Received February 16, 2022, accepted March 23, 2022, date of publication March 31, 2022, date of current version April 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3163728

A Scalable and Energy-Efficient MAC Protocol for Linear Sensor Networks

ICLIA VILLORDO-JIMENEZ[1], NOÉ TORRES-CRUZ[1], ROLANDO MENCHACA-MENDEZ[2], AND MARIO E. RIVERO-ANGELES[2]

[1]UPIITA, Instituto Politécnico Nacional, Mexico City 07340, Mexico
[2]CIC, Instituto Politécnico Nacional, Mexico City 07738, Mexico

Corresponding author: Iclia Villordo-Jimenez (ivillordo@ipn.mx)

This work was supported in part by the Secretaría de Investigación y Posgrado del Instituto Politécnico Nacional (SIP-IPN) under Project 20221996.

ABSTRACT Linear topologies arise naturally in the context of Internet-of-Things (IoT) applications for smart cities, where the infrastructure itself commonly has a linear or semi-linear structure. This is the case of buildings, public transportation systems, road infrastructure, and utility distribution networks. Given the prevalence of this type of topologies, several Medium Access Control (MAC) protocols have been designed to take advantage of their particular properties. Unfortunately, most of them do not scale well as the node density and the distance in hops to the sink increases. The result is that packets generated many hops away from the sink tend to experience unacceptable high end-to-end delay and low delivery probabilities. This paper introduces HP-MAC, a synchronized duty-cycled MAC protocol for Linear Sensor Networks (LSNs) that assigns transmission priorities to nodes to avoid collisions, through the implementation of distributed elections based on hash functions. HP-MAC also implements a packet queuing scheme that acts as a mechanism to control the amount of network resources allocated to data flows generated at different distances to the sink. This way, packets can reach their destination with loss probability and end-to-end delay that do not depend on their distance to the sink. We use a Discrete-Time Markov Chain (DTMC) to model the performance of the proposed protocol. Numerical solutions of this model show that HP-MAC outperforms state-of-the-art representatives in terms of throughput, end-to-end delay, power consumption, and packet loss probability. These results are validated through extensive discrete-event simulations.

INDEX TERMS Linear sensor networks (LSN), synchronized duty-cycled MAC protocol, collision-free MAC protocol, Markov chain, energy-efficient Internet of Things (IoT).

I. INTRODUCTION

The development of smart-city applications requires an infrastructure that physically and virtually interacts with various devices such as sensors and actuators [1]–[3]. In this regard, the Internet of Things (IoT) has become a fundamental technology because, through the use of wireless links, IoT devices can communicate with each other and with the Cloud to implement coordinated operations that provide advanced monitoring and control services [4], [5].

Given the particular characteristics of the urban infrastructure, smart-city applications often require the deployment of Wireless Sensor Networks (WSNs) along linear or semi-linear structures such as utility distribution networks, road networks, public transportation systems, and buildings. For instance, in the case of applications for smart mobility and intelligent transportation systems, the road infrastructure is sensed to coordinate urban transport or to manage vehicular traffic [6]. In intelligent utility distribution networks, the infrastructure is sensed to detect leaks or measure user consumption [1]–[3], [7]–[9]. In these scenarios, where the coverage area or the infrastructure itself is linear, the WSNs deployed along with these structures also exhibit a linear topology. These Linear Sensor Networks (LSN) collect information that is transported hop-by-hop to sink nodes that are typically located at one end of the network [10]. Although some variables in these scenarios could be monitored with mobile nodes, there are a lot of phenomena where fixed nodes are an excellent option to this end, hence, this work is focused on LSN's with static nodes.

The associate editor coordinating the review of this manuscript and approving it for publication was Rashid Mehmood.

As is usually the case in WSNs, energy-efficiency is an important design aspect in LSN because it may impact the network lifetime [10], particularly if sensor nodes do not have access to a continuous power supply [9]. Moreover, several authors have identified that MAC protocols play a significant role in this regard [3], [11]–[13]. The designers of LSN have taken advantage of the linear topology of these networks to propose energy-efficient MAC protocols based on Duty-Cycled, Pipelined-Forwarding (DCPF). In DCPF protocols, nodes are organized into *grades* depending on their distance in hops to the sink. Nodes in the same grade share the same duty cycle, and the pipeline is defined so that nodes in a given grade are in receiving state only when the nodes in the next grade are in transmitting state (see Figure 2). Energy-efficiency comes as a result of reducing idle listening and overhearing periods attained by this synchronized duty cycle [16], [18]–[22]. Unfortunately, as the Related Work Section shows, most previous proposals rely on traditional contention mechanisms that tend to create bottlenecks near the sink, limiting network throughput. Moreover, most MAC protocols for LSNs proposed up to date do not scale well as the distance in hops to the sink increases. Packets generated many hops away from the sink tend to experience unacceptable high end-to-end delays and low delivery probabilities.

As a solution to the contention problem, the authors of [22], [24], [25], [28] have proposed protocols based on hash functions to provide collision-free communications. Even though these collision-free proposals effectively increase the network throughput, they still suffer from reduced performance as the distance in hops to the sink increases. This problem arises because packets from remote nodes are queued in many buffers along the network, competing for transmission at each hop against locally generated packets. As a result, these packets exhibit unacceptable loss probability and end-to-end delay levels. To address this problem, some authors (e.g., [29], [30]) have proposed prioritizing packets based on the number of hops required to reach the sink or the packet's deadline. Other authors take the more traditional approach of prioritizing packets by the type of service (e.g., real-time versus elastic) [31]–[33] and do not consider the distance that the packets have to traverse to reach the sink.

This paper presents the Hash-based with Packet prioritization MAC protocol (HP-MAC), a synchronized DCPF protocol particularly suited for Linear Sensor Networks (LSNs). HP-MAC implements distributed elections based on hash functions that assign transmission priorities to nodes in one grade to avoid collisions. HP-MAC also implements a packet queuing scheme that acts as a mechanism to control the amount of network resources assigned to data flows from different grades. This way, even packets generated many hops away from the sink experience high delivery probability and reduced end-to-end delay. Moreover, HP-MAC achieves these features without increasing the signaling overhead or requiring additional synchronization. An analytical model based on a Discrete-Time Markov Chain (DTMC) shows that these characteristics of the proposed protocol increase the LSN capacity while reducing energy consumption and end-to-end delay. All the results provided by our analytical method were validated through extensive discrete-event simulations of the system.

The rest of the paper is organized as follows. In Section II we review a representative sample of related work. Then, in Section III, we describe how our protocol works. Based on this description, in Section IV we present the DTMC that models the system. In Section V we derive expressions for the LSN performance parameters. Lastly, we discuss relevant numerical results and present our conclusions in Sections VII and VIII, respectively.

II. RELATED WORK

In [14], W. Ye, *et al.* represented S-MAC, a Synchronized Duty-Cycled (SDC) MAC protocol for WSN's that has become highly influential in the design of subsequent proposals because of its energy efficiency and simplicity. In addition, Yang, *et al.* developed a Markov queueing model to analyze SDC-MAC protocols with applications to S-MAC [15]. Based on these works, Tong, *et al.* introduced the Pipelined Forwarding (PF) feature to SDC protocols by designing a protocol known as DCPF, particularly suited for LSN's [16]. These authors evaluated the DCPF performance in terms of throughput, active time ratio per cycle, and packet delivery latency through a Markov chain model.

Many SDC-MAC protocols for LSNs have been proposed in recent years (e.g., [17]–[21], [23]). These proposals take advantage of the assumption of having a linear or semi-linear topology to provide energy-efficient communications. Unfortunately, most of these protocols still rely on traditional contention mechanisms, which makes them suffer from reduced network capacity and high packet loss probabilities due to packet collisions, particularly under high-node density conditions. Another common property of these proposals is that they usually do not include mechanisms to prioritize packets from remote nodes causing them to suffer from excessive end-to-end delay.

To ameliorate the negative impact of contention, some authors have proposed collision-free MAC protocols [22], [24]–[28]. In particular, those based on hash functions have proved to be very effective. Some other authors have proposed mechanisms to reduce the end-to-end delay of specific packets by using multiple techniques to differentiate priorities [29]–[36]. However, except for [22], none of these proposals were designed for LSN.

We provide brief reviews of the works mentioned above in the remainder of this section.

A. MAC PROTOCOLS FOR LSN's

Wang *et al.* [17] introduced a Utility-based Adaptive Duty-cycle (UADC) algorithm that increases energy efficiency and reduces transmission delay, in comparison with previous works. This algorithm selects the relay node that maximizes a utility function. This function synthesizes comprehensive metrics like reliability, energy consumption, and delay of a node; hence, the network performance can be improved on the premise of not harming its lifetime.

In [18], F. Tong *et al.* presented an interference analysis of the DCPF protocol (presented in [16]) that provides the distribution of the Signal-to-Interference-plus-Noise Ratio (SINR), as well as the network capacity attained by the protocol.

In [19], Zhang *et al.* presented the Redundancy DCPF (RDCPF) protocol. This proposal incorporates redundant nodes to reduce the traffic bottlenecks and the excessive energy consumption experienced by nodes located near the sink. Their results show that RDCPF outperforms the existing DCPF protocols in terms of packet delivery latency, average power consumption, network survival time, and energy usage efficiency.

SA-MAC is another DCPF-based protocol [21] where nodes selectively awake depending on the network node density, traffic load conditions, and the state of the nodes' buffers. An analytical model shows that SA-MAC significantly reduces the nodes' energy consumption compared to previous DCPF protocols, particularly under high traffic loads.

