Analysis of a Novel Media Access Control Protocol for LoRa

Chen Zhong and Andreas Springer, *Member, IEEE*

***Abstract*—Long-range wide-area networks (LoRaWANs) are emerging technologies for the Internet of Things. They provide ultralong-range communication and low power consumption but have poor scalability due to the ALOHA media access method. In this article, we first present a novel medium access control (MAC) protocol which is time-slot-based and channel contention constrained to enhance the network performance in terms of throughput and energy consumption. We propose an analytical model which is based on three assumptions and Markov chains to analyze the performance. Various protocol and network scale settings are considered during the analysis of the performance of the network. We performed simulations in MATLAB to verify the validity of the assumptions and the accuracy of the model. We show that our new protocol enhances the throughput by approximately two times as compared to ALOHA in LoRaWAN networks, while the average energy consumption per packet is significantly reduced if the optimum protocol settings are used.**

***Index Terms*—Analysis, long-range wide-area network (LoRaWAN), low-power, medium access control (MAC), simulation, throughput.**

I. Introduction

THE LOW-POWER wide-area networks (LPWANs) market is predicted to reach over a hundred billion dollars in the next five years [1]. Long-range wide-area network (LoRaWAN) is the fastest growing technology in this sector which provides long-range, low-power, and low-data-rate IoT connectivity services. It primarily operates in unlicensed Sub-GHz spectrum all over the world and supports both public networks which are owned and maintained by telecom operators, and private networks in which network elements, such as gateways and network servers are under the control of users for the sake of security and privacy.

There are three operation modes defined in the LoRaWAN specification [2] which are Classes A, B, and C. All end nodes are required to support the Class A mode, with which the lowest power consumption can be achieved. In this mode, all communications are initiated by end nodes and, as a result, they can stay in the sleep mode most of the time if only a small amount of data needs to be reported to the server. After the uplink transmission, an end node will open two consecutive time-scheduled receive windows for the downlink transmissions that are sent by the gateway. Class A mode is most widely used in many applications, especially where the low-power operation is required [3], [4]. The main weakness of the Class A mode is that many packet collisions occur since the nodes start to transmit as soon as data is available (i.e., the ALOHA protocol is adopted in the current LoRaWAN specifications). To mitigate the packet collision issue, the Class B mode is designed in a way that all end nodes are synchronized and each of them opens a receive window at a certain time slot which is allocated by a random generator initialized by the device identification of the nodes. All communications in the Class B mode are initiated by the gateway, where the gateway first sends a request to the end node in a certain slot, and only the appointed end node will respond in that slot. As a result, concurrent transmissions are eliminated. But the drawback is that the packet delay of Class B mode and its variants, which are also based on the scheduling approach [5], [6], [7], [8], is much higher than in the Class A mode since the node needs to hold the data until it is requested. Class C mode targets real-time applications [9] where downlink data needs to be delivered to an end node promptly. End nodes need to open their receive window continuously and, thus, power supply for the end devices is usually required for the network operation in the Class C mode.

LoRaWAN provides reliable communications in two modes, which are unconfirmed and confirmed, respectively. If confirmed messaging is adopted, each uplink transmission will be acknowledged by a net server that is usually deployed in a cloud. There is no feedback on the uplink transmission to the end node in unconfirmed messaging. Confirmed messaging is required if the data is crucial for an application [10], [11], [12]. In contrast, unconfirmed messaging is more suitable in applications, such as the collection of ambient information which is changing continuously, and retransmissions of obsolete information are unnecessary [13].

Though LoRaWAN attracts increasing interest both from industry and academia, there are still issues to be solved [14]. First, there has been no model proposed yet for the LoRaWAN medium access control (MAC) protocol. Network performance analysis based on modeling is common [15], [16], as it helps compare the performance of different protocols in all network layers and optimize system parameters, while it takes less effort and time as compared to a network simulator or experiments conducted on hardware. Mikhaylov *et al.* [17] computed the maximum number of end nodes that can be served by a single LoRaWAN gateway and discussed the

Manuscript received 6 May 2022; revised 21 July 2022; accepted 15 August 2022. Date of publication 22 August 2022; date of current version 22 December 2022. This work was supported in part by the Huiju Intelligent Tech Company Ltd., and in part by the CSOLMN Project. *(Corresponding author: Chen Zhong.)*

The authors are with the Institute for Communications Engineering and RF-Systems, Johannes Kepler University Linz, 4040 Linz, Austria (e-mail: chen.zhong.ltu@gmail.com).

Digital Object Identifier 10.1109/JIOT.2022.3200435

spatial distribution of these end nodes with the assumption that the performance of the LoRaWAN MAC protocol conforms to pure ALOHA, which has been experimentally shown is inaccurate [18], [19]. A second analysis that is similar to the adoption of pure ALOHA was presented in [20], and Adelantado *et al.* [21] also estimated LoRaWAN capacity as the superposition of independent ALOHA-based networks. In summary, many studies assumed that the behavior of LoRaWAN complies with pure ALOHA where concurrent transmissions are always lost regardless of their received power levels and timings. Recognizing the error produced by using a pure ALOHA model, a number of works that used simulations have been published. A model implemented in the Network Simulator 3 was presented in [22]. Interference between multiple simultaneous transmissions and the communication range was considered in an error model. Bor *et al.* [23] developed models describing LoRa link behaviors which comprise communication range and collision. Based on these physical layer models, a simulation tool called LoRaSim was developed as well to analyze the network performance on the MAC layer. In conclusion, published works either used pure ALOHA to analyze the scalability of the LoRaWAN MAC protocol with inevitable mismatch to reality or developed a dedicated simulation tool that better describes the network but at the cost of high implementation effort.

Second, the throughput of the ALOHA protocol, which is currently used in LoRaWAN for confirmed traffic, is very low. Polonelli *et al.* [24] showed that the maximum achievable throughput for ALOHA is 8% at an offered load of 0.25 when the confirmed traffic is used. Additionally, the scenarios, where confirmed and unconfirmed traffic are simultaneously used were also investigated. Centenaro *et al.* [25] built a simulator in MATLAB to study how confirmed traffic negatively impacts network performance. They found that the throughput for both confirmed and unconfirmed traffic reduces with the ratio between these two types of traffic. Similarly, Varsier and Schwoerer [3] also built a MATLAB-based simulation tool to investigate the cases where 25%, 33%, and 100% of end nodes are sending confirmed uplink traffic and their results showed that the packet error rate (PER) per end node increases drastically with the percentage of nodes sending confirmed traffic and the size of the network.

Third, though a number of works have been published with the purpose of improving throughput and the energy consumption, however, further improvements are desired. Kouvelas *et al.* [26] suggested a p-persistent Carrier Sense Multiple Access (p-CSMA) [27] component for the simulation of LoRaWAN in Network Simulator 3. They measured the packet reception ratio (PRR) under various persistence values, node numbers, communication ranges, and interference situations. The PRR performance of p-CSMA is better than ALOHA and the lower the persistence value is, the higher the achieved PRR. However, its energy consumption was not investigated. p-persistent CSMA adopts carrier sense to avoid packet collisions, which drains the same amount of current as receiving a packet. Therefore, p-CSMA increases energy consumption compared to the ALOHA protocol which does not conduct any channel sensing prior to transmitting a packet. Gamage *et al.* [18] proposed their novel MAC protocol consisting of LMAC-1, 2, and 3. LMAC-1 always conducts carrier sense during its contention access period (CAP) which is similar to p-CSMA. Therefore, we predict that the energy consumption performance of LMAC-1 is unsatisfactory, especially when the traffic load is high and the channel is continuously busy. However, its energy efficiency is higher than that of ALOHA (the explicit energy consumption performance on each single channel-spreading factor combination was not presented). A similar MAC protocol called np-CECADA with the consideration of the hidden terminal issues was presented in [28]. Before transmission, end nodes infer the potential hidden terminal interference through the information broadcasted by the gateway periodically. This MAC protocol reduced the energy consumption by 30% of the standard LoRaWAN MAC in the case of spreading factor of 12, while it cannot improve the energy consumption in the case of spreading factor of 7 compared with the standard LoRaWAN MAC.

Finally, most of the state-of-the-art solutions exploit the multiple channel and orthogonal spreading factor features of LoRa to achieve performance enhancements [29], [30], [31], while the studies on single channel and single spreading factor LoRa networks are increasing: their capacity is sufficient for most of the low-power-wide-area IoT applications, they achieve 13-fold reduction of costs for the simpler gateway, and they can be easily deployed as private networks [32], [33]. This article thus targets single channel and single spreading factor networks.

The contributions of this work are as follows.

1) We propose a novel MAC protocol to improve the throughput and retain low-power consumption for both unconfirmed and confirmed traffic in LoRaWAN.
2) We propose a model to describe the operation and analyze the performance of our new MAC protocol for LoRaWAN. The validity of the analysis is verified by comparing its results with that of simulations.

This article is organized as follows. Section II introduces the design of our novel MAC protocol. Section III describes network settings at the physical layer of LoRaWAN. The analysis that is based on Markov chain modeling for our MAC protocol is detailed in Section IV. Simulation results to validate our analysis of the new MAC protocol and evaluate its performance under diverse circumstance are presented in Section V.

II. Design of the Protocol

Our new MAC protocol is a time-slotted contention constrained persistent carrier sense multiple access method (CCP). First, we define a generic time element called backoff period (BP), which has a duration of 2 LoRa signal symbols for sufficiently sensing the carrier [34], and the duration of a LoRa signal symbol, T_{symbol} is given by

$$T_{\text{symbol}} = \frac{2^{\text{SF}}}{W} \tag{1}$$

where SF is the spreading factor and W represents the bandwidth of the communication channel (either 125, 250, or

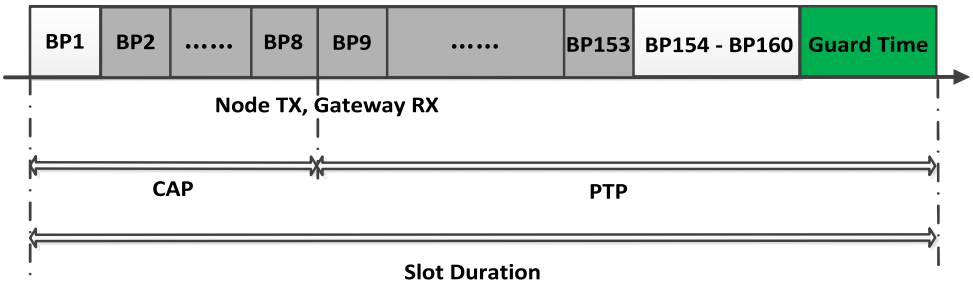

Fig. 1. Design of a slot for contention constrained persistent-Carrier Sense Multiple Access (CCP) when unconfirmed messaging is used. The slot consists of two sections: CAP and PTP. The gray and green segments represent an uplink transmission and slot guard time intervals, respectively. All white portions are left unused. A sequence number (in the range of 1–160) is attached to each BP.

500 kHz). The LoRa packet consists of a preamble and a payload, and according to the description in [34], its duration in number of symbols can be obtained through (2) and (3), shown at the bottom of the page, with the following dependencies and settings.

