmission in duty-cycled WSNs: Modeling and performance evaluation," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 563–579, 2017.

[19] F. Tong, L. Zheng, M. Ahmadi, M. Ni, and J. Pan, "Modeling and analyzing duty-cycling pipelined-scheduling MAC for linear sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2608–2620, 2016.

[20] H. S. Khanh, C.-Y. Ock, and M. K. Kim, "RP-MAC: A cross-layer duty cycle MAC protocol with a Reduced Pipelined-forwarding feature for Wireless Sensor Networks," in *Proceedings of the 11th International Wireless Communications and Mobile Computing Conference, IWCMC 2015*, pp. 1469–1474, August 2015.

[21] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[22] R. Singh and S. Chouhan, "A cross-layer MAC protocol for contention reduction and pipelined flow optimization in wireless sensor networks," in *Proceedings of the IEEE 2nd International Conference on Recent Trends in Information Systems, ReTIS 2015*, pp. 58–63, July 2015.

[23] R. Singh, B. K. Rai, and S. K. Bose, "A Joint Routing and MAC Protocol for Transmission Delay Reduction in Many-to-One Communication Paradigm for Wireless Sensor Networks," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1031–1045, 2017.

[24] S. Khaliq and S. Henna, "HE-PRMAC: Hop extended pipelined routing enhanced MAC protocol for wireless sensor networks," in *Proceedings of the 6th International Conference on Innovative Computing Technology, INTECH 2016*, pp. 392–397, August 2016.

[25] F. Tong, M. Ni, L. Shu, and J. Pan, "A Pipelined-forwarding, Routing-integrated and effectively-Identifying MAC for large-scale WSN," in *Proceedings of the 2013 IEEE Global Communications Conference, GLOBECOM 2013*, pp. 225–230, December 2013.

[26] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling*, Princeton University Press, Princeton, NJ, USA, 1st edition, 2009.

[27] J. F. Traub, *Iterative Methods for the Solution of Equations*, American Mathematical Society, 2nd edition, 1982.