In [23], Acosta *et al.* also analyze a WSN with linear topology. They propose an algorithm based on implicit ACK messages. A transmitting node acknowledges that its packet has been received when it listens that the intended receiver transmits the same packet towards the sink. According to the authors, this scheme achieves better energy consumption and end-to-end delay than protocols with explicit confirmations.

B. COLLISION-FREE MAC PROTOCOLS

In [24], Rajendran *et al.* propose TRAMA, a scheme composed of an access protocol, a schedule exchange protocol, and an adaptive election algorithm. To avoid collisions, TRAMA implements distributed elections where each node determines transmission priorities for the nodes contending for the channel. TRAMA achieves higher throughput than contention-based protocols, it is well suited for applications that require high delivery guarantees and energy efficiency, but it is limited in delay-sensitive scenarios.

In [25], Camacho-Escoto *et al.* proposed TEMA, a MAC protocol where, similarly to TRAMA, nodes use a dynamic set of tickets to contend for interference-free access to the channel. TEMA achieves energy efficiency by opportunistically setting the nodes' radios in sleeping mode when nodes lose the election and infer that they are not the intended receivers of the winner. Similar to TRAMA, TEMA does not provide a mechanism to control the end-to-end experienced by packets generated by sources located many hops away from their corresponding destinations.

Villordo-Jimenez *et al.* [22] presented H-MAC, a MAC protocol specifically designed for LSN that combines the advantages of pipelined scheduling and collision-free channel access based on hash functions. Numerical results show that H-MAC outperforms previous proposals for LSN's in terms of throughput, energy consumption, and delay; however, these results were obtained only through simulations, and no mathematical analysis is presented.

The protocols presented in [26], [27] are examples of recent collision-free MAC protocols that are not based on hash functions. In [26] Alfouzan *et al.* propose GC-MAC, a collision-free reservation-based MAC protocol for underwater WSN. GC-MAC is based on TDMA and assigns separate time-slots to every sensor node in every two-hop neighborhood. This way, GC-MAC avoids collisions and increases the throughput and energy efficiency compared to contention-based protocols. On the other hand, in [27], Movva and Rao propose a ring-partitioned-based MAC (RP-MAC) protocol to implement an energy-efficient WSN with a mobile sink node. RP-MAC schedules transmissions among adjacent nodes, achieving collision-free data transmission in the network. Like most of the works summarized in this sub-section, GC-MAC and RP-MAC were not designed to operate in LSN.

C. MAC PROTOCOLS WITH PACKET PRIORITIZATION

Recent proposals have addressed the problem of reducing the end-to-end delay experienced by nodes located many hops away from the sink by prioritizing particular packets. In [29], Farhan *et al.* propose a scheduling algorithm called Long Hop (LH), where packets from remote nodes are prioritized in a queue. Since this strategy reduces remote retransmissions, the overall system performance is improved. Argoubi *et al.* address a similar problem in [30], but they take into consideration the packet deadline (which depends on the kind of service) to determine their priority in a single queue.

Other proposals employ multiple priority queues (e.g., [31]–[33]); however, the main objective of them is to provide different classes of services and do not consider the distance (in hops) between the source and the sink.

Lastly, some recent works have proposed various techniques to prioritize packets in terms of QoS. In [34], Sakib *et al.* propose a Timeout Multi-priority based MAC (TMPQ-MAC) protocol that, based on the load variations in the network, dynamically determine the nodes' active time. TMPQ-MAC considers four different packet priorities to reduce delay. In [35], Muzakkari, *et al.* present an energy-efficient and QoS-aware (EEQ) MAC protocol that has an adaptive duty cycle in terms of queue size and packet priority. The authors show that they can improve energy efficiency and extend the lifespan of the WSN. In [36], Thu-Hang, *et al.* also propose a protocol with differentiated QoS which is based on Carrier Sense Multiple Access p-persistent MAC protocol. Specifically, they propose using different p values for different priority levels. Higher priority packets are more likely to access the shared medium and experience better QoS in terms of delay and reliability.

Unlike these works, we propose using multiple queues to classify packets in terms of their grade of origin. Our strategy is particularly suited for LSN, and our goal is to reduce the delay of packets generated at remote nodes by assigning higher priorities to their queues.

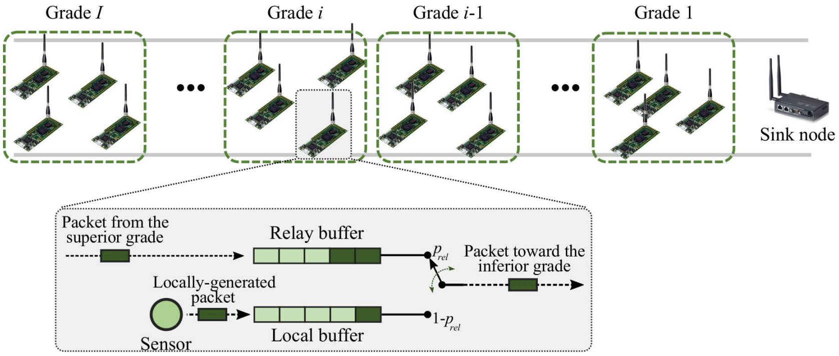

FIGURE 1. LSN architecture and packet buffering at nodes.

III. HP-MAC PROTOCOL

In this section, we present the characteristics of the LSN under consideration, as well as an overview of the general functioning of the Hash-based with Packet prioritization MAC protocol (HP-MAC) (Section III-A). Then, in Section III-B we describe the use of a strongly 2-universal family of hash functions to avoid collisions among nodes contending for accessing the channel.

A. PROTOCOL OVERVIEW AND CHANNEL STRUCTURE

We assume an LSN with static nodes that are classified into *grades* according to their corresponding distance in hops to the sink node. Accordingly, we define the *grade* i as the set of nodes located i hops away from the sink. (see Fig. 1). Similar to previous proposals [16], [21], we assume that nodes acquire information about the constituency of their corresponding grade during an initialization stage. Nodes in the same grade share the same transmitting slot, and hence, they have to contend for the channel.

We assume that all the nodes in the network (except the sink) can locally generate sensory packets and relay packets received from their immediate superior grade.

In the LNS, data packets are disseminated hop-by-hop from nodes at consecutive grades until they reach the sink node. More specifically, packets generated by a node at grade i are forwarded to a node at grade $i-1$, then to a node in grade $i-2$ until reaching the sink node. We consider that nodes identify intended receivers in the following grade towards the sink during the initialization stage, hence, we consider unicast transmissions.

As shown in Fig. 1, HP-MAC nodes are equipped with two independent data queues, a *local buffer* that is used for queuing locally-generated sensory data and a *relay buffer* that is used for queuing data packets received from nodes in superior grades. These queues are First-Come First-Served (FCFS), and their length equals K packets.Note that even though a node at grade i only receives packets from nodes in the immediate superior grade $(i+1)$, as a result of the hop-by-hop transmissions, their relay buffers can store packets generated at grades larger than $(i+1)$.

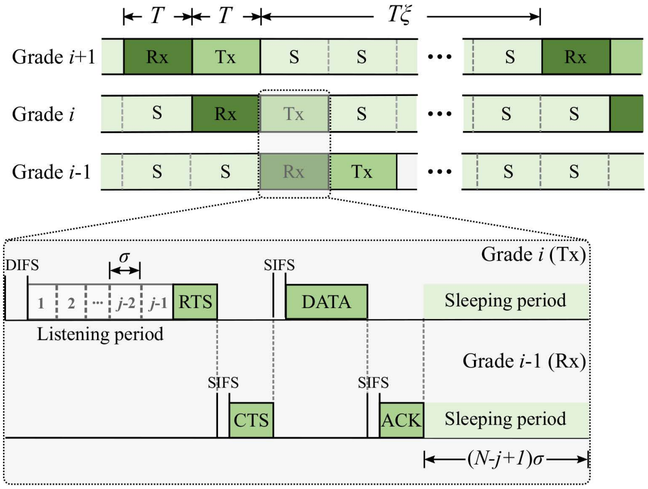

FIGURE 2. Frame and slot structure.

When a node has packets in both buffers, it selects for transmission a packet from the relay buffer with probability p_{rel} or from the local buffer with probability $p_{loc} = 1 - p_{rel}$. The value of p_{rel} is chosen to control the priority that relayed packets have over locally-generated ones. If only one buffer has packets, it will be automatically served. To save energy, only nodes with packets-to-send awake during transmitting slots, and nodes with saturated relay buffers do not awake during receiving slots.

We assume that nodes have a single radio that can either be in a sleeping state, a transmitting state, or a receiving state. We are considering a synchronized duty-cycled protocol where nodes share a single wireless channel that is time-slotted and organized into a frame consisting of a fixed number of time slots. As we show in Fig. 2, this frame is composed of a single receiving slot, a single transmitting slot, and a sequence of ξ consecutive sleeping slots.

As we also show in Fig. 2, the frame for nodes in grade i is delayed by exactly one slot in relation to the frame of nodes in grade $i+1$. This way, whenever the nodes in grade i are in a transmitting slot, the nodes in the grade $i-1$ (their potential intended receivers) are in a receiving slot, and, as a result, the delay experienced by a packet traversing from one grade to the next may be as short as one slot. This technique is known as pipeline scheduling [16].