1) PL is the physical layer payload length (number of bytes) and we set it to 255 and 220 bytes for unconfirmed and confirmed messaging, respectively.
2) The spreading factor SF is set to 9, which is a median value in the range of 6–12 and is widely used for single channel and single spreading factor LoRa networks.
3) Cyclic redundancy check (CRC) is used in our study to detect corrupted packets; thus, CRC is equal to 1.
4) We chose a coding rate (CR) of 4/5.
5) n_{payload} is the number of symbols for the payload.
6) n_{preamble} is the number of symbols for the preamble, and we set it to 5 which is the minimum length for the reception of a LoRa packet [23].
7) B represents the number of symbols for a BP (i.e., 2 LoRa symbols for one BP according to the definition).
8) N is the number of BPs to transmit a packet including the preamble and the payload.

Different parameter settings lead to performance differences. For example, the maximum PL and transmission duration for SF12 is much smaller and longer compared to SF7, which results in much lower throughput and higher energy consumption. However, our MAC protocol does not have any requirement or preference on the selection of the LoRa parameters. Because the protocol control of CCP does not rely on LoRa parameters but its own settings (i.e., persistent value and segment lengths). Using the same LoRa parameters, we will show that the performance of CCP is always far better than ALOHA and p-CSMA. We suggest that LoRa parameters should be chosen according to the applications. For the purpose of illustrating our protocol, we choose these listed LoRa parameters which are most widely used according to our experience and are given as an example to demonstrate the efficiency of the CCP (i.e., first, the LoRa parameters should be chosen to suit the applications and then, the performance of CCP with these parameters can be analyzed or simulated to see if CCP outperforms ALOHA and p-CSMA).

In the following, we describe our new protocol. As illustrated in Fig. 1, time is divided into slots which consist of two sections: 1) CAP and 2) packet transmission period (PTP). The duration for CAP is set to 8 BP [35]. In order to accommodate a packet with a payload of 255 bytes and a preamble of five symbols, the duration of PTP is designed to be 152 BPs [as computed from (2) and (3)] to which a guard time of 2 BPs is appended to consider synchronization errors among end nodes. As a result, the length of PTP is $152 + 2 = 154$ BPs, and the total length of a slot can be computed as $8 + 154 = 162$ BPs. It is worth pointing out that the packet length and the resulting PTP duration can be shortened depending on the application need, and the setting of slot duration can be communicated either during the joining procedure of the end node or the periodic beacon broadcast. After knowing the slot duration and synchronized with the gateway, any end node which wants to send a packet will wake up at the start of a slot and first, experience a uniformly distributed random BP in the range between 0 and 7 BPs. After completing the random backoff, the end node senses the channel once. If it is idle, the end node transmits a packet in the next BP with a probability p. Otherwise, it repeats the channel sensing with probability $(1 - p)$. This process is executed over and over like p-CSMA until one of the following conditions is met.

1) The packet is transmitted.
2) The end of CAP is reached (i.e., BP8).
3) The channel is sensed busy in a certain BP of CAP.

In the case the second condition is met, the end node enters sleep mode immediately and remains in that mode until the end of the current slot to conserve power. Thus, the contention for the channel is constrained to CAP with persistence probability p. If the third condition is encountered, the node will sleep until the start of the next slot because the current slot

$$n_{\text{payload}} = 8 + \max\left(\text{ceil}\left(\frac{8\text{PL} - 4\text{SF} + 28 + 16\text{CRC}}{4\text{SF}}\right)(\text{CR} + 4), 0\right) \tag{2}$$

$$N = \frac{\left(n_{\text{preamble}} + 4.25\right) + n_{\text{payload}}}{B} \tag{3}$$

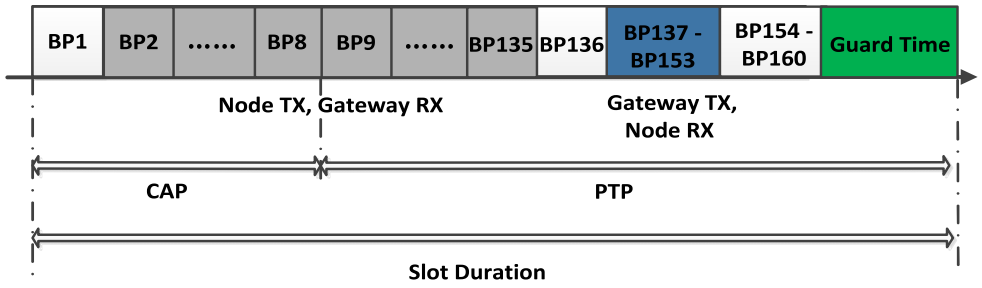

Fig. 2. Design of a slot for contention constrained persistent-Carrier Sense Multiple Access (CCP) when confirmed messaging is used. The slot consists of two sections: CAP and PTP. The gray, blue, and green segments represent an uplink transmission, an acknowledgment, and slot guard time intervals, respectively. All white portions are left unused.

is seized by some other node and all nodes holding a packet have a new chance to access the channel in the next slot.

Note that for a slot-based protocol, time synchronization is a critical issue due to the clock drift phenomenon among the devices. Our proposed slot design (Figs. 1 and 2) a guard time of 2 BPs to deal with timing synchronization errors. For LoRa networks synchronization algorithms either via dedicated downlink broadcast message in a predefined period or via regular bidirectional communication have been introduced in [36] and [37], respectively. Both of them can be used with our protocol. We further assume that end nodes should be equipped with commonly used external crystals running at 32.768 kHz with a precision of 10 or 20 ppm. In that case the maximum synchronization period T_{SYN} between the gateway and end nodes can be computed via

$$T_{\mathrm{SYN}} = \frac{T_{\mathrm{GT}}}{2E_{\mathrm{PPM}}} \tag{4}$$

where T_{GT} and E_{PPM} denote the guard time (2 BPs) and the crystal precision, respectively. For a crystal with precision of 20 ppm, the synchronization period is 5×10^4 BPs, which is approximately as long as 309 slots. If regular bidirectional communication is used for synchronization, no extra expenditure for synchronizations is required and all time slots are utilized for regular communications (i.e., the effective throughput is maximized). In the worst case, if a periodically broadcasted message (i.e., a beacon) is transmitted in a dedicated slot, the synchronization overhead accounts for a fraction of $1/309 \approx 0.003$ of the entire throughput, which is negligible. Furthermore, to avoid the collisions between the beacon message and the uplink transmissions, the beacon message is transmitted in the first BP of the slot where all nodes either sleep or sense the carrier. Thus, the synchronization can be ensured with negligible cost in our protocol. In the following, we thus only take regular communication into account for the analysis and simulation of the performance of our protocol.

A. Unconfirmed Messaging

Fig. 1 illustrates the scenario that an unconfirmed packet is transmitted after zero random backoff and one carrier sense activity in BP1. The uplink transmission is started in BP2 and completed in BP153, and the interval that begins in BP154 and ends in BP160 is left idle. Thus, the packet size is 152 BPs including the preamble which is able to carry 255 bytes of payload. The guard time duration is 2 BPs which is used to handle the timing deviations on all nodes.

B. Confirmed Messaging

The timing allocation for confirmed messaging is more complex than for unconfirmed messaging. Fig. 2 shows an example where the upload transmission starts in BP2 again as in the unconfirmed scenario in Fig. 1. However, in order to reserve some BPs for the downlink transmission, the uplink transmission ends in BP135 which is 18 BPs earlier than in the case of unconfirmed messaging. Thus, the permitted length of the confirmed uplink packet is reduced to 134 BPs, which can carry a payload of 220 bytes. BP136 is left unused for the turn-around time between transmit and receive modes of the devices. The acknowledgment packet is transmitted by the gateway in the period from BP137 to BP153. Again the rest of the slot ranging from BP154 to BP160 is unused, and it is worth pointing out that this unused period will be shortened up to 0 BP when the uplink transmission starts later.

III. Network Settings

Besides the basic settings that are described previously, several network settings and deployments that are necessary for the analysis and simulations are described in the following.

1) The duration for each simulation iteration is 400 s.
2) Uplink and downlink packet reception rates without collision are both set to 98%. This value is drawn from an experiment where an end node continuously transmits 255-bytes packets to a gateway that is placed 9-m away from the transmitter [38].
3) All devices are stationary and located in a collision domain such that all devices mutually interfere/collide with each other. The number of nodes is set to 100, 200, and 500, respectively. As there are only 8 BPs in the CAP it seems that many packet collisions appear (i.e., multiple nodes start to transmit a packet in the same BP of CAP) if numerous nodes feature a high packet arrival rate (the probability that a node receives a packet from up protocol layer in each slot). Thus, our setup includes also situations with a large number of collisions.
4) Buffering at the nodes is not considered. New packets are discarded when the node is currently transmitting or is attempting a transmission.

5) Both unconfirmed and confirmed uplink packets are transmitted once without retransmissions.
6) The transmission power of all devices is set to 10 dBm. We presume that there is only one gateway and all end nodes can deliver packets to the gateway with a probability of 98% if no collisions occur.
7) The packet arrival rate in a slot for both CCP and CSMA is in a range of 10^{-4} to 1. Therefore, the total offered traffic (measured in the number of packets) to the network is equal to the product of packet arrival rate, the number of nodes, and the number of slots.
8) Retransmission for confirmed messaging is not used in analysis, thus confirmed packets are sent only once.
9) LoRa is based on frequency modulation which shows a capture effect [39]. Furthermore, LoRa provides the so-called channel activity detection (CAD) feature, which correlates the captured samples against the ideal signal waveform with a specified spreading factor to detect LoRa signals [18]. Owing to the network synchronization and CAD function used in this work, the time difference of multiple uplink transmissions starting in the same BP of CAP is always less than one BP. According to the research results presented in [23] and [40], the strongest packet in the case of diverse received signal strengths (RSSs) [23] or the slightly later transmitted packet in the case of identical RSSs [40] can be received successfully even if collisions occur (simultaneous arrival of two or more packets), while the weaker or earlier transmitted packets cannot be received.
10) As discussed before, our CCP protocol is mainly dedicated to single channel and single spreading factor LoRa networks, as used for, e.g., home automation and smart agriculture. The number of nodes deployed in a certain area of these scenarios is likely on a scale of several hundred. Thus, we decide to set the number of nodes to 100, 200, and 500, respectively.

Note that all settings are identically applied for both CCP and p-CSMA in simulations and analysis to ensure a fair performance comparison.

IV. Analysis of the CCP Protocol

In the following, we use Markov chains to model the CCP protocol. Three approximations are introduced and discussed below for the sake of tractability.

1) *APPROXIMATION 1:* For the probability that the channel is idle in a given BP, we use the approach proposed in [41], which employs the steady-state probability that the channel is idle as an approximation. This probability is denoted by p_i^{CI}, where the subscript i, $1 \le i \le 9$ denotes a specific BP in a slot (i.e., this i has the same function as the i in BPi, $1 \le i \le 160$ as illustrated in Fig. 1).
2) *APPROXIMATION 2:* The probability that an individual end node starts to transmit in the allowed BPs which is the range of BP2–BP9 is essential to obtain the channel throughput and collision probability. However, it is difficult to compute. Thus again, we use the approach proposed in [41], which approximates this probability with the steady-state probability that any node starts to transmit. This probability is denoted by p_i^{NT}, where the subscript i denotes a specific BP in a slot in which the node starts to transmit a packet.
3) *APPROXIMATION 3:* Due to the single channel and single spreading factor assumption for our investigation and due to the capture effect of LoRa, only one uplink packet per slot is received correctly. Thus, the gateway only needs to acknowledge this received uplink packet. Due to network synchronization and the CAD function of LoRa, the difference of packet reception times between the received packet and all other lost packets is less than one BP. As a result, the received signal and an acknowledgment signal at the gateway might be overlapping by up to one BP. To avoid this, we inserted one BP between uplink and downlink transmission (BP136 in Fig. 2). Based on these considerations, we assume that in our CCP protocol the acknowledgment is always successfully received once the uplink transmission was successful. Thus, in both unconfirmed and confirmed messaging, there will be only one uplink transmission successfully received, and the throughput difference between unconfirmed and confirmed messaging is only one BP (i.e., in Fig. 2 BP136) which results in $1/152 \approx 0.66\%$. Thus, the throughput performance of confirmed messaging is almost identical to unconfirmed messaging in our CCP protocol if the acknowledgment in confirmed messaging is regarded as a part of the throughput. Therefore, we use the results of unconfirmed messaging to approximate the performance of confirmed messaging.