According to the previous information, the cycle duration T_c, can be expressed as

$$T_c = (\xi + 2)T, \tag{1}$$

where T is the slot duration and ξ must be larger than 1 to avoid interference between adjacent grades. Furthermore, notice that the maximum network throughput is $\frac{1}{T_c}$ packets/s, since only one packet can be transmitted between two adjacent grades during one slot.

As also shown in Fig. 2, a slot initiates with a DCF interframe space (DIFS) followed by N minislots of σ seconds, where N is the number of nodes per grade. As explained in Section III-B, these minislots are used by the protocol to identify the winner of the distributed election, namely, the node that will send data during the current slot. Then, the sender (a node in grade i) and its intended receiver (a node in grade $i-1$) engage in an RTS-CTS- DATA-ACK exchange. The proposed slot structure includes three Short Interframe Spaces (SIFSs) because both nodes require a SIFS to process their received messages before sending their corresponding replies.

According to the previous description, the slot duration can be expressed as

$$T = \tau_{msg} + \sigma N; \tag{2}$$

where τ_{msg} is given by

$$\tau_{msg} = \tau_{difs} + \tau_{rts} + \tau_{cts} + \tau_{data} + \tau_{ack} + 3\tau_{sifs}, \tag{3}$$

and τ_{difs}, τ_{rts}, τ_{cts}, τ_{data}, τ_{ack} and τ_{sifs} are the duration of the messages and time intervals to implement the handshake and the data transmission.

In addition to the initialization stage, the sink node must periodically transmit special frames to maintain synchronization, detect node failures, and re-establish sender-receiver links (in case of node failure).[1] We point the reader to [14], and [20] for detailed descriptions of how to implement some of these functions. However, since the sink node sends these special frames in a time scale much larger than the frame duration, their effects are omitted in the proposed analytical model.

B. HASH-BASED MAC PROTOCOL

In HP-MAC, accessing the channel during transmission slots is based on a distributed election. At the beginning of the slot, every awake node in grade i generates a set of N priority tickets, i.e., one ticket for every node in its grade. The value of a given ticket represents that node's priority for accessing the channel. The node with the largest ticket value has the highest priority to transmit.

The value of every ticket is calculated using a hash function from a strongly 2-universal family, which is given by

$$h_n(k) = (a_n k + b_n) \mod p, \tag{4}$$

where $k \in [0, N-1]$ is a unique node identifier, p is a prime number that satisfies $p \geq N$, and $a_n, b_n \in [0, p-1]$ are pseudo-random numbers drawn from a uniform distribution [37]. Nodes use a monotonically increasing transmission slot number as the seed for generating the values of a_n and b_n at each transmission slot. This way, the priority computed by (4) also changes on a per transmission slot basis. Since all the nodes know the current slot number, they compute the same set of tickets in a specific slot, without the need for additional signaling in comparison to traditional contention-based protocols.

Note that the slot-number counter can wrap around, hence, the nodes cyclically generate the same sequence of priority tickets; however, this do not interfer with the correct network operation because the typical network lifetime is much smaller than the wraparound time. For instance, with a 32-bit counter and the values used in the numerical evaluation, the wraparound time is in the order of decades.

Before accessing the channel, the node with the j-th priority listens to the channel for $j-1$ mini-slots. If, during that time, it receives an RTS message from a node with higher priority, it goes to the sleeping state (see Fig. 2). Otherwise, the node transmits its own RTS message to indicate to the remaining nodes that it has a packet to send. Then, this node and its intended receiver engage in a CTS-DATA-ACK exchange. After that, sender and receiver go to the sleeping state for the remaining duration of the slot. The length of this latter sleeping period varies because the length of the listening period also varies.

Despite the simplicity of (4), it has the following desirable properties [25], [38], [39]:

- There are no collisions because the input of the hash function depends on unique node identifiers and the current transmission slot number.
- Since the hash function is a dispersion hash function, it generates any permutation of the tickets with the same probability.

As a result of the first property, our protocol is collision-free without requiring additional signaling or synchronization. From the second property, all the participant nodes have the same probability of obtaining the highest (or any other specific) priority at each election.

IV. ANALYTICAL MODEL

In this section, we derive expressions for computing the probability of transmitting and receiving a packet in one cycle (Section IV-A), then, from these results, we propose a Markov chain to represent the state of the two buffers of a node (Section IV-B). Table 1 summarizes the notation used throughout the analysis.

[1] As a result of node failure, some packets may be lost and mechanisms to recover them may be implemented. We believe, however, that these mechanisms can be implemented and analyzed from the perspective of upper layers, whereas, we focus this work on the MAC sub-layer.

TABLE 1. Summary of notation.

Symbol	Variable description
a	Probability that a node generates a data packet during a cycle
D_i	Expected end-to-end delay for a packet generated at grade i
$h_n(k)$	Priority ticket for node k during cycle n
I	Number of grades in the LSN
K	Buffer length
N	Number of nodes in each grade
$p_{e,e}(i)$	Probability that both queues are empty
p_{loc}	Probability of selecting for transmission a packet from the local buffer
p_{lp}	Packet loss probability
$P_{(n,v)}^{(m,u)}$	DTMC transition probability from state (m, u) to (n, v)
$p_r(i)$	Packet reception probability
$p_t(i)$	Packet transmission probability
p_{rel}	Probability of selecting for transmission a packet from the relay buffer
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
$P_{tot}(i)$	Expected power consumption for a node at grade i
S	Network throughput
T	Slot duration
T_c	Cycle duration
ξ	Number of slots that a node spends in sleeping mode per cycle
$\pi_{m,u}^i$	DTMC steady-state probabilities
σ	Duration of one minislot
τ_{difs}	DCF interframe space duration
τ_{rts}	RTS message duration
τ_{cts}	CTS message duration
τ_{data}	Data message duration
τ_{ack}	ACK message duration
τ_{sifs}	Short interframe space duration

A. TRANSMISSION AND RECEPTION PROBABILITIES

We first derive expressions for computing the probabilities of transmitting a packet from grade i and receiving a packet at grade i during a given cycle, which are denoted by $p_t(i)$ and $p_r(i)$, respectively. For simplicity, these probabilities are computed in terms of the probability $p_{e,e}(i)$ that both buffers of a particular node are empty at the beginning of a transmitting slot. As we explain later, by defining and solving a DTMC, we can compute $p_{e,e}(i)$.

Please remember that an awake node with the jth priority initiates its transmission only if the $j-1$ nodes with higher priorities have no packets in their buffers. In addition, note that the probability of obtaining the j-th position in the election is $\frac{1}{N}$. As a result, the probability of transmitting, given that the j-th position was obtained, is $\frac{1}{N}p_{e,e}(i)^{j-1}$. By considering all the possible values of j, we obtain

$$p_t(i) = \frac{1}{N}\sum_{j=1}^{N} p_{e,e}(i)^{j-1} = \frac{1-p_{e,e}(i)^N}{N\left(1-p_{e,e}(i)\right)}, \tag{5}$$

where $p_t(i)$ is conditioned on the existence of packets in any of the two node's buffers.

Now, since the proposed contention mechanism is collision-free, and considering unicast transmissions, a node in grade i receives a packet with probability $p_r(i)$, given by

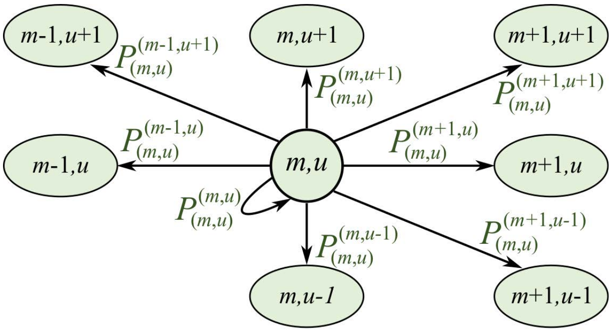

FIGURE 3. DTMC.

$$p_r(i) = \begin{cases} p_t(i+1)\left[1-p_{e,e}(i+1)\right]; & i < I \\ 0; & i = I \end{cases} \tag{6}$$

We should notice that $p_r(i)$ does not model the buffer saturation in the receiving node.

In addition, the probability $p_b(i)$ of detecting a busy channel for an awake node is simply the complement of $p_t(i)$, hence,

$$p_b(i) = 1 - p_t(i). \tag{7}$$

B. MARKOV CHAIN MODEL

Since (5)-(7) are defined in terms of $p_{e,e}(i)$, in this Section we obtain this probability by modeling the node's buffers through a Discrete-Time Markov Chain (DTMC).

Let m and u be the number of packets at the beginning of the transmitting slot, in the relay and local buffers, respectively. We consider that the maximum size of every buffer is K, therefore, $m, u \in [0 : K]$. The chain states are represented by the duple (m, u), and the transition probability from state (m, u) to state (n, v) is denoted by $P_{(m,u)}^{(n,v)}$. Notice that this chain has to be solved for every grade i; however, we omitted i in the transition probabilities to simplify the notation. To define the possible transitions, we observe that:

- n can not exceed m for more than a unit because only one packet can be received per cycle. The same is true for v and u, since only one packet can be locally generated during one cycle.
- If neither a transmission, a reception nor a new-packet generation occurs during a cycle, then $n = m$ and $v = u$.
- If $n = m - 1$, then $v = u$, because only one packet can be transmitted per cycle. Analogously, if $v = u - 1$, then $n = m$.

Additionally, we consider that a node locally generates a packet during a cycle with probability a. Considering this, we illustrate all possible transitions in Fig. 3, and define their probabilities in the following equations (8-49), where $p_t(i)$ and $p_r(i)$ are abbreviated as p_t and p_r, respectively:

$$P_{(0,0)}^{(0,0)} = (1-p_r)(1-a); \tag{8}$$

$$P_{(0,0)}^{(1,1)} = p_r a; \tag{9}$$

$$P_{(0,0)}^{(1,0)} = p_r(1-a); \tag{10}$$

$$P_{(0,0)}^{(0,1)} = (1-p_r)a; \tag{11}$$

$$P_{(m,0)}^{(m,0)} = p_r(1-a)p_t + (1-p_r)(1-a)(1-p_t); \quad m \in [1:K-1] \tag{12}$$

$$P_{(m,0)}^{(m+1,0)} = p_r(1-a)(1-p_t); \quad m \in [1:K-1] \tag{13}$$

$$P_{(0,u)}^{(0,u)} = (1-p_r)ap_t + (1-p_r)(1-a)(1-p_t); \quad u \in [1:K-1] \tag{14}$$

$$P_{(0,u)}^{(0,u+1)} = (1-p_r)a(1-p_t); \quad u \in [1:K-1] \tag{15}$$

$$P_{(m,0)}^{(m,1)} = (1-p_r)a(1-p_t) + p_r a p_t; \quad m \in [1:K-1] \tag{16}$$

$$P_{(0,u)}^{(0,u-1)} = (1-p_r)(1-a)p_t; \quad u \in [1:K-1] \tag{17}$$

$$P_{(0,u)}^{(1,u-1)} = p_r(1-a)p_t; \quad u \in [1:K-1] \tag{18}$$

$$P_{(m,0)}^{(m-1,1)} = (1-p_r)ap_t; \quad m \in [1:K-1] \tag{19}$$

$$P_{(0,u)}^{(1,u)} = p_r(1-a)(1-p_t) + p_r a p_t; \quad u \in [1:K-1] \tag{20}$$

$$P_{(m,0)}^{(m-1,0)} = (1-p_r)(1-a)p_t; m \in [1:K-1] \tag{21}$$

$$P_{(m,0)}^{(m+1,1)} = p_r a(1-p_t); m \in [1:K-1] \tag{22}$$

$$P_{(m,u)}^{(m,u)} = (1-p_r)(1-a)(1-p_t) + p_r(1-a)p_t p_{rel} + (1-p_r)ap_t p_{loc}; m,u \in [1:K-1] \tag{23}$$

$$P_{(m,u)}^{(m,u+1)} = (1-p_r)a(1-p_t) + p_r a p_t p_{rel}; \quad m,u \in [1:K-1] \tag{24}$$

$$P_{(m,u)}^{(m,u-1)} = (1-p_r)(1-a)p_t p_{loc}; \quad m,u \in [1:K-1] \tag{25}$$

$$P_{(m,u)}^{(m+1,u+1)} = p_r a(1-p_t); m,u \in [1:K-1] \tag{26}$$

$$P_{(m,u)}^{(m+1,u-1)} = p_r(1-a)p_t p_{loc}; \quad m,u \in [1:K-1] \tag{27}$$

$$P_{(m,u)}^{(m-1,u+1)} = (1-p_r)ap_t p_{rel}; \quad m,u \in [1:K-1] \tag{28}$$

$$P_{(m,u)}^{(m+1,u)} = p_r(1-a)(1-p_t) + p_r a p_t p_{loc}; \quad m,u \in [1:K-1] \tag{29}$$

$$P_{(m,u)}^{(m-1,u)} = (1-p_r)(1-a)p_t p_{rel}; \quad m,u \in [1:K-1] \tag{30}$$

$$P_{(m,K)}^{(m,K)} = (1-p_r)(1-p_t) + p_r p_t p_{rel}; \quad m \in [1:K-1] \tag{31}$$

$$P_{(m,K)}^{(m+1,K)} = p_r(1-p_t); \quad m \in [0:K-1] \tag{32}$$

$$P_{(K,u)}^{(K,u)} = (1-a)(1-p_t) + ap_t p_{loc}; \quad u \in [1:K-1] \tag{33}$$

$$P_{(K,u)}^{(K,u+1)} = a(1-p_t); u \in [0:K-1] \tag{34}$$

$$P_{(m,K)}^{(m-1,K)} = (1-p_r)p_t p_{rel}; \quad m \in [1:K-1] \tag{35}$$

$$P_{(m,K)}^{(m,K-1)} = (1-p_r)p_t p_{loc}; \quad m \in [1:K-1] \tag{36}$$

$$P_{(m,K)}^{(m+1,K-1)} = p_r p_t p_{loc}; \quad m \in [1:K-1] \tag{37}$$

$$P_{(K,u)}^{(K-1,u)} = (1-a)p_t p_{rel}; \quad u \in [1:K-1] \tag{38}$$

$$P_{(K,u)}^{(K-1,u+1)} = ap_t p_{rel}; \quad u \in [1:K-1] \tag{39}$$

$$P_{(K,u)}^{(K,u-1)} = (1-a)p_t p_{loc}; \quad u \in [1:K-1] \tag{40}$$

$$P_{(K,0)}^{(K-1,0)} = (1-a)p_t; \tag{41}$$

$$P_{(K,0)}^{(K-1,1)} = ap_t; \tag{42}$$

$$P_{(K,0)}^{(K,0)} = (1-a)(1-p_t); \tag{43}$$

$$P_{(0,K)}^{(0,K)} = (1-p_r)(1-p_t); \tag{44}$$

$$P_{(0,K))}^{(0,K-1} = (1-p_r)p_t; \tag{45}$$

$$P_{(0,K)}^{(1,K-1)} = p_r p_t; \tag{46}$$

$$P_{(K,K)}^{(K-1,K)} = p_t p_{rel}; \tag{47}$$

$$P_{(K,K)}^{(K,K-1)} = p_t p_{loc}; \tag{48}$$

$$P_{(K,K)}^{(K,K)} = (1-p_t); \tag{49}$$

The steady-state probabilities of this chain are denoted by $\pi_{m,u}(i)$. Its solution is unique because the DTMC is irreducible and aperiodic. Note that according to (6), the solution for a given grade, except for $i = I$, depends on the solution of the immediate superior grade. Therefore, first, we solve for $i = I$, then we can solve for $i = I - 1$, and so on, until we find the solution for $i = 1$.

Now, we observe that he probability of having no packets to transmit $p_{e,e}(i) = \pi_{0,0}(i)$; therefore, $p_t(i)$ depends on the solution of (8-49), as it can be observed from (5). As a result of this relation between $p_{e,e}(i)$ and $p_t(i)$, the system of equations cannot be explicitly solved, and hence, we propose the following iterative numerical solution for each grade i [15], [16], [21]:

1) Calculate $p_t(i)$, according to (5), with an arbitrary initial value for $p_{e,e}(i)$.
2) Substitute $p_t(i)$ and $p_r(i)$ in (8)-(49) and solve the DTMC, for example, through the Gauss-Seidel method [40].
3) Make $p_{e,e}(i) \leftarrow \pi_{0,0}(i)$.
4) Repeat Steps 2 and 3 until $p_{e,e}(i)$ has converged according to certain criteria (e.g., mean squared error within a threshold value).
5) Compute $p_r(i-1)$ according to (6), because it will be required to compute the solution for the next grade.

As shown in the following section, we can obtain the network performance parameters by jointly solving (5)-(7) and (8)-(49).

To simplify the notation, we define $\pi_k^l(i)$ and $\pi_k^r(i)$ as the probabilities that a local and a relay buffer in grade i has k packets, respectively. These probabilities can be computed according to

$$\pi_k^l(i) = \sum_{m=0}^{N} \pi_{m,k}(i), \tag{50}$$

and

$$\pi_k^r(i) = \sum_{u=0}^{N} \pi_{k,u}(i). \tag{51}$$

V. PERFORMANCE ANALYSIS

In this section, we derive expressions for energy consumption (Sub-section V-A), throughput and packet-loss probability (Sub-section V-B) and, end-to-end delay (Sub-section V-C).

A. ENERGY CONSUMPTION

Let P_{tx}, P_{rx} and P_{sp} be the average power consumption for a node in transmission, reception and sleeping mode, respectively. These parameters are commonly provided by the hardware manufacturers. In addition, let $T_{tx}(i)$, $T_{rx}(i)$ and $T_{sp}(i)$ be the expected time that a node in grade i spends in transmission, reception and sleeping mode, respectively, during one cycle. Therefore, the expected power consumption for a node in grade i can be calculated as

$$P_{tot}(i) = \frac{1}{T_c}\left[P_{tx}T_{tx}(i) + P_{rx}T_{rx}(i) + P_{sp}T_{sp}(i)\right]. \quad (52)$$

First, to obtain $T_{tx}(i)$, we observe that in the transmitting slot, an awaken node listens to the channel until it detects the transmission of a higher-priority node or until it discovers that the channel is free for transmission; therefore, $T_{tx}(i)$ can be given by

$$T_{tx}(i) = \left[1 - p_{e,e}(i)\right]\left[p_b(i)\left(\sigma W_b(i) + \tau_{difs}\right) + p_t(i)\left(\sigma W_t(i) + \tau_{msg}\right)\right]; \quad (53)$$

where $W_b(i)$ and $W_t(i)$ represent the expected number of mini-slots that a node spends before detecting that the channel is busy, or initiating its own transmission, respectively (see Fig. 2).