A. Node State Model

We model the behavior of an individual end node by means of a Markov chain as illustrated in Fig. 3. A node is in the sleep state (denoted by L_S in Fig. 3 with the subscript S representing the duration of an entire slot) when it does not have a packet to transmit. If a packet arrives with probability λ, the node transitions to one out of the eight backOff stages, O_0 to O_7 with a uniform probability of $(1/8)\lambda$. There is no backoff at all when the node transitions to O_0 and the longest backoff is 7 BPs when it transitions to O_7.

After the initial random backoff, the node starts to sense the channel. Thus, on leaving O_i, $0 \le i \le 7$, it changes to Channel sensing state C_{i+1} with probability 1. Here, C_{i+1} denotes that the node senses the carrier in BP$i+1$. For example, the change from O_3 to C_4 denotes that the node senses the channel in BP4. If the channel is sensed for idle in BPi, which occurs with probability p_i^{CI}, the node changes to the packet transmission state A_{i+1} with probability $p_i^{\mathrm{CI}}P$, p_i^{CI} is the steady-state probability that the channel is idle in BPi (see Approximation 1). P is the transmission probability when the node senses the channel idle. In state A_{i+1} the node transmits the packet starting from BP$i+1$ to the end of the transmission (e.g., the period from BP2 to BP153 in Fig. 1) and stays in sleep mode during the unused and guard time BPs (i.e., BP154 – BP160 and

GuardTime in Fig. 1). Note that the transition probability from C_1 to A_2 is P since any node either sleeps or senses the channel in BP1 and thus the channel is definitively idle (i.e., p_1^{CI} is equal to 1). If the node does not transmit in BPi, it enters the next carrier sense state, which is C_{i+1} with probability $p_i^{\mathrm{CI}}(1-P)$. Note that the probability of transition from C_1 to C_2 is $(1-P)$ since p_1^{CI} is equal to 1. After the node completes the packet transmission in state A_i, $2 \le i \le 9$, it either jumps to state L_S when there is no packet to transmit in the next slot or jumps to O_i, $0 \le i \le 7$ with uniform probability $(1/8)\lambda$ when there is a packet to transmit.

On the other hand, if the channel is sensed busy in state C_i, $2 \le i \le 8$, which happens with probability $(1-p_i^{\mathrm{CI}})$, the node transitions to state L_{i+1}, which denotes that the node remains in sleep state from BP$i+1$ until the end of the current slot. In such circumstance, the end node needs to wake up at the beginning of the next slot to be ready for the channel contention since there is a packet left in its buffer and it changes to one out of eight O_i states with a uniform probability 1/8.

The steady-state probabilities of a node can be computed by solving the balance of the Markov chain (5) and (6), shown at the bottom of the page. The notation $\pi(\text{state}_i)$ denotes the steady-state probability of transition into state$_i$.

B. Node Transmission Probability

Ramachandran *et al.* [41] proposed a new way to evaluate the probability that any node would start to transmit in a generic backoff slot for the IEEE 802.15.4 standard [42]. We adopted the same way with the adaptation that the evaluation of the probability of the start of transmission is conducted in each BP $\in$ [BP2 BP9] for every slot, as all nodes are restricted to start to transmit in these BPs with different probabilities which are given by (7), shown at the bottom of the page. The probability that a node starts to transmit (i.e., p_i^{NT}) in BPi is equal to the probability that it was sensing the carrier in the previous BP (i.e., p_{i-1}^{NS}) multiplied by p_{i-1}^{CI} and P, where p_{i-1}^{CI} is the probability that the channel was idle in the previous BP and P denotes the transmission probability. Note that $\pi(L_S)$, $\pi(O_j)$, $\pi(C_i)$, and $\pi(A_j)$ are the steady-state probabilities of being in states: sleeping during the entire slot (L_S), backoff until O_j, carrier sensing in BPi (C_i), and transmission since BPj (A_j).

C. Channel State Model

Based on the probability p_i^{NT} that an individual node starts to transmit in an allowed BP, a Markov chain model for the channel states can be developed. Suppose the channel is idle in BPi which is denoted by D_i, $1 \le i \le 9$, it continues to remain in that state (i.e., D_{i+1}) if none of the nodes start to transmit, which occurs with probability given by (8)

$$\alpha_i = \left(1 - p_{i|\mathrm{CIP}}^{\mathrm{NT}}\right)^M \tag{8}$$

$$p_{i|\mathrm{CIP}}^{\mathrm{NT}} = \frac{p_i^{\mathrm{NT}}}{p_{i-1}^{\mathrm{CI}}} = p_{i-1}^{\mathrm{NS}} P \tag{9}$$

where $p_{i|\mathrm{CIP}}^{\mathrm{NT}}$ is the probability that the node is transmitting in BPi given that the channel was idle in the previous BP, and

$$\begin{aligned}
\pi(L_S) &= (1-\lambda)\pi(L_S) + (1-\lambda)\sum_{i=2}^{9}\pi(A_i)\\
\pi(O_i) &= \frac{1}{8}\lambda\pi(L_S) + \frac{1}{8}\lambda\sum_{j=2}^{9}\pi\left(A_j\right) + \frac{1}{8}\sum_{j=3}^{9}\pi\left(L_j\right), \text{for } i = 0, 1, \ldots, 7\\
\pi(C_1) &= \pi(O_0)\\
\pi(C_2) &= \pi(O_1) + (1-P)\pi(C_1)\\
\pi(C_i) &= \pi(O_{i-1}) + p_{i-1}^{\mathrm{CI}}(1-P)\pi(C_{i-1}), \text{for } i = 3, 4, \ldots, 8\\
\pi(A_2) &= P\pi(C_1)\\
\pi(A_i) &= p_{i-1}^{\mathrm{CI}} P\pi(C_{i-1}), \text{for } i = 3, 4, \ldots, 9\\
\pi(L_i) &= \left(1 - p_{i-1}^{\mathrm{CI}}\right)\pi(C_{i-1}), \text{for } i = 3, 4, \ldots, 8\\
&\pi(L_S) + \sum_{i=0}^{7}\pi(O_i) + \sum_{i=1}^{8}\pi(C_i) + \sum_{i=2}^{9}\pi(A_i) + \sum_{i=3}^{9}\pi(L_i) = 1
\end{aligned} \tag{5, 6}$$

$$\begin{aligned}
p_i^{\mathrm{NT}} &= p_{i-1}^{\mathrm{NS}} p_{i-1}^{\mathrm{CI}} P\\
&= \begin{cases}
\left(\dfrac{\pi(C_1)}{\pi(L_S) + \sum_{j=1}^{7}\pi\left(O_j\right) + \pi(C_1)}\right)P & \text{for} \quad i = 2\\
\left(\dfrac{\pi(C_{i-1})}{\pi(L_S) + \sum_{j=i-1}^{7}\pi\left(O_j\right) + \pi(C_{i-1}) + \sum_{j=2}^{i-1}A_j}\right)p_{i-1}^{\mathrm{CI}}P & \text{for } i = 3, 4, \ldots, 8\\
\left(\dfrac{\pi(C_8)}{\pi(L_S) + \pi(C_8) + \sum_{j=2}^{8}A_j}\right)p_8^{\mathrm{CI}}P & \text{for} \quad i = 9
\end{cases}
\end{aligned} \tag{7}$$

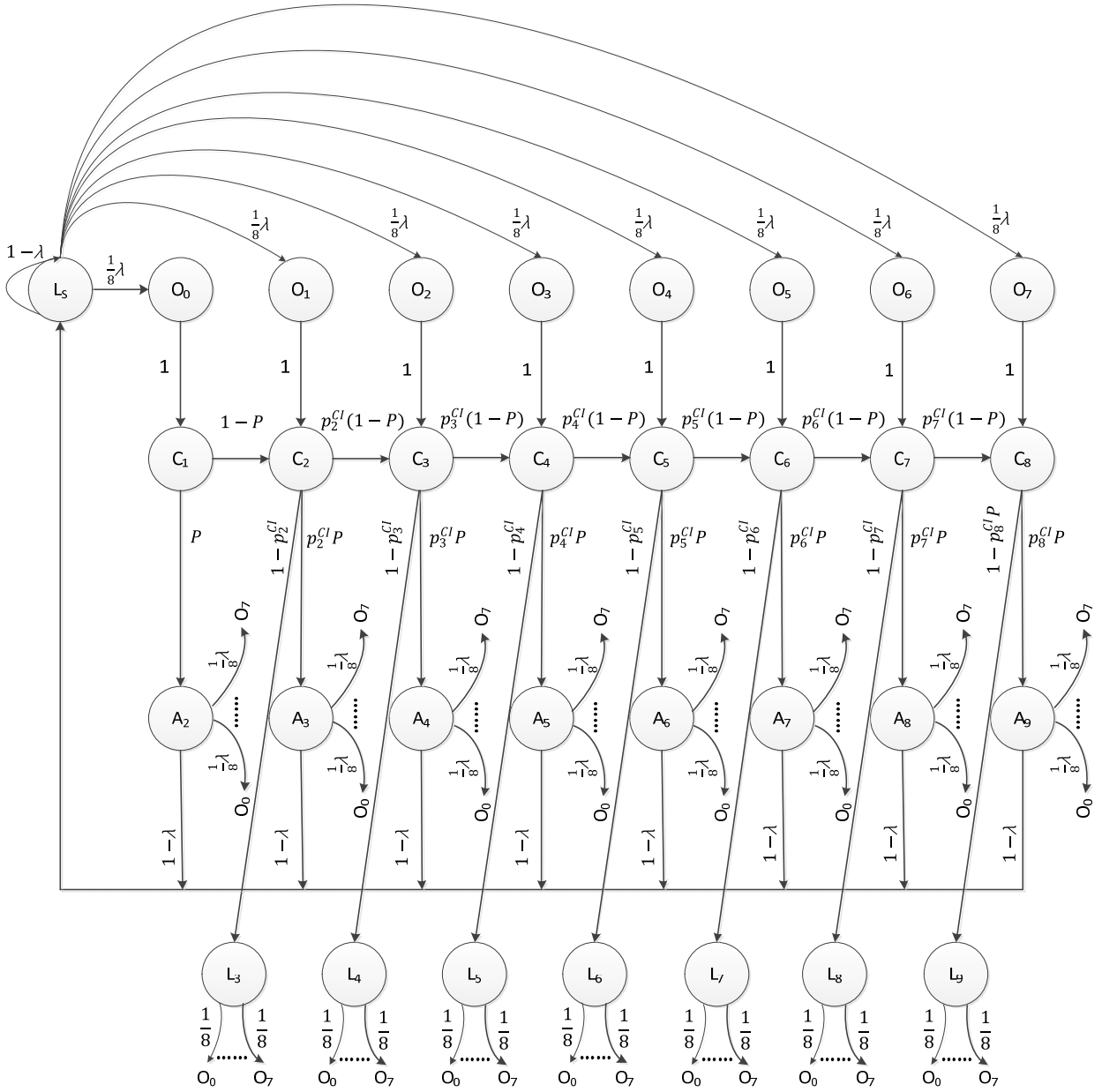

Fig. 3. Markov chain model for a contention constrained pCSMA (CCP) node.

M is the number of sensing nodes. Note that $p_{i|\mathrm{CIP}}^{\mathrm{NT}}$ is equal to $p_{i-1}^{\mathrm{NS}}P$ according to (7).

Knowing the probability α_i, $1 \le i \le 9$, that none of the nodes is transmitting in BPi given that the channel was idle in BP$i-1$, we can develop a Markov chain model for the channel states, which is illustrated in Fig. 4. There are only two possible states: 1) the channel is idle if none of the nodes transmits and 2) the channel is busy if one or more nodes are transmitting. Its steady-state occupancy can be obtained by solving its balance as follows:

$$\begin{aligned}
\pi(D_1) &= \pi(D_9) + \sum_{i=2}^{9} \pi(Y_i) \\
\pi(D_i) &= \alpha_{i-1}\pi(D_{i-1}) \qquad \text{for } i = 2, 4, \ldots, 9 \\
\pi(Y_i) &= (1-\alpha_{i-1})\pi(D_{i-1}) \quad \text{for } i = 2, 4, \ldots, 8
\end{aligned} \tag{10}$$

$$\sum_{i=1}^{9} D_i + \sum_{i=2}^{9} Y_i = 1. \tag{11}$$

The notation D_i, $1 \le i \le 9$ represents the probability that the channel is idle (i.e., none of the nodes is transmitting) in BPi except for D_9, which denotes that the channel is continuously in an idle state until the end of the current slot. Y_i, $2 \le i \le 9$ describes that the channel is continuously in a busy state (i.e., at least one node is transmitting) from BPi until the end of the current on-air packet and the rest of PTP (the unused BPs and slot guard time). The probabilities that the channel is idle in BPi ($2 \le i \le 8$) which are used in (5) now can be computed

$$p_i^{\mathrm{CI}} = \frac{\pi(D_i)}{\pi(D_i) + \sum_{j=2}^{i} Y_j} \text{ for } i = 2, 3, \ldots, 8. \tag{12}$$

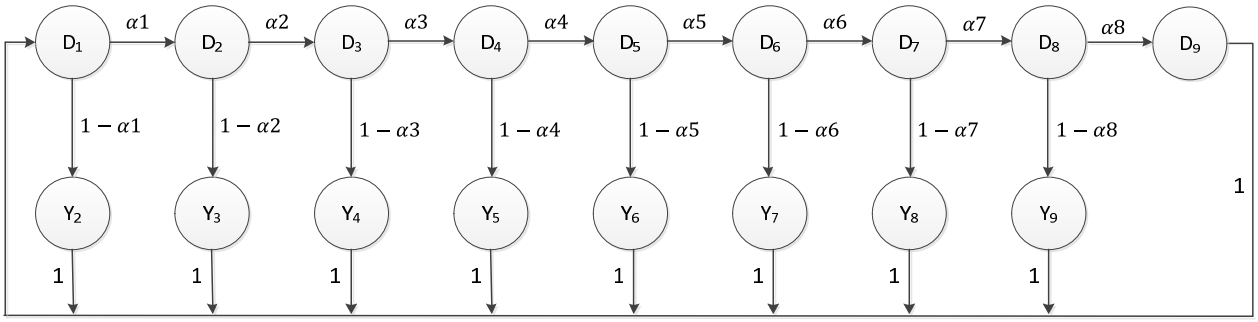

Fig. 4. Channel state model.

D. Aggregated Channel Throughput

The channel in each slot can either be used for the transmission of a packet (i.e., the channel is in $\bigcup_{i=2}^{9} Y_i$) or idle if none of the nodes is transmitting. Note the steady-state probabilities of a slot being idle and busy are equal to $\pi(D_9)$ and $\sum_{i=2}^{9}\pi(Y_i)$, respectively. The aggregated channel throughput H, which is defined as the fraction of time spent in the busy state, can be computed by

$$H = \frac{N\sum_{i=2}^{9}\pi(Y_i)}{(N+G+8)\left(\sum_{i=2}^{9}\pi(Y_i)+\pi(D_9)\right)} \tag{13}$$

where N and G denote the lengths of the packet and slot guard time in terms of the number of BPs, respectively, and the length of CAP is 8. Thus, the numerator of (13) represents the proportion of time spent in transmission excluding random backoff, unused BPs, and the slot guard time, and the denominator of (13) denotes the sum of the duration of all slots.

E. Average Energy Consumption Per Successfully Transmitted Packet

The evaluation of energy performance in simulation is based on the parameters that are measured from real-world implementations. For an end node in a LoRaWAN, it usually comprises a microcontroller unit (MCU) and LoRa modem. Thus, the energy consumption of the node depends on the selection and configuration of the MCU and the LoRa modem [43], [44]. For example, the energy dissipation of the STM32L151 [45], which is a widely used MCU for many LoRaWAN end node implementations, is lower than that of the STMG32G030 [46], but the cost of the STM32L151 is nearly double that of the STMG32G030. The receive mode current of the SX1276, which is a widespread LoRa transceiver, is 9.9 mA [34], and for the new transceiver LLCC68 [47], the receive current is reduced to 4.6 mA. We use the hardware produced by [48], which integrates an LLCC68 low-power LoRa transceiver with an STM32G030C8T6 MCU that runs at 64 MHz. The board works with a supply voltage of 3.3 V. It is a low cost and low-power consumption design and currently mass-produced in the order of tens of thousands.

To compute the average energy consumption, we need to identify the current and the average time that an end node spends in each state. Table I illustrates the current of the node in different states, which we measured from the hardware. Note that the current for sleep mode and backoff mode is equal since the end node enters sleep mode when it is in the backoff state to conserve power.

TABLE I
CURRENT OF NODE IN DIFFERENT STATES

I_{SLEEP}	I_{BO}	$I_{\text{CS}}(I_{\text{RX}})$	I_{TX}	I_{FO}
5.5 uA	5.5 uA	11 mA	26 mA	6 mA

The fraction of time that a node spends in each state is computed from

$$p^{\text{CS}} = \frac{\sum_{i=1}^{8}\pi(C_i)}{p^{\text{OT}}} \tag{14}$$

$$p^{\text{TX}} = \frac{\sum_{i=2}^{9}N\pi(A_i)}{p^{\text{OT}}} \tag{15}$$

$$p^{\text{OT}} = S\pi(L_S) + \sum_{i=0}^{7} i\pi(O_i) + \sum_{i=3}^{9}(S-i+1)\pi(L_i) + \sum_{i=2}^{9}(S-i+1)\pi(A_i) \tag{16}$$

$$p^{\text{SLEEP}} = 1 - p^{\text{CS}} - p^{\text{TX}} \tag{17}$$

where p^{CS} and p^{TX} represent the time fractions that a node is in the carrier sense and transmitting states, respectively, and the fraction of time p^{SLEEP} that a node spends in sleep includes the states L_S, O_i, and L_i. Note that in order to obtain the steady-state proportion of the time that a node stays in a certain state, the dwell time that the node spends in each state should be considered [49]. The transmission duration of a packet is N BPs, thus in the numerator of (15) for computation of p^{TX}, the steady-state probability is multiplied by N. The dwell time in the carrier sensing states C_i is 1 BP since it takes 1 BP for the node to sense the channel. In the L_S state, the dwell time is S BPs where S denotes the entire slot duration. When the node is in O_i state, it remains in sleep mode until BPi; thus, the dwell time in this state is i BPs. Once the node jumps either to a L_i or a A_i state, it continuously remains in that state until the end of the current slot. Therefore, the dwell time in both L_i, $3 \le i \le 9$ and A_i, $2 \le i \le 9$ states, which include unused BPs and slot guard time in PTP, is $(S-i+1)$ BPs. The notation p^{OT} represents the sum of the duration of

all slots (defined as the entire network operation time). Note that in the last term in (16), the dwell time is not N BPs but $(S-i+1)$ BPs, because in order to compute the entire network operation time, not only the transmission time (i.e., N BPs), but also the unused BPs and guard time in PTP which are always appended to the end of the transmission should be accounted for.

We are now in a position to compute the average energy consumption per successfully transmitted packet, E_{av}

$$E_{\text{av}} = \frac{\text{UI}_{\text{SLEEP}} p^{\text{SLEEP}} \text{RB} + \text{UI}_{\text{CS}} p^{\text{CS}} \text{RB} + \text{UI}_{\text{TX}} p^{\text{TX}} \text{RB}}{\frac{\text{RH}}{\text{NM}}} = \frac{\text{NMBU}(I_{\text{SLEEP}} + I_{\text{CS}} + I_{\text{TX}})}{H} \tag{18}$$

where U represents the hardware board supply voltage (i.e., 3.3 V), R denotes the network running time in terms of the number of BPs, and B represents the time duration of a BP. Knowing the network throughput H, the total network busy time for packet transmissions is RH. As a result, the number of packets that are successfully transmitted per node is (RH/NM).

Note that there is a small difference in energy consumption between unconfirmed and confirmed messaging as in the latter case, end nodes need to spend some time to receive the acknowledgment, and the power for transmission to reception turn-around activity and receiving is smaller than for transmission. However, we show that this difference is constant. The energy consumption for unconfirmed packet transmission is (NBP_{TX}), where P_{TX} is the transmission power. In contrast, the total energy consumption for confirmed packet transmission, transceiver mode turn-around, and the subsequent acknowledgment reception is $[(N_C B)P_{\text{TX}} + (1B)P_{\text{FO}} + (17B)\ P_{\text{RX}}]$, where N_C and P_{FO} represent the confirmed packet length and the power of the node when it is fully operational but without any radio activity, respectively. P_{RX} is the reception power of the node which lasts for 17 BPs. Using the unconfirmed and confirmed packet lengths, 152 BP and 134 BP, the fraction of these two energy consumption values, F_{UC} is given by

$$F_{\text{UC}} = \frac{152\text{BP}_{\text{TX}}}{134\text{BP}_{\text{TX}} + 1\text{BP}_{\text{FO}} + 17\text{BP}_{\text{RX}}} = \frac{152 I_{\text{TX}}}{134 I_{\text{TX}} + 1 I_{\text{FO}} + 17 I_{\text{RX}}} = 1.07 \tag{19}$$

where $I_{\text{TX}}, I_{\text{FO}}$, and I_{RX} are the currents when the node is transmitting, fully operational, and receiving, respectively, and their values are given in Table I. According to (19), the energy consumption of unconfirmed transmission is constantly 7% higher than that of confirmed transmission including acknowledgment reception under our CCP protocol due to the fact that the packet length of an unconfirmed message is 18 BPs longer than that of a confirmed message. Therefore, it is sufficient to only analyze the unconfirmed messaging energy consumption and the performance of confirmed messaging can be deduced accordingly since it conforms to the same pattern as unconfirmed messaging.