Since the transmission probability for an awaken node with priority k is $\frac{p_{e,e}(i)^k}{Np_t(i)}$, $W_t(i)$ can be calculated as

$$W_t(i) = \sum_{k=0}^{N-1} k\frac{p_{e,e}(i)^k}{Np_t(i)}, \quad (54)$$

which can also be expressed as

$$W_t(i) = \frac{1}{Np_t(i)}\sum_{k=0}^{N-1} kp_{e,e}(i)^k \quad (55)$$

To obtain $W_b(i)$, we observe that an awake node listens to a transmission in the k-th mini-slot if: (1) there are at least other k nodes with higher priority, (2) these nodes, with priorities 1 to $k-1$ are sleeping, and (3) the node with the k-th priority has packets to transmit, hence,

$$W_b(i) = \frac{1}{p_b}\sum_{k=1}^{N-1} kp_{e,e}(i)^{k-1}\left(1 - p_{e,e}(i)\right)\frac{N-k}{N}, \quad (56)$$

and this expression can be rewritten as

$$W_b(i) = \frac{\left(1 - p_{e,e}(i)\right)}{Np_b(i)p_{e,e}(i)}\sum_{k=1}^{N-1} kp_{e,e}(i)^k\,(N-k). \quad (57)$$

On the other hand, to define $T_{rx}(i)$, we observe that a node awakes at the beginning of the reception slot only if its relay buffer is not saturated, which occurs with probability $1 - \sum_{j=0}^{K}\pi_{K,j}(i)$. If a transmission from the superior grade occurs, the nodes in the current grade are awake during the same time as their corresponding transmitting nodes, i.e., $p_r(i)\left(\sigma W_t(i+1) + \tau_{msg}\right)$; otherwise, they remain awake until the end of the N mini-slots (see Fig. 2). According to this, $T_{rx}(i)$ can be calculated as

$$T_{rx}(i) = \left[1 - \sum_{j=0}^{K}\pi_{K,j}(i)\right]\left[p_r(i)\left(\sigma W_t(i+1) + \tau_{msg}\right) + \left(1 - p_r(i)\right)\left(\sigma N + \tau_{difs} + \tau_{rts}\right)\right] \quad (58)$$

And the expected sleeping time per cycle can be calculated as

$$T_{sp}(i) = T_c - T_{tx} - T_{rx}. \quad (59)$$

By substituting (53), (58) and (59) in (52), we obtain the expected power consumption for a node in grade i. If we are interested in the expected power consumption per node in the LSN, we evaluate

$$P_{tot} = \frac{1}{I}\sum_{i=1}^{I} P_{tot}(i). \quad (60)$$

B. THROUGHPUT AND PACKET-LOSS PROBABILITY

Let S_i be the throughput in packets from grade i to grade $i-1$ per cycle. Then, the expected number of transmitted packets from grade i is given by $Np_t(i)\left[1 - p_{e,e}(i)\right]$, and nodes in grade $i-1$ can receive only if their relay buffers are not saturated, as a result, we obtain

$$S_i = \frac{1}{T_c}Np_t(i)\left[1 - p_{e,e}(i)\right]\left[1 - \sum_{u=0}^{K}\pi_{K,u}^{i-1}\right] \quad (61)$$

for $2 \le i \le I$.

Since we assume that the buffer in the sink node is infinite, for this special case the throughput is given by

$$S_1 = \frac{1}{T_c}Np_t(1)\left[1 - p_{e,e}(1)\right]; \quad (62)$$

which equals the network throughput, S.

Now, let $p_l(i)$ be the probability of losing a packet generated at grade i. This event occurs when the local buffer of the source node or a required relay buffer is saturated.

The rate at which packets are admitted in the local buffer is given by

$$r_q(i) = a\left[1 - \sum_{m=0}^{K}\pi_{m,K}^{i}\right], \quad (63)$$

and the probability of admitting a packet in all the relay buffers required to reach the sink is given by

$$p_{DS}(i) = \prod_{j=1}^{i-1}\left[1 - \sum_{u=0}^{K}\pi_{K,u}^{j}\right]. \quad (64)$$

As a result, the rate of packets generated at grade i that reach the sink is

$$r_s(i) = \frac{Na}{T_c}\left[1-\sum_{m=0}^{K}\pi^{i}_{m,K}\right]\prod_{j=1}^{i-1}\left[1-\sum_{u=0}^{K}\pi^{j}_{K,u}\right]. \tag{65}$$

Since new packets are generated at grade i at rate $\frac{Na}{Tc}$, $p_l(i)$ can be defined as

$$p_l(i) = 1 - \frac{T_c r_s(i)}{Na}. \tag{66}$$

Note that the previous analysis does not consider packet loss due to node failures. This effect can be incorporated into the model by introducing the probability of node failure in (61), (62) and (64).

C. END-TO-END DELAY

Let D_i be the expected value of the end-to-end delay experienced by a packet generated at grade i. D_i can be computed as the sum of the expected delays experienced at each buffer while traversing from source to sink; hence, it can be expressed as

$$D_i = D^l_i + \sum_{h=1}^{i-1} D^r_h; \tag{67}$$

where D^l_i and D^r_i are the expected delays in the local and relay buffers in grade i, respectively.

To calculate D^l_i and D^r_i, we use the solution of the Markov chain (see Section IV) and Little's Theorem. According to this theorem, the average queue length equals the product of the arrival rate and the average queueing delay. In our scenario, new packets arrive at the local buffer at rate λ^l_i packets/s, and this rate is given by

$$\lambda^l_i = \frac{a}{T_c}\left(1-\pi^l_K(i)\right). \tag{68}$$

We can obtain the expected value of the buffer's length from the probability distribution; however, in our analysis, we know this distribution only at specific points, namely, at the beginning of the transmission slots given by $\pi^l_k(i)$. Fortunately, between two of these points, the buffer length increases by one unit if a new packet arrives; decreases by one unit if a packet is transmitted; or does not change at all. Because exactly one arrival and one transmission occur per packet, we can determine D^l_i as follows.

First, we use Little's Theorem to obtain the expected buffer's length while assuming that such a length is constant during every cycle. We denote this length by L^*_i and can be expressed as

$$L^*_i = \sum_{k=0}^{K} k\pi^l_k(i), \tag{69}$$

then, by combining this equation with (68), we obtain the associated delay as,

$$D^*_i = \frac{T_c}{a\left(1-\pi^l_K(i)\right)}\sum_{k=0}^{K} k\pi^l_k(i). \tag{70}$$

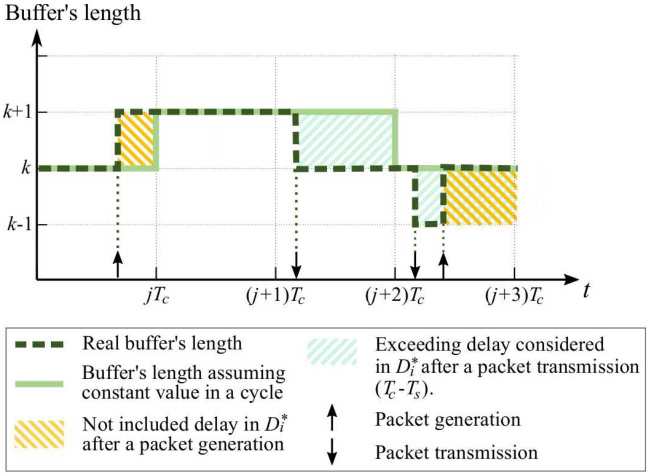
Buffer's length
k+1
k
k-1
jT_c
(j+1)T_c
(j+2)T_c
(j+3)T_c
t
Real buffer's length
Buffer's length assuming constant value in a cycle
Not included delay in D_i^* after a packet generation
Exceeding delay considered in D_i^* after a packet transmission (T_c-T_s).
Packet generation
Packet transmission

FIGURE 4. Differences between the exact delay and the approximation that considers constant buffer length during a cycle.

Clearly, D^*_i does not exactly represent the actual expected delay in a local buffer because it does not include the delay between the arrival of a packet and the beginning of the next cycle whose expected value is $\frac{T_c}{2}$ (see Fig. 4). Furthermore, D^*_i includes $T_c - T_s$ units of exceeding delay after a packet transmission, i. e., the time between the end of the transmission slot and the beginning of the next cycle. Then, by including these observations in (71), we obtain

$$D^l_i = \frac{T_c}{a\left(1-\pi^l_K(i)\right)}\sum_{k=0}^{K} k\pi^l_k(i) - \frac{T_c}{2} + T_s. \tag{71}$$

To calculate D^r_i, we follow an analogous method. In this case, however, the arrival rate is proportional to $p_r(i)$ instead of a, and there is no delay between the packet arrival and the beginning of the transmission slot. Therefore, we obtain

$$D^r_i = \frac{T_c}{p_r(i)\left(1-\pi^r_K(i)\right)}\sum_{k=0}^{K} k\pi^r_k(i) - T_c + T_s, \tag{72}$$

By substituting (71) and (72) in (67), we obtain the expected end-to-end packet delay.

VI. SIMULATION SETTINGS

We validated our analytical model through extensive discrete-event simulations in Matlab. These simulations were executed for 1×10^5 cycles for every set of input parameter values. In every cycle and node, we simulated the process of generating sensory packets and all components of the protocol frame, including idle times and protocol messages.

At the beginning of a transmission slot, the simulation generates priority tickets for every node in each grade (including those nodes with no packets to send). The node with the highest priority and packets-to-send (winning node) engages in a CTS-DATA-ACK exchange with its intended receiver while the remaining awake nodes go to sleep. If the winning node has packets in both buffers, it selects a packet with probability p_{rel} from the relay buffer, and with probability

TABLE 2. Numerical values for the network parameters.

Variable	Numerical value	Variable	Numerical value
I	7	K	7
P_{rx}	59.9mW	P_{sp}	0mW
P_{tx}	52.2mW	ξ	18
σ	1ms	τ_{difs}	10ms
τ_{sifs}	5ms	τ_{rts}	11ms
τ_{cts}	11ms	τ_{ack}	11ms
τ_{data}	43ms		

$1 - p_{rel}$ from the local buffer; otherwise, it simply transmits a packet from the non-empty buffer.

Once this process is over, we update the statistics related to the packet transmission. Simultaneously, we simulate the reception slot for the next grade. In this case, we identify which nodes have non-saturated queues, execute their corresponding reception events, and update the statistics about the packet reception. We simulate the transmission/reception process between every couple of adjacent nodes, including the process between grade 1 and the sink node.

In parallel with the transmission/reception process, we simulate the data packet arrival process at each node where nodes generate a new packet per cycle with probability a.

In all the transmission/reception events, we consider an ideal channel.

VII. RESULTS AND DISCUSSION

In this section, we present numerical results that characterize the performance of our proposal and that of two representatives of state-of-the-art DCPF protocols, namely, PRI-MAC [16] and SA-MAC [21]. To validate our analytical results, we also evaluate the system through discrete-event simulations as described in Section VI.

Table 2 presents the values of the network parameters. The value $I = 7$ was selected to show the non-homogeneous performance along the LSN topology while avoiding excessive computational load. In addition, the value $K = 7$ was selected to observe significant variations of both packet loss probability and end-to-end delay. For the remaining parameters, we use values that are commonly used in the literature [15], [16], [21], [22].

In our model, packet arrivals are characterized by the probability a of generating a packet during a cycle. From this probability, we can compute the arrival rate λ as $\lambda = aT_{cycle}$, which allows us to compare the performance exhibited by our proposal against that of PRI-MAC and SA-MAC.

To obtain the numerical results, we solved the DTMC for grade I (with $I = 7$) using the Gauss-Seidel method for a relative error of 1e-4. At this point, the reception probability $p_r(I)$ equals 0 because nodes at the last grade do not receive packets from other nodes. From this solution we can obtain the value of $p_r(I - 1)$, and then, solve the DTMC for grade $I - 1$. We continue with this strategy until solving the DTMC for grade 1.

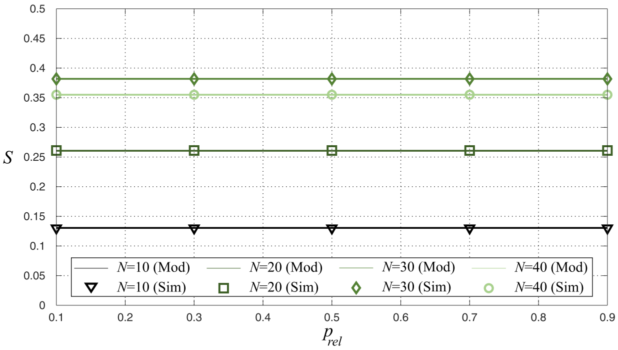

FIGURE 5. Throughput, in packets/s, as a function of the number of nodes in each grade (*N*) and the probability of choosing a packet from the relay buffer (p_{rel}), for $\lambda = 0.001875$.

Figs. 5-8 present the results provided by the analytical model when we evaluate the effect of the probability p_{rel} of selecting a packet from the relay buffer on the network performance. These figures also show the results of the simulation-based experiments for the same scenarios. As discussed in the following paragraphs, the results provided by these two methods are consistent.

From Fig. 5, we observe that p_{rel} does not affect the network throughput because this probability is not related to the network capacity or the packet arrival rate. Moreover, this is true for all the evaluated values of N. On the other hand, N significantly affects the throughput because the cycle duration depends on it; i.e., the largest N, the largest the cycle duration, and, as a result, the smallest maximum network capacity (e.g., for $N = 40$, $1/Tc = 0.3546$ packets/s). However, this is not the case for small values of N because the throughput is not limited by the network capacity but by the packet arrival rate. In this case, the analytical model and the simulation experiments yield practically the same results with deviations below 0.11%).

Fig. 6 shows that the expected value of the power consumption (P_{tot}) is practically insensitive to p_{rel}. This is because the time nodes spend awake (either receiving or transmitting) does not depend on p_{rel}. On the other hand, the number of nodes that awake at the beginning of the transmission slot directly depends on N, hence, the largest N, the largest P_{tot}, as we also observe in Fig. 6. Similar to the case of the throughput, our analytical model for P_{tot} is consistent with the simulation results. In this case, the deviations are below 0.19%, except for $N = 40$ and $p_{rel} = 0.9$ where the difference is approximately 2%.

In Fig. 7 we show the expected delay (D_i) as a function of the source grade (i) and the probability p_{rel}. To highlight the impact of p_{rel} over the network performance, for these experiments, we consider high node densities and high packet arrival rates. The figure shows that for small values of p_{rel} (e.g., $p_{rel} = 0.7$), the nodes that are close to the sink experience a superior performance than the remote nodes. The performance becomes more homogeneous by selecting a larger value of p_{rel} (e.g., 0.75); however, excessively large values of p_{rel} may significantly degrade the delay experienced

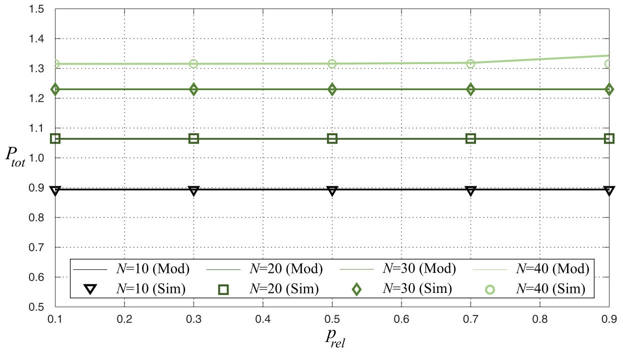

FIGURE 6. **Expected power consumption per node, in mW, in terms of the number of nodes in each grade (*N*) and the probability of choosing a packet from the relay buffer (p_{rel}), for $\lambda = 0.001875$.**

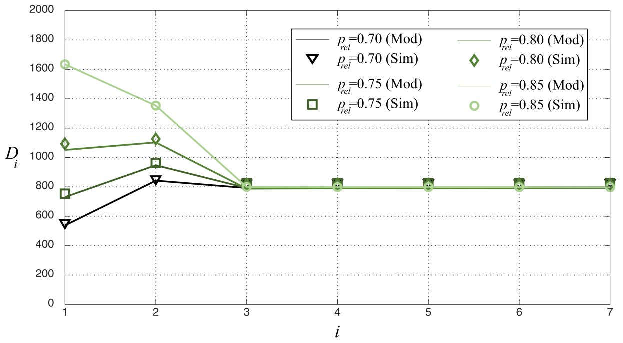

FIGURE 7. **Expected end-to-end delay, in seconds, for packets generated at grade *i* as a function of p_{rel}, for $N = 35$ and $\lambda = 0.001875$.**

by nodes close to the sink without improving the delay for remote nodes (e.g., 0.85). In this case, the differences between the analytical and simulated results are below 3%. These deviations result from small time variations (due to the variation of number of minislots required to access the channel in the link between grade 1 and the sink node) omitted from the analytical model for simplicity. These results indicate that the analytical model achieves a reasonable tradeoff between accuracy and simplicity.

Fig. 8 shows the behavior of the packet loss probability (p_{lp}) as a function of p_{rel} under high node density. For this set of parameters, we observe that the loss probability across the gades is insensitive to p_{rel}. Similar to the previous metrics, the figure shows that analytical and simulated results are very similar. In general, the deviation is smaller than 2.7% with the exception of the results for grade 2, where the differences are 6.4% and 13.2%, for $p_{rel} = 0.75$ and $p_{rel} = 0.8$, respectively. These deviations occur at the closest grade to the curve's inflection point and, in any case, they do not significantly affect the evaluation of the network.

By comprehensively analyzing the results from Figs. 5-8, we observe that the end-to-end delay is the most sensitive metric to the probability of serving the relay buffer. Therefore, p_{rel} should be selected to attain small end-to-end delays for the data packets generated from all the grades.