V. Simulations and Discussion

We have verified our analytical modeling by simulations for different model parameters like the traffic rate λ, persistence value P, and nodes number M. The approximations described in Section IV were only meant to simplify the analysis and they have not been implemented in the simulation. The results in this section are also used to verify the validity of these approximations. Furthermore, classic p-CSMA, which is considered as the state-of-the-art technique, can be adopted for the LoRaWAN MAC layer, and its simulated performance is used as a benchmark to evaluate the performance of our CCP protocol.

A. Throughput Performance

In our first set of simulations, we investigated the throughput performance. Figs. 5–7 plot the channel throughput over the packet arrival probability λ in each slot duration for persistence values of 0.01, 0.1, and 1, when the node number is set to 100, 200, and 500, respectively. Furthermore, the throughput performance of ALOHA with the consideration of packet capture effect introduced by the LoRa is derived in a way described in [50] and is also plotted in these three figures. It is notable that our analysis model is very accurate in all three cases, as the analytical result matches the simulation results very well. This justifies the analytical model assumptions described in Section IV for all considered network sizes. Another notable general observation from Figs. 5–7 is that the maximum throughputs of classic p-CSMA and CCP are comparable and higher than two times that of ALOHA (0.36). Such enhancement remains once $\lambda \geq 0.012$.

In case of $P = 0.01$ and $M = 100$ as illustrated in Fig. 5(a), the throughput of p-CSMA keeps increasing with a constant slope of 1 when $\lambda \leq 0.012$. If we compute the offered load, which we define as the ratio of the total number of packets offered to the network to its capacity (i.e., the maximum number of packets that can be transmitted without any collisions), we find that the offered load reaches 1 at $\lambda = 0.012$ (i.e., all packets arriving at the nodes can be successfully transmitted). After that, additional packets are lost which leads to a rapid saturation at a level of 0.96 once λ is beyond 0.02. In comparison with p-CSMA, the throughput of CCP increases with a smaller slope until $\lambda = 0.02$ where the offered load reaches 1. After that, CCP starts to saturate, and the explanation for the significantly slower saturation is that when $P = 0.01$, a node strongly tends to defer its transmission even if the channel is sensed idle. As a result, although more packets arrive in the case of higher λ, an increasing portion of the packets are discarded by CCP due to its buffering of the current packet. p-CSMA, however, transmits packets in a more aggressive manner and consequently, more packets are accepted from the upper layer compared to CCP, which leads to a much more rapid saturation. Apart from the slower saturation, the maximum throughput of CCP is slightly lower (0.9 instead of 0.96) than that of p-CSMA due to the overhead of the CAP (8 BPs) and the guard time (2 BPs) in each slot of CCP which accounts for a fraction of $(8+2)/162 = 0.062$ on the throughput. Note that there is a small difference between simulation and computation for $\lambda \geq 0.01$. The explanation for this is that if the

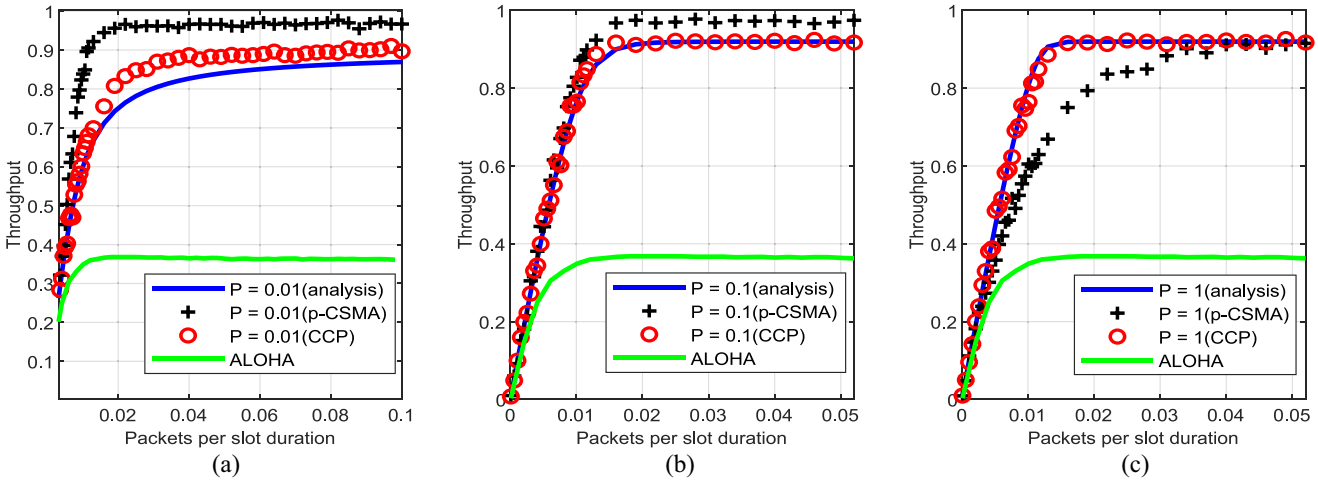

Fig. 5. Throughput performance as a function of packet arrival rate per slot duration for $M = 100$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

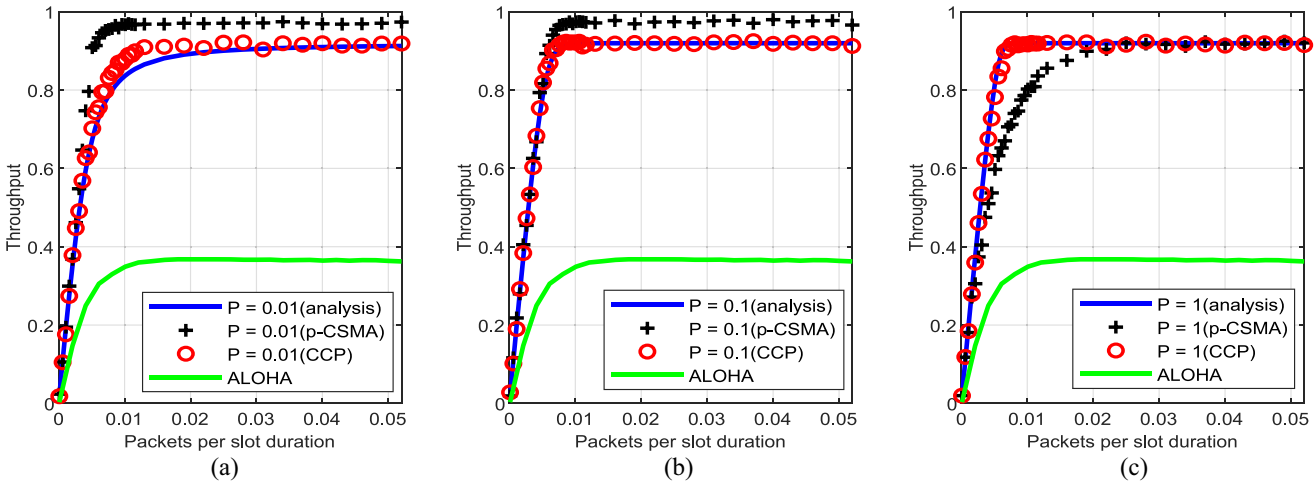

Fig. 6. Throughput performance as a function of packet arrival rate per slot duration for $M = 200$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

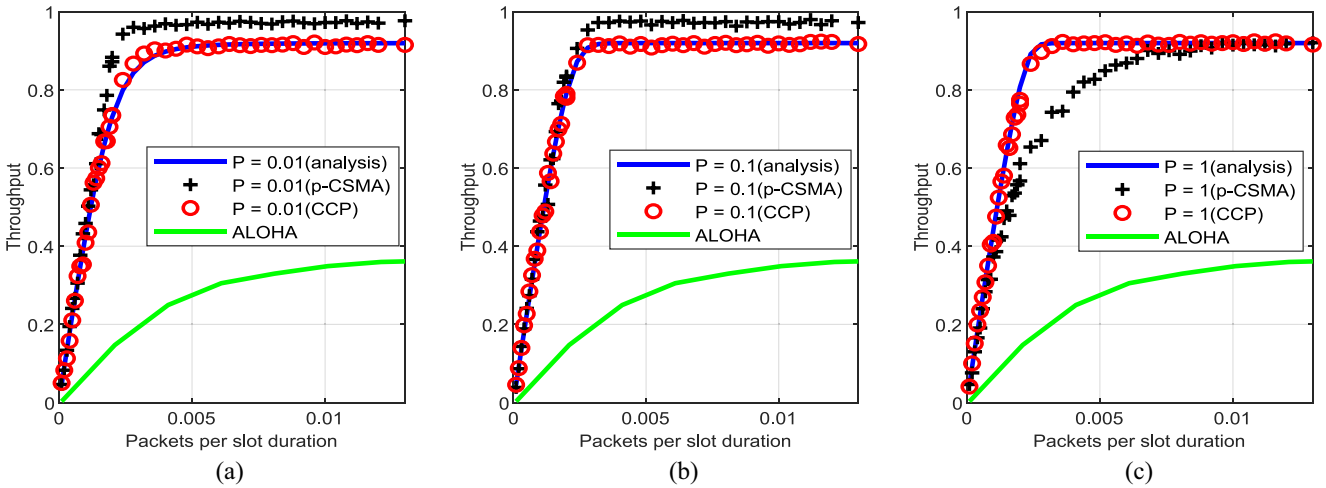

Fig. 7. Throughput performance as a function of packet arrival rate per slot duration for $M = 500$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

persistence value P is very small (i.e., 0.01 in our case), the pseudo random number generator gave slightly wrong results. As a result, the probability to produce a number smaller than 0.01 was lower than it was supposed to be. At higher values of P, this is not an issue as can be seen from Fig. 5(b) and (c). The throughput performance of CCP in the case of $P = 0.1$

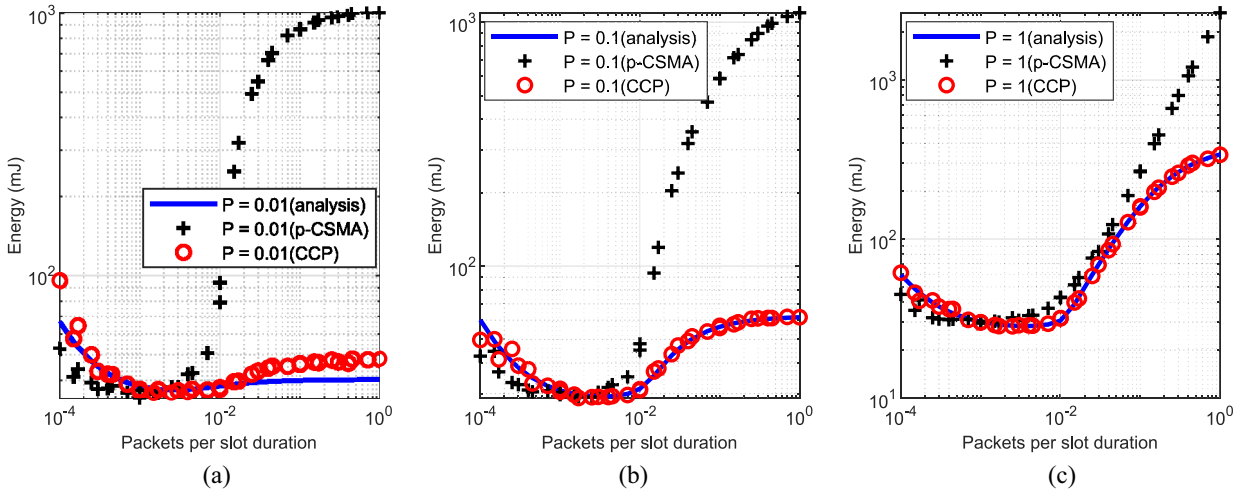

Fig. 8. Average energy per successfully transmitted packet for $M = 100$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

and 1 in Fig. 5(b) and (c) is quite similar to $P = 0.01$ in Fig. 5(a). Both of them saturate at the same throughput level of 0.96 as the case of $P = 0.01$ but saturation appears already at $\lambda = 0.016$ which is approximately six times lower. The explanation for the rapid saturation is that the higher persistent value makes the node transmit a packet with a higher probability when the channel is sensed free and the node can attempt to transmit more packets, which results in a higher throughput when λ is low. The performance of p-CSMA for $P = 1$ and $\lambda \leq 0.04$ [as illustrated in Fig. 5(c)] is worse as compared to $P = 0.1$ and 0.01. This is caused by the definitive transmission approach when the node senses the channel idle, which leads to more collisions on the channel although the offered load is high.

For both medium and large network sizes ($M = 200$ and 500), p-CSMA and CCP throughput saturate at the same level as they achieved in the small network ($M = 100$), but at nearly half the value of λ (0.01 and 0.005 in Figs. 6 and 7, respectively) since the offered load is much higher when more nodes are attempting to transmit. We also compute the offered load where the slope starts to decrease drastically for p-CSMA ($\lambda \approx 0.006$ and 0.002 as shown in Figs. 6 and 7, respectively) and we again find that it reaches at 1 at these two λ values. Due to the same reason as explained for Fig. 5, most of the additional packets are discarded when the offered load is beyond 1, which subsequently, leads to rapid saturation. As a comparison, CCP increases with a similar slope with p-CSMA until its offered load reaches 1 at $\lambda \approx 0.008$, 0.005, and 0.002 in Figs. 6(a)–(c) and 7, respectively.

In the last evaluation of the throughput performance, we consider the particular scenario when both the packet arrival rate and the number of nodes are low. It is conceivable that in many time slots, no transmission happens when very few nodes attempt to transmit with a probabilistic transmission approach. Table II illustrates the simulation results of throughput performance with various settings of low packet arrival rate per slot (λ) and their corresponding offered loads. To assess the negative impact of low persistent value and node number, these two parameters are set to 0.01 and 10, respectively. It is notable that nearly all offered traffic is transmitted successfully by CCP (i.e., offered load $\approx$ throughput) which indicates that CCP can perform adequately even with low persistent values and transmission attempts.

TABLE II
CCP THROUGHPUT UNDER VERY LOW OFFERED LOAD
($P = 0.01$, $M = 10$, SIMULATION)

$\lambda(\times 10^{-2})$	0.01	0.41	0.81	1.01	1.21
Offered load $(\times 10^{-2})$	0.08	3.5	5.55	7.83	9.19
Throughput $(\times 10^{-2})$	0.08	3.42	5.32	7.22	8.74

In summary, the throughput performance of CCP is only slightly lower than that of p-CSMA in the case of $P = 0.01$ and 0.1. However, CCP outperforms p-CSMA in the case of $P = 1$. Moreover, the results show that in small networks using CCP P should be chosen as 0.1, as it outperforms the setting with $P = 0.01$. Finally, CCP can also handle the scenarios with very low offered loads and few nodes.

B. Energy Performance

For the energy performance evaluations, our analytical model again obtained results that perfectly match the simulation results, as illustrated in Figs. 8–10. The issue with the random number generator again caused small differences between simulation and computation as shown in Fig. 8(a).

The average energy consumption per successfully transmitted packet declines first in all figures for both p-CSMA and CCP when λ is below a certain value. Then, the energy consumption of CCP and p-CSMA increases with λ either gradually or drastically. The explanation for such behavior is that the energy of the nodes is mostly consumed by their sleep state compared to the very limited number of transmissions and carrier sensing events at low values of λ, while the number of successfully transmitted packets increases linearly with λ. Thus, the average energy consumption per successfully transmitted packet decreases at low λ. Beyond a certain value of λ [e.g., $\lambda = 0.0025$ in the case of $P = 0.01$ and $M = 100$ illustrated in Fig. 8(a)], packet transmissions and

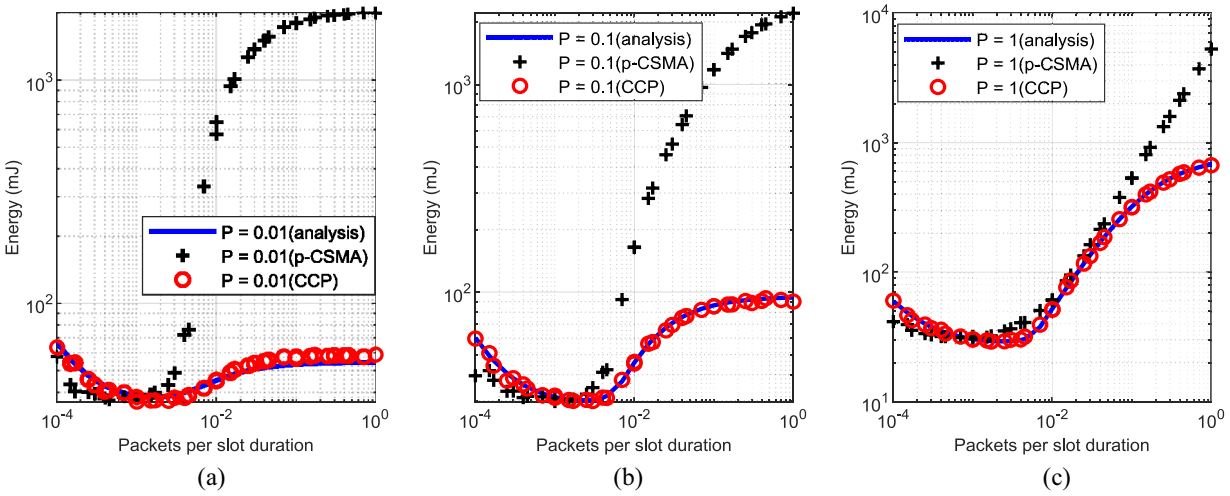

Fig. 9. Average energy per successfully transmitted packet for $M = 200$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

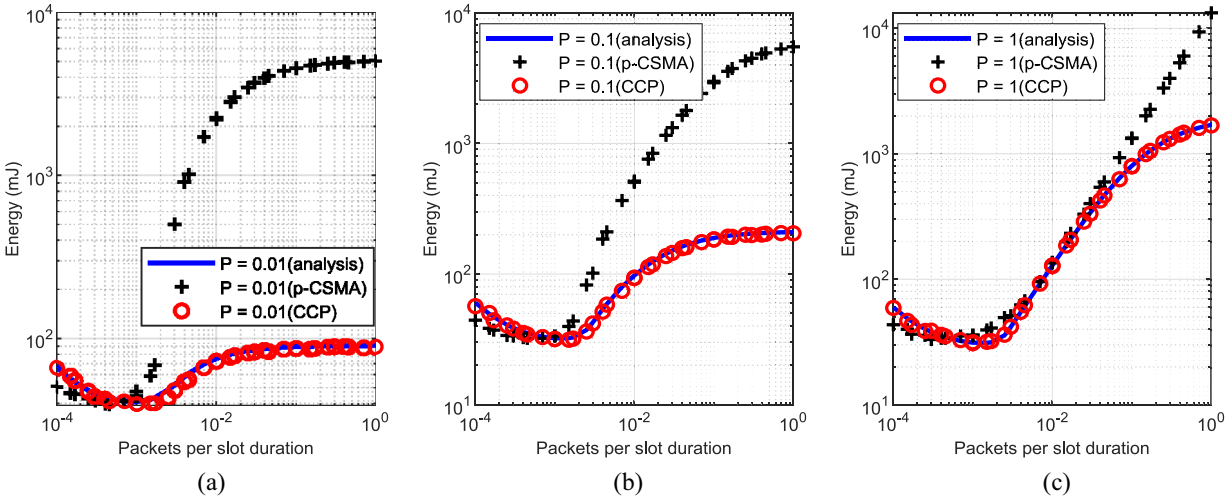

Fig. 10. Average energy per successfully transmitted packet for $M = 500$ nodes. The persistence values for the analysis and simulation of CCP, and the classic p-CSMA are set to 0.01, 0.1, and 1, respectively. The results are depicted in the subparts of (a), (b), and (c) accordingly.

the associated carrier sense activities come to dominate the energy consumption as the number of transmissions and carrier senses rise more rapidly than the number of packets successfully transmitted due to the throughput limitation. The energy consumption of CCP saturates at various but limited values (in a range of 30–1700 mJ), which increase with the network size and persistent value as more nodes attempt to transmit or use the channel in a more aggressive manner. Note that in all figures, CCP with $P = 0.01$ and 0.1 saturates once $\lambda \geq 0.2$. This is because the channel is almost completely occupied at $\lambda = 0.2$. Furthermore, nodes with low persistent values tend to defer their transmissions even if they sense the channel as idle, which leads to an approximately constant number of transmissions and carrier sense activities. It follows that the energy consumption remains almost constant even if λ increases. If $P = 1$ is set for CCP, packets are transmitted in a more aggressive manner and the number of accepted and transmitted packets increases with λ. This causes a rise in energy consumption. Different from CCP, p-CSMA keeps increasing its energy consumption after its minimum in all scenarios as the number of packets that are successfully transmitted increases when λ is below a certain value (fewer nodes attempt transmission and the packet collision is low). After that value, nodes attempt the transmission more frequently when λ is rising, which results in increasing packet collisions and average energy consumption per successful transmission. More specifically, p-CSMA energy consumption rises to 1020 mJ as shown in Fig. 8(a) and 13 300 mJ as shown in Fig. 10(c), respectively.

In summary, the energy performance of CCP is far better than that of p-CSMA for all persistent values and network sizes. Therefore, if low power consumption is a critical requirement, p-CSMA is not suitable and CCP with $P = 0.01$ or 0.1 should be used. Furthermore, if the network size is around 100, $P = 0.1$ should be used for maximum throughput. When the network size becomes larger than 200, $P = 0.01$ should be chosen for CCP as the energy consumption stays below 90 mJ.

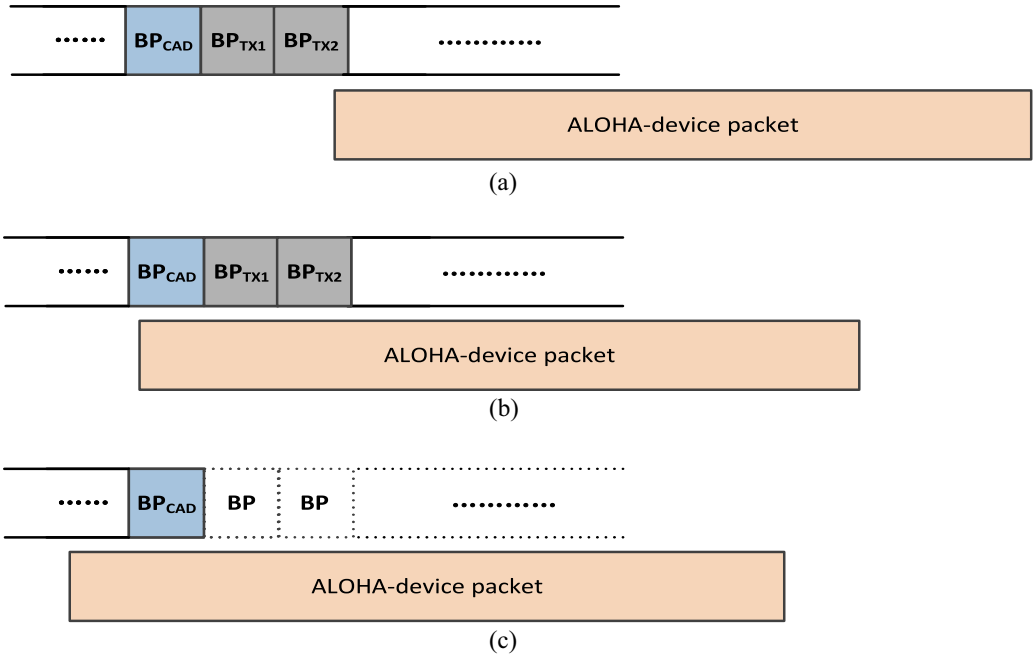

Fig. 11. All possible timings in case CCP and other-MAC (e.g., ALOHA) devices simultaneously attempt to transmit a packet. The subparts (a), (b), and (c) illustrate the scenarios that an ALOHA-device starts to transmit a packet later than, overlapped with, and before the CAD BP of CCP device, respectively.