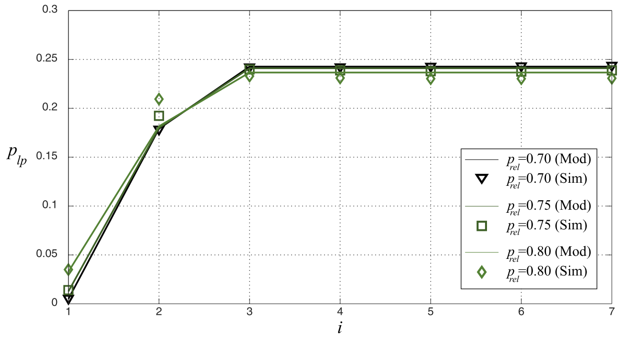

FIGURE 8. **Packet loss probability as a function of the grade (*i*) where the packet is generated and p_{rel}, for $N = 35$ and $\lambda = 0.001875$.**

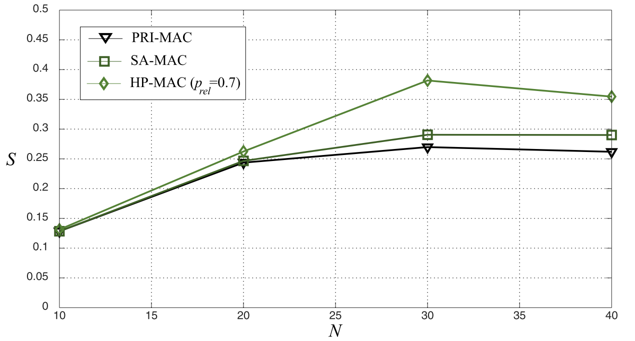

FIGURE 9. **Throughput, in packets/s, as a function of the number of nodes per grade (N), for $\lambda = 0.001875$.**

Figs. 9-13 present the results where we compare the performance attained by the proposed protocol against that of PRI-MAC [16] and SA-MAC [21]. We observe that HP-MAC outperforms the other protocols in terms of throughput, average power consumption, end-to-end delay, and packet loss probability.

Fig. 9 shows that HP-MAC achieves better throughput as the node density increases, which is a consequence of avoiding contention hot-spots due to collisions and reducing the cycle duration by limiting the number of minislots to the number of nodes in the grade. The latter feature contrasts with previous proposals where the number of required minislots exceeds the number of nodes in a grade.

As we observe in Fig. 10, HP-MAC significantly reduces power consumption in comparison with PRI-MAC and SA-MAC. This reduction is achieved because HP-MAC nodes with a packet to send can go to sleep as soon as they receive a CTS from a higher priority ticket holder. However, these savings are reduced as the number of nodes per grade increases because more nodes wake up at the beginning of the transmission slots to determine if they can access the channel (because nodes with higher priority tickets did not wake up).

Fig. 11 shows the average end-to-end delay attained by the different protocols under high packet arrival rate conditions. We observe that HP-MAC exhibits the best performance as it consistently achieves similar or lower delays than PRI-MAC and SA-MAC. This is particularly true for

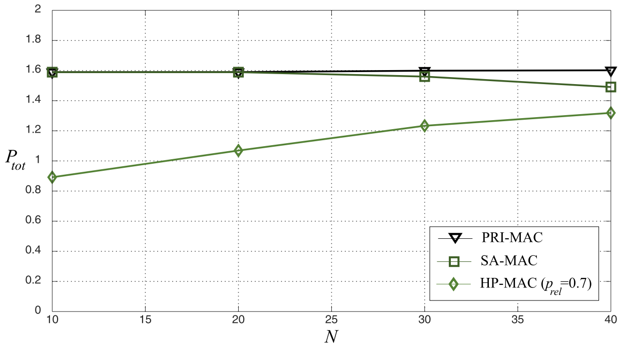

FIGURE 10. Average power consumption per node, in mW, as a function of N, $\lambda = 0.001875$.

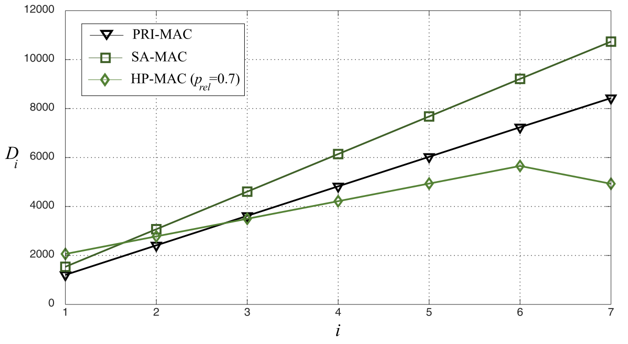

FIGURE 11. Average end-to-end delay, in seconds, for a packet generated at grade i, for $N = 35$ and $\lambda = 0.03$.

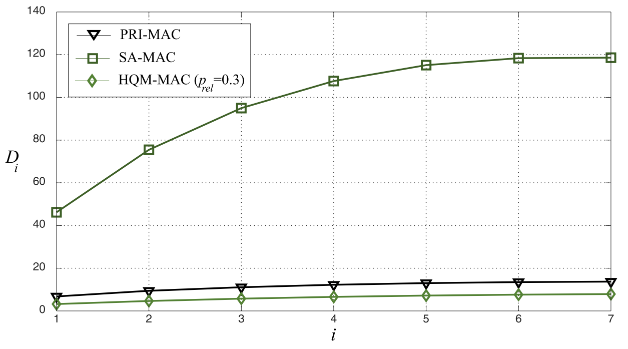

FIGURE 12. Average end-to-end delay, in seconds, for a packet generated at grade i, for $N = 35$ and $\lambda = 0.001$.

the packets generated at the last grade, where HP-MAC achieves a reduction of more than half of the delay attained by SA-MAC. On the other hand, from Fig. 12 we can observe that HP-MAC also outperforms the other protocols under low packet arrival rate conditions. In this case, however, we must select a small value for p_{rel} because there is no need to prioritize relay packets.

Lastly, in Fig. 13 we show the packet loss probability p_{lp} experienced by packets generated at the different grades. These results show the performance gains attained by eliminating collisions, and hence, contention hot-spots near the sink. From the figure, we can observe that HP-MAC outperforms SA-MAC and PRI-MAC by reducing the probability of losing packets generated at the first grade by an order of magnitude and the probability of losing packets generated at the last grades by up to half. It is important to point out that these performance gains come at no cost in terms of extra signaling overhead because HP-MAC uses distributed elections based on hash functions to determine which node can access the channel in an interference-free fashion.

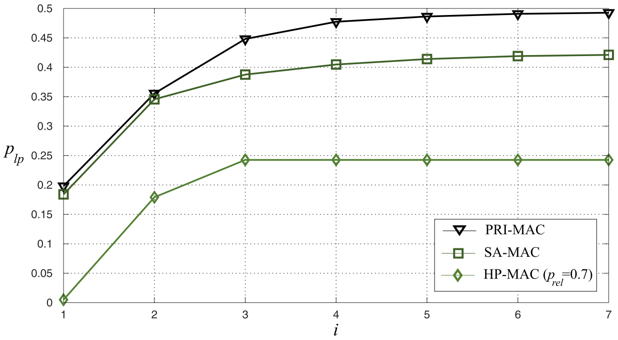

FIGURE 13. Packet loss probability as a function of the grade (i), for $N = 35$ and $\lambda = 0.001875$.

VIII. CONCLUSION

Despite the seeming simplicity of LSNs, these networks pose particular challenges to their designers. As discussed in the related work section, previous proposals do not scale well as the node density and network length (in number of grades) increases. The proposed analytical model and the simulation-based analysis show that without careful design, sensory data collected a few hops away from the sink experience unacceptably high end-to-end delays and rarely reach its destination. In practice, the differentiated performance experienced by nodes located at different grades is not acceptable since all sensory information is equally important.

In this paper, we presented HP-MAC, a MAC protocol specifically designed for the moderately long and dense LSNs found in IoT applications for smart cities. HP-MAC implements distributed elections based on hash functions to provide collision-free access to the channel without the need for additional control signaling or synchronization requirements. HP-MAC also implements a simple two-queue scheme that effectively assigns extra network resources to data flows generated hops away from the sink. Our simulation-based and analytical results show that, unlike previous proposals, HP-MAC provides adequate end-to-end delay and packet loss probability to nodes located at the last grades, even under high-traffic conditions. All these while reducing energy consumption. Our results show that HP-MAC outperforms representatives of the state-of-the-art in MAC protocols for LNS.

The Discrete-Time Markov Chain solution developed in our analytical model allowed us to derive closed-form expressions for the LSN performance parameters. In this regard, we would also like to highlight the novelty of the method that we developed to obtain the end-to-end delay, which is simpler and more accurate than previous proposals such as the ones presented at [16] and [21]. Moreover, our analytical model was validated by the very small deviations observed between its results and those obtained during the simulation-based experiments.

Future work includes extending our analyses to consider packet prioritization schemes where priorities depend on the packet lifetime and the buffer state. These schemes may further improve the end-to-end packet delay and energy consumption. In addition, we consider formulating the network design problem as a multi-objective optimization problem. In particular, we are interested in determining the right amount of resources assigned to each grade to simultaneously optimize energy consumption, loss probability, and end-to-end delay.

REFERENCES

[1] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustain. Soc.*, vol. 38, pp. 697–713, Apr. 2018.

[2] M. Weber and I. P. Žarko, "A regulatory view on smart city services," *Sensors*, vol. 19, no. 2, p. 415, Jan. 2019.

[3] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The sensable city: A survey on the deployment and management for smart city monitoring," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1533–1560, 2nd Quart., 2019.

[4] R. Lea and M. Blackstock, "City hub: A cloud-based IoT platform for smart cities," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2014, pp. 799–804.

[5] G. Suciu, A. Vulpe, S. Halunga, O. Fratu, G. Todoran, and V. Suciu, "Smart cities built on resilient cloud computing and secure Internet of Things," in *Proc. 19th Int. Conf. Control Syst. Comput. Sci.*, May 2013, pp. 513–518.