C. Impact of Interference From Other LoRa MAC Protocols

This evaluation is performed on the behavior of CCP regarding signal interference caused by LoRa devices conforming to other MAC protocols (we denote these protocols as other-MAC). In LoRa networks, there are two factors which are relevant for the effect of this type of interference: transmission starting time difference between LoRa signals and RSS. Note that the starting of CCP packet transmission will be only before or slightly later (less than 1 BP due to CCP performing CAD prior to transmission) than other-MAC devices. Fig. 11 illustrates all possible timing scenarios in case a CCP packet is transmitted simultaneously with a packet of an other-MAC device (here ALOHA is assumed). If the ALOHA device starts sufficiently late [as depicted in Fig. 11(a) where the CCP device accomplishes CAD with no symbol detected], both packets are transmitted and a collision occurs. A collision also happens if the ALOHA device starts slightly later than the start of CAD of the CCP device [as shown in Fig. 11(b)]. Because, to successfully detect the signals of the ALOHA device, the CCP device needs to capture at least 2 symbols (1 BP). If the ALOHA device starts even earlier [Fig. 11(c)], the CCP device can detect the presence of the signal of the ALOHA-device, and will refrain from transmission. Besides transmission timings, we also need to discuss the RSS of the received packets. There are three cases of relations between RSS of CCP (denoted by RSS_C) and RSS of other-MAC (denoted by RSS_o).

1) $\mathrm{RSS}_C > \mathrm{RSS}_o$.
2) $\mathrm{RSS}_C = \mathrm{RSS}_o$.
3) $\mathrm{RSS}_C < \mathrm{RSS}_o$.

In the first case, CCP packets will be successfully received according to the experimental results presented in [23]. In the second case where the RSSs are equal (the case studied in [40]), CCP packets is corrupted only when $t_C < t_o$ and $t_o - t_C \leq 3T_{\mathrm{symbol}}$, where t_C and t_o represent the transmission starting time of CCP and other-MAC, respectively. Note that the duration of CCP packets is as long as $304T_{\mathrm{symbol}}$. Therefore, the vulnerable period of CCP packets ($3T_{\mathrm{symbol}}$) accounts for less than 1%, which is negligible. In the last case, the CCP packet which is received with weaker signal strength has only a probability of 0.13 that it can still be correctly demodulated [23]. In summary, roughly speaking, for the first two cases, CCP will be successful with a probability near 1.

D. Network Performance Under Other Settings

To extend the application range of our CCP protocol, we also simulated the network performance under three categories of different settings: higher data rate (SF7 [51]), shortened packet length (20–180 bytes), and more allowed retransmissions attempts (up to 2 and 7). Fig. 12(a) illustrates the network throughput when shorter packets are transmitted at a higher data rate using a SF of 7. Apparently, a higher data rate does not affect the throughput performance. The highest throughput achieved with a packet length of 180 bytes still remains at 0.92 which is comparable with SF9. In contrast, shortening the packet length lowers the saturation value of the throughput. This is due to the fact that the PTP duration is proportional to the packet length while the CAP duration is constant, which leads to a higher overhead for short packets. The impact of different allowed retransmission attempts is depicted in Fig. 12(b). If up to two retransmission attempts are allowed, a maximum of additional 7 mJ of energy is required for those retransmissions at low λ. At high values of λ the energy consumption is almost independent from the allowed

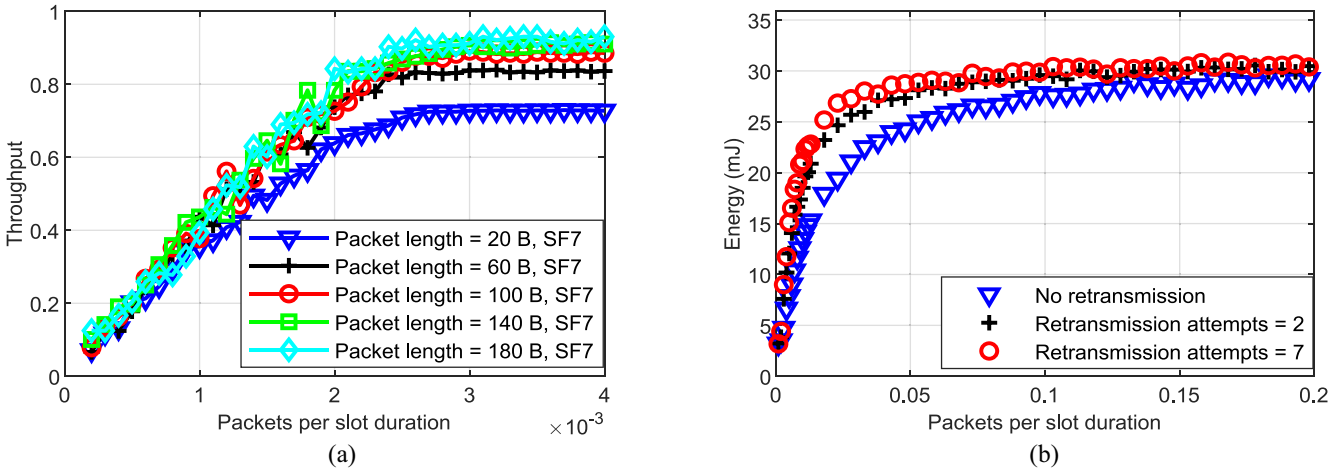

Fig. 12. (a) Throughput performance with different packet lengths. SF = 7, $M = 500$ nodes, and $W = 125$ kHz, and no retransmission. (b) Energy performance versus retransmission attempts. SF = 7, $M = 500$ nodes, $W = 125$ kHz, and *Packet length* = 20 bytes. The transmission power was set to 10 dBm.

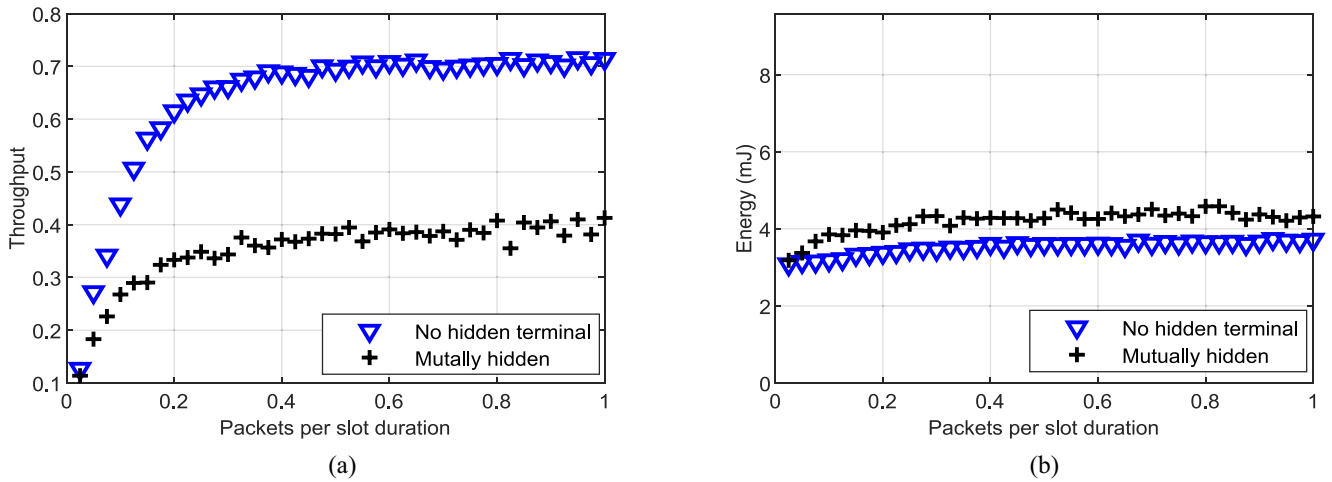

Fig. 13. Throughput and average energy per successfully transmitted packet performance (depicted in (a) and (b), respectively) with and without the hidden terminal problem. SF = 7, $M = 8$ nodes, $W = 125$ kHz, *Packet length* = 20 bytes, and retransmission attempts is 2. The transmission power is set to 10 dBm.

number of retransmission attempts. Furthermore, if retransmissions are allowed the difference between 2 and 7 allowed retransmissions is negligible, as most packets are received successfully after the 3rd transmission.

E. Hidden Terminal Problem

The hidden terminal problem is a common challenge for distributed MAC protocols. We set up a simulation scenario where any node cannot sense the signal of its neighbor but can reach the gateway. Eight end nodes and the gateway are placed on the vertices and in the center of a cube-shaped space, respectively. In comparison, we also simulated a converse scenario where the distances between the nodes are small enough so that end nodes can sense each others' signals. Fig. 13 illustrates the throughput and energy performance comparison. Note that all transmission attempts (including the first one and the following retransmission attempts if allowed) for a packet are consecutive. The throughput is reduced by 43% if the hidden terminal problem occurs. This is because much more packet transmissions in hidden terminal scenario are concentrated in a portion of entire slots, which caused numerous vacant slots (i.e., channel underutilization). In contrast, the negative impact of hidden terminal problem on the energy consumption performance is limited to 20%. This is due to the factor that in the hidden terminal scenario, though nodes cannot sense the signals of the other nodes, the number of collided transmissions is still highly bounded due to the low persistent probability (0.1).

VI. Conclusion

In this article, we have analyzed the issues of LoRaWAN resulting from the ALOHA protocol which is regularly used. A novel MAC protocol called CCP, which is based on time-slotted and constrained channel access attempt, has been proposed to address these issues. Markov chains have been applied to model the behavior of the CCP protocol for which we have used three assumptions to simplify our analysis. Simulation results have been obtained, which verified that these assumptions are valid and our analysis model is able

to accurately compute the network throughput and the energy consumption of the CCP protocol in the cases of various persistent values and network sizes. Furthermore, both analysis and simulation results have shown that optimal performance in terms of network throughput and energy consumption could be simultaneously achieved by our CCP protocol with $P = 0.01$ for medium and large networks and $P = 0.1$ for a small network. Finally, the impact of higher data rate, variable packet length, allowed retransmission attempts, and the hidden terminal problem on the network performance were investigated in simulations. A clear degradation of the throughput was observed in case of the hidden terminal scenario, while the proposed CCP retained a high level of performance in the rest of scenarios.

REFERENCES

[1] E. Kunz. "Low power wide area network (LPWAN) market to reach over USD 675.75 billion by 2028." Feb. 2022. [Online]. Available: https://www.globenewswire.com/news-release/2022/02/09/2381562/0/en/Low-Power-Wide-Area-Network-LPWAN-Market-to-Reach-Over-USD-675-75-Billion-by-2028-Low-Power-Wide-Area-Network-Vantage-Market-Research.html

[2] J. Haxhibeqiri, E. D. Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, 2018.

[3] N. Varsier and J. Schwoerer, "Capacity limits of LoRaWAN technology for smart metering applications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2017, pp. 1–6.

[4] Y. Lalle, L. C. Fourati, M. Fourati, and J. P. Barraca, "A comparative study of LoRaWAN, SigFox, and NB-IoT for smart water grid," in *Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS)*, 2019, pp. 1–6.

[5] L. Leonardi, F. Battaglia, G. Patti, and L. L. Bello, "Industrial LoRa: A novel medium access strategy for LoRa in industry 4.0 applications," in *Proc. IECON 44th Annu. Conf. IEEE Ind. Electron. Soc.*, 2018, pp. 4141–4146.

[6] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3097–3109, Apr. 2019.

[7] D. Zorbas, K. Abdelfadeel, P. Kotzanikolaou, and D. Pesch, "TS-LoRa: Time-slotted LoRaWAN for the industrial Internet of Things," *Comput. Commun.*, vol. 153, pp. 1–10, Mar. 2020.

[8] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni, and M. Gidlund, "Using LoRa for industrial wireless networks," in *Proc. IEEE 13th Int. Workshop Factory Commun. Syst. (WFCS)*, 2017, pp. 1–4.

[9] A. Shaker, "A survey of smart buildings and homes using low-power wide-area network (LoRa WAN)," in *Proc. 4th Int. Symp. Multidiscipl. Stud. Innov. Technol. (ISMSIT)*, 2020, pp. 1–7.

[10] P. Boccadoro, B. Montaruli, and L. A. Grieco, "QuakeSense, a LoRa-compliant earthquake monitoring open system," in *Proc. IEEE/ACM 23rd Int. Symp. Distrib. Simulation Real Time Appl. (DS-RT)*, Oct. 2019, pp. 1–8.

[11] G. Hristov, J. Raychev, D. Kinaneva, and P. Zahariev, "Emerging methods for early detection of forest fires using unmanned aerial vehicles and LoRaWAN sensor networks," in *Proc. 28th EAEEIE Annu. Conf. (EAEEIE)*, 2018, pp. 1–9.

[12] E. A. Kadir, A. Efendi, and S. L. Rosa, "Application of LoRa WAN sensor and IoT for environmental monitoring in Riau Province Indonesia," in *Proc. 5th Int. Conf. Elect. Eng. Comput. Sci. Inform. (EECSI)*, 2018, pp. 281–285.

[13] B. Citoni, F. Fioranelli, M. A. Imran, and Q. H. Abbasi, "Internet of Things and LoRaWAN-enabled future smart farming," *IEEE Internet Things Mag.*, vol. 2, no. 4, pp. 14–19, Dec. 2019.

[14] J. P. S. Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, 1st Quart., 2020.

[15] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 535–547, Mar. 2000.

[16] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, "Throughput analysis in multihop CSMA packet radio networks," *IEEE Trans. Commun.*, vol. 35, no. 3, pp. 267–274, Mar. 1987.

[17] K. Mikhaylov, J. Petaejaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *Proc. Eur. Wireless 22th Eur. Wireless Conf.*, 2016, pp. 1–6.

[18] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient carrier-sense multiple access for LoRa," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13.

[19] O. Georgiou and U. Raza, "Low power wide area network analysis: Can LoRa scale?" *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 162–165, Apr. 2017.

[20] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, p. 1466, Sep. 2016.

[21] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, Sep. 2017.

[22] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.

[23] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" in *Proc. 19th ACM Int. Conf. Model. Anal. Simul. Wireless Mobile Syst.*, 2016, pp. 59–67.

[24] T. Polonelli, D. Brunelli, A. Marzocchi, and L. Benini, "Slotted ALOHA on LoRaWAN-design, analysis, and deployment," *Sensors*, vol. 19, no. 4, p. 838, 2019.

[25] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *Proc. IEEE PIMRC*, Montreal, Canada, Oct. 2017, pp. 1–6.

[26] N. Kouvelas, V. Rao, and R. Prasad, "Employing p-CSMA on a LoRa network simulator," 2018, *arXiv:1805.12263*.

[27] H. Takagi and L. Kleinrock, "Throughput analysis for persistent CSMA systems," *IEEE Trans. Commun.*, vol. 33, no. 7, pp. 627–638, Jul. 1985.

[28] N. Kouvelas, R. V. Prasad, and N. Yazdani, and D. E. Lucani, "np-CECADA: Enhancing ubiquitous connectivity of LoRa networks," in *Proc. IEEE 18th Int. Conf. Mobile Ad Hoc Smart Syst. (MASS)*, Oct. 2021, pp. 374–382.

[29] K. Q. Abdelfadeel, D. Zorbas, V. Cionca, and D. Pesch, "*FREE*—Fine-grained scheduling for reliable and energy-efficient data collection in LoRaWAN," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 669–683, Jan. 2020.

[30] Z. Xu, J. Luo, Z. Yin, T. He, and F. Dong, "S-MAC: Achieving high scalability via adaptive scheduling in LPWAN," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Jul. 2020, pp. 506–515.

[31] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018.

[32] S. Kim, H. Lee, and S. Jeon, "An adaptive spreading factor selection scheme for a single channel LoRa modem," *Sensors*, vol. 20, no. 4, p. 1008, 2020.

[33] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, "Throughput, coverage and scalability of LoRa LPWAN for Internet of Things," in *Proc. IEEE/ACM 26th Int. Symp. Qual. Service (IWQoS)*, 2018, pp. 1–10.

[34] "Semtech SX1276." Semtech. 2021. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-core/sx1276

[35] C. Zhong and A. Springer, "Design and evaluation of innovative protocols for LoRa," *IET Wireless Sens. Syst.*, vol. 12, no. 1, pp. 12–20, 2022.

[36] L. Tessaro, C. Raffaldi, and M. Rossi, and D. Brunelli, "Lightweight synchronization algorithm with self-calibration for industrial LORA sensor networks," in *Proc. Workshop Metrol. Ind. 4.0 IoT*, 2018, pp. 259–263.

[37] T. Polonelli and D. Brunelli, and L. Benini, "Slotted ALOHA overlay on LoRaWAN—A distributed synchronization approach," in *Proc. IEEE 16th Int. Conf. Embedded Ubiquitous Comput. (EUC)*, 2018, pp. 129–132.

[38] C. Zhong and A. Springer, "A novel network architecture and MAC protocol for confirmed traffic in LoRaWAN," *IEEE Access*, vol. 9, pp. 165145–165153, 2021.

[39] K. Leentvaar and J. Flint, "The capture effect in FM receivers," *IEEE Trans. Commun.*, vol. COM-24, no. 5, pp. 531–539, May 1976.

[40] C. Pham and M. Ehsan, "Dense deployment of LoRa networks: Expectations and limits of channel activity detection and capture effect for radio channel access," *Sensors*, vol. 21, no. 3, p. 825, 2021.

[41] I. Ramachandran, A. K. Das, and S. Roy, "Analysis of the contention access period of IEEE 802.15.4 MAC," *ACM Trans. Sens. Netw.*, vol. 3, no. 1, p. 4, 2007.

[42] *IEEE 802.15.4 2003. Part 15.4: Wireless Medium Access Control and Physical Layer Specifications for Low Rate Wireless Personal Area Networks*, ANSI/IEEE Standard 802.15.4, Sep. 2003.
[43] G. Tresca, F. Vista, and P. Boccadoro, "Experimenting LoRa-compliant solutions in real-world scenarios," *Internet Technol. Lett.*, vol. 3, no. 2, p. e136, 2020.
[44] M. Pulpito, P. Fornarelli, C. Pomo, P. Boccadoro, and L. Grieco, "On fast prototyping LoRaWAN: A cheap and open platform for daily experiments," *IET Wireless Sens. Syst.*, vol. 8, no. 5, pp. 237–245, 2018.
[45] "STM32L151/152." STMicroelectronics. 2021. [Online]. Available: https://www.st.com/content/st_com/en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-ultra-low-power-mcus/stm32l1-series/stm32l151-152.html
[46] "STM32G030." STMicroelectronics. 2021. [Online]. Available: https://www.st.com/content/st_com/%\en/products/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus/stm32-mainstream-mcus/stm32g0-series/stm32g0x0-value-line/stm32g030f6.html
[47] "Semtech LLCC68." Semtech. 2021. [Online]. Available: https://www.semtech.com/products/wireless-rf/lora-core/llcc68
[48] X. Nie. "LoRa module." 2021. [Online]. Available: http://www.ywevoer.com/products.html
[49] S. Ross, *Introduction to Probability Models*. London, U.K.: Academic, 2014.
[50] A. S. Tanenbaum, *Computer Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, 2003.
[51] B. Ghena, J. Adkins, and L. Shangguan, K. Jamieson, P. Levis, and P. Dutta, "Challenge: Unlicensed LPWANs are not yet the path to ubiquitous connectivity," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–12.

Chen Zhong received the B.S. degree in automation from Zhejiang University, Hangzhou, China, in 2006, and the M.Sc. degree in communication electronics from the Technical University of Munich, Munich, Germany in 2008. He is currently pursuing the Ph.D. degree in communication electronics with Johannes Kepler University Linz (JKU), Linz, Austria.

He joined the Institute for Communications Engineering and RF-Systems, JKU, at the beginning of 2020. His main research areas is low-power wide-area network, network protocols, and architectures.

Andreas Springer (Member, IEEE) received the Dipl.-Ing. degree in electrical engineering from the Technical University of Vienna, Vienna, Austria, in 1991, and the Dr.techn. (Ph.D.) and Univ.-Doz. (Habilitation) degrees from the Johannes Kepler University Linz (JKU), Linz, Austria, in 1996 and 2001, respectively.

He was with the Microelectronics Institute, JKU, from 1991 to 1996, where he joined the Institute for Communications and Information Engineering in 1997, and became a Full Professor in 2005. He has been the Head of the Institute for Communications Engineering and RF-Systems (formerly, Institute for Communications and Information Engineering), JKU, since July 2002, the Co-Leader of the Christian Doppler Lab for Digitally Assisted RF Transceivers for Future Mobile Communications since 2017, and serves as a Research Area Coordinator for the Austrian K2 Center for Symbiotic Mechatronics. His current research interests are focused on wireless communication systems, architectures and algorithms for multiband/multimode transceivers, wireless sensor networks, and recently molecular communications. In these fields, he has published more than 280 papers in journals and at international conferences, one book, and two book chapters.

Dr. Springer was a co-recipient of the Science Prize of the German Aerospace Center (DLR) in 2006. He was a member of the editorial board of the International Journal of Electronics and Communications from 2012 to 2019, and he serves as a reviewer for a number of international journals and conferences. He is a member of the IEEE Microwave Theory and Techniques, the Communications, and the Vehicular Technology societies, OVE, and VDI. He served as the Chair for the IEEE Austrian Joint COM/MTT Chapter from 2002 to 2012.