[6] R. Faria, L. Brito, K. Baras, and J. Silva, "Smart mobility: A survey," in *Proc. Int. Conf. Internet Things Global Community (IoTGC)*, Jul. 2017, pp. 1–8.

[7] G. Dileep, "A survey on smart grid technologies and applications," *Renew. Energy*, vol. 146, pp. 2589–2625, Feb. 2020.

[8] A. H. Alavi, P. Jiao, W. G. Buttlar, and N. Lajnef, "Internet of Things-enabled smart cities: State-of-the-art and future trends," *Measurement*, vol. 129, pp. 589–606, Dec. 2018.

[9] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Comput. Netw.*, vol. 144, pp. 17–39, Oct. 2018.

[10] S. Varshney, C. Kumar, and A. Swaroop, "Linear sensor networks: Applications, issues and major research trends," in *Proc. Int. Conf. Comput., Commun. Autom. (ICCCA)*, May 2015, pp. 446–451.

[11] A. Triantafyllou, P. Sarigiannidis, and T. D. Lagkas, "Network protocols, schemes, and mechanisms for Internet of Things (IoT): Features, open challenges, and trends," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–24, Sep. 2018.

[12] F. Alfayez, M. Hammoudeh, and A. Abuarqoub, "A survey on MAC protocols for duty-cycled wireless sensor networks," *Proc. Comput. Sci.*, vol. 73, pp. 482–489, Jan. 2015.

[13] A. Kumar, M. Zhao, K. Wong, Y. L. Guan, and P. H. J. Chong, "A comprehensive study of IoT and WSN MAC protocols: Research issues, challenges and opportunities," *IEEE Access*, vol. 6, pp. 76228–76262, 2018.

[14] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Jul. 2002, pp. 1567–1576.

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

[16] F. Tong, L. Zheng, M. Ahmadi, M. Ni, and J. Pan, "Modeling and analyzing duty-cycling pipelined-scheduling MAC for linear sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2608–2620, Apr. 2016.

[17] J. Wang, C. Hu, and A. Liu, "Comprehensive optimization of energy consumption and delay performance for green communication in Internet of Things," *Mobile Inf. Syst.*, vol. 2017, pp. 1–17, Mar. 2017.

[18] F. Tong, S. He, and J. Pan, "Modeling and analysis for data collection in duty-cycled linear sensor networks with pipelined-forwarding feature," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9489–9502, Dec. 2019.

[19] Q. Zhang, D. Li, Y. Fei, J. Zhang, Y. Chen, and F. Tong, "RDCPF: A redundancy-based duty-cycling pipelined-forwarding MAC for linear sensor networks," *Sensors*, vol. 20, no. 19, p. 5608, Sep. 2020.

[20] F. Tong and Y. Peng, "A data-gathering, dynamic duty-cycling MAC protocol for large-scale wireless sensor networks," *Sensors*, vol. 20, no. 15, p. 4071, Jul. 2020.

[21] I. Villordo-Jimenez, N. Torres-Cruz, M. M. Carvalho, R. Menchaca-Mendez, M. E. Rivero-Angeles, and R. Menchaca-Mendez, "A selective-awakening MAC protocol for energy-efficient data forwarding in linear sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–18, May 2018, doi: 10.1155/2018/6351623.

[22] I. Villordo-Jimenez, R. Menchaca-Mendez, M. E. Rivero-Angeles, and N. Torres-Cruz, "An energy-efficient hash-based MAC protocol for LSN," in *Proc. IEEE 10th Latin-Amer. Conf. Commun. (LATINCOM)*, Guadalajara, Mexico, Nov. 2018, pp. 1–6.

[23] C. E. Acosta., F. Gil-Castiñeira, E. Costa-Montenegro, and J. S. Silva, "Reliable link level routing algorithm in pipeline monitoring using implicit acknowledgements," *Sensors*, vol. 21, no. 3, p. 968, Feb. 2021.

[24] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Netw.*, vol. 12, no. 1, pp. 63–78, Feb. 2006.

[25] J. J. Camacho-Escoto, R. Menchaca-Mendez, R. Menchaca-Mendez, J. Bernal, M. E. Rivero-Angeles, J. Gomez, and J. J. Garcia-Luna-Aceves, "Energy and bandwidth-efficient channel access for local area machine-to-machine communication," *Wireless Netw.*, vol. 27, no. 1, pp. 401–421, Jan. 2021.

[26] F. A. Alfouzan, A. Shahrabi, S. M. Ghoreyshi, and T. Boutaleb, "A collision-free graph coloring MAC protocol for underwater sensor networks," *IEEE Access*, vol. 7, pp. 39862–39878, 2019.

[27] M. Pavani and P. T. Rao, "Novel two-fold data aggregation and MAC scheduling to support energy efficient routing in wireless sensor network," *IEEE Access*, vol. 7, pp. 1260–1274, 2019.

[28] H. Lee, J. Hong, S. Yang, I. Jang, and H. Yoon, "A pseudo-random asynchronous duty cycle MAC protocol in wireless sensor networks," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 136–138, Feb. 2010.

[29] L. Farhan, A. E. Alissa, S. T. Shukur, M. Hammoudeh, and R. Kharel, "An energy efficient long hop (LH) first scheduling algorithm for scalable Internet of Things (IoT) networks," in *Proc. 11th Int. Conf. Sens. Technol. (ICST)*, Dec. 2017, pp. 1–6.

[30] S. Argoubi, K. Maalaoui, M. H. Elhdhili, and L. A. Saidane, "Priority-MAC: A priority based medium access control solution with QoS for WSN," in *Proc. IEEE/ACS 13th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2016, pp. 1–6.

[31] N. Nasser, L. Karim, and T. Taleb, "Dynamic multilevel priority packet scheduling scheme for wireless sensor network," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1448–1459, Apr. 2013.

[32] S. Bansode and S. Sambare, "Performance evaluation of dynamic multilevel priority (DMP) packet scheduling method for wireless sensor networks (WSNs)," in *Proc. Int. Conf. Pervasive Comput. (ICPC)*, Jan. 2015, pp. 1–6.

[33] W. Yantong and Z. Sheng, "An enhanced dynamic priority packet scheduling algorithm in wireless sensor networks," in *Proc. UKSim-AMSS 18th Int. Conf. Comput. Modelling Simulation (UKSim)*, Apr. 2016, pp. 311–316.

[34] A. N. Sakib, M. Drieberg, and A. A. Aziz, "Energy-efficient synchronous MAC protocol based on QoS and multi-priority for wireless sensor networks," in *Proc. IEEE 11th IEEE Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Apr. 2021, pp. 347–352.

[35] B. A. Muzakkari, M. A. Mohamed, M. F. Kadir, and M. Mamat, "Queue and priority-aware adaptive duty cycle scheme for energy efficient wireless sensor networks," *IEEE Access*, vol. 8, pp. 17231–17242, 2020.

[36] N. T. Thu-Hang, N. C. Trinh, and N. T. Ban, "Delay and reliability analysis of p-persistent carrier sense multiple access for multievent wireless sensor network," in *Proc. 26th Int. Conf. Telecommun. (ICT)*, Apr. 2019, pp. 427–431.

[37] M. Mitzenmacher and E. Upfal, *Probability and Computing Randomization and Probabilistic Techniques in Algorithms and Data Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2017, ch. 15.

[38] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for ad hoc networks," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2001, pp. 210–221.

[39] L. Bao and J. J. Garcia-Luna-Aceves, "Distributed dynamic channel access scheduling for ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 63, no. 1, pp. 3–14, Jan. 2003.

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

ICLIA VILLORDO-JIMENEZ received the B.Sc. degree in electronic engineering from the Instituto Tecnológico de Puebla, Puebla, México, in 2002, the M.Sc. degree in electronic from INAOE, Tonantzintla, Puebla, in 2007, and the Ph.D. degree in computer science from the Instituto Politécnico Nacional (IPN), México, in 2020. She is currently a Professor at the Instituto Politécnico Nacional (IPN). Her research interests include green wireless sensor networks and data transmission in 5G and beyond cellular networks.

NOÉ TORRES-CRUZ received the B.S. degree in electronic engineering from ITO, México, the M.Sc. degree in electrical engineering from CINVESTAV-IPN, México, and the Ph.D. degree in computer science from CIC-IPN, Mexico, in 2019. He spent a research stay at the Universidade de Brasília (UnB), Brazil, in 2017. He is currently a Professor at the Instituto Politécnico Nacional (IPN), México. His research interests include resource management in mobile networks and video distribution strategies for video streaming services.

ROLANDO MENCHACA-MENDEZ received the Ph.D. degree in computer engineering from the University of California at Santa Cruz, Santa Cruz, CA, USA, in 2009. He is currently a Professor of the Network and Data Science Laboratory, Computer Research Center, Mexican National Polytechnic Institute, and a Professor at the Instituto Politécnico Nacional (IPN). His current research interests include mobile computing, combinatorial optimization, and the analysis and design of algorithms and protocols for computer communications.

MARIO E. RIVERO-ANGELES received the B.Sc. degree in electrical engineering from Metropolitan Autonomous University (UAM), Mexico, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering from CINVESTAV-IPN, in 2000 and 2006, respectively. He was a Postdoctoral Fellow for the Dyonisos Research Project at the Institut National de Recherche en Informatique et en Automatique (INRIA), Rennes, France, from 2007 to 2010. He has been a Professor at the Instituto Politécnico Nacional (IPN), Mexico City, since 2002. His research interests include random access protocols and data transmission in cellular networks.