IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 June 2024, accepted 8 July 2024, date of publication 18 July 2024, date of current version 29 July 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3430548*

RESEARCH ARTICLE

# Performance Evaluation of Asynchronous Pulse Code Multiple Access in Massive IoT Networks

**KENJI LEIBNITZ[1], FERDINAND PEPER[1], MIKIO HASEGAWA[2], (Member, IEEE), AND NAOKI WAKAMIYA[3], (Member, IEEE)**

[1]Center for Information and Neural Networks (CiNet), National Institute of Information and Communications Technology (NICT), Osaka 565-0871, Japan
[2]Department of Electrical Engineering, Tokyo University of Science, Tokyo 125-8585, Japan
[3]Graduate School of Information Science and Technology, Osaka University, Osaka 565-0871, Japan

Corresponding author: Kenji Leibnitz (leibnitz@nict.go.jp)

This work was supported by the Strategic Information and Communications Research and Development Promotion Programme (SCOPE), Ministry of Internal Affairs and Communications, Japan, under Grant 205007001.

**ABSTRACT** The Internet of Things (IoT) has revolutionized the way we interact with everyday objects by connecting sensors/actuators to the Internet to monitor and control various aspects of our environment. As the number of IoT devices continues to grow, efficient medium access control protocols are needed to ensure that communication takes place in a reliable and scalable manner. In this paper, we investigate the performance of the Asynchronous Pulse Code Multiple Access (APCMA) protocol in massive IoT scenarios through simulations and analysis. We show that APCMA is able to outperform CSMA/CA in terms of the number of successfully transmitted messages in scenarios with up to 30,000 transmitting sensor nodes while also being able to utilize the channel more efficiently. We also examine the sensitivity of APCMA's performance with respect to its parameters.

**INDEX TERMS** Massive Internet of Things (IoT), asynchronous pulse code multiple access (APCMA), high-density sensor networks, resource-restricted devices, pulse-based signals, communication through silence (CtS).

## I. INTRODUCTION

The Internet of Things (IoT) has seen a dramatic increase in the number of connected terminal nodes, and it is expected to continue to grow at an annual rate of 12% to hundreds of billions by 2030, ultimately with a connection density of 10 million nodes per square kilometer [1]. The massive amount of short-packet payload data generated by IoT applications has already created a mismatch with the capabilities of 5G networks that still needs to be overcome [1], [2]. This problem will become even worse for 6G, requiring the development of new efficient multiple access schemes for massive IoT systems. In general, massive IoT is characterized by the following features [3]:

- massive connectivity of nodes to their receiver,
- sporadic transmissions of small data packets,
- communication being mostly performed on the uplink,
- the need for extremely high energy efficiency.

Providing multiple access under massive connectivity is a key challenge that cannot be easily solved using traditional multiple access techniques [4]. Multiple access refers to the ability of a large number of nodes to simultaneously access and transmit their data over the shared wireless channel in a fair and efficient way. This issue becomes particularly challenging in scenarios with massive numbers of sensor nodes, where contention for channel access increases and collisions become more frequent.

Orthogonal multiple access (OMA) methods, such as TDMA, FDMA, or CDMA, can only allocate a radio resource block (RRB) to a single node to avoid nodes from interfering with each other. Non-orthogonal multiple access (NOMA) [5] schemes attempt to overcome this by allowing multiple nodes to simultaneously transmit their packets over the same RRB. Most NOMA schemes developed to date require a grant-based access mechanism to a shared RRB, but common grant-based IoT technologies, such as Low

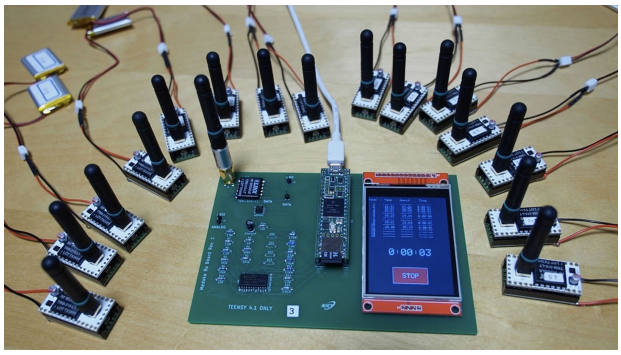
The associate editor coordinating the review of this manuscript and approving it for publication was Hosam El-Ocla.

Power Wide Area (LPWA) networks using LoRaWAN or Sigfox, have been shown to suffer from scalability issues [6], [7], [8]. Moreover, grant-based access imposes excessive delay and signaling overhead for massive IoT when using traditional multiple access techniques. For this reason, grant-free NOMA has been proposed, in which devices can transmit their data whenever they need to, without needing any scheduling method to deal with transmission collisions. However, the grant-free NOMA schemes proposed so far face many challenges [4], including the design of contention units selected by transmitters, synchronization with minimal signaling overhead, etc.

In [9], we address this problem by introducing Asynchronous Pulse Code Multiple Access (APCMA) as a grant-free method to encode information using codewords consisting of sparse pulse patterns. These pulse codes are designed in such way that a receiver can extract messages with a high probability of success, even when a large number of messages are superimposed on each other. APCMA has been inspired by Communication through Silence (CtS) [10], a scheme that encodes information using the time intervals between subsequent pulses in a message. However, since the intervals between pulses need to be kept silent in CtS, it is not robust to collisions of messages from different nodes, which is why it has been combined with other multiple access mechanisms like Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) in [11] and [12]. APCMA is inspired by CtS, but it resolves the problem of collisions in a different way, i.e., by adding more pulses to each codeword in such a way that combinations of pulse intervals are unique among all codewords. As a result, the original transmitted messages can be decoded even in the presence of many collisions. For an overview of schemes that are based on CtS, see Section II of [9].

To demonstrate its capabilities in real environments, we developed two types of prototype nodes for APCMA: (1) with Teensy 3.2 microcontrollers using On-Off Keying (OOK) at a carrier frequency of 315 MHz (Fig. 1(a)), and (2) with 1,500 nodes operating in the 920 MHz band (Fig. 1(b)), compliant with ARIB STD-T108 [13]. The latter version uses Chirp Spread Spectrum (CSS) for pulse modulation, improving noise robustness and communication range. CSS-APCMA leads to lower packet error rates than OOK-APCMA and LoRa in noisy and congested environments, respectively [14], [15], [16], [17]. We tested APCMA in various scenarios, including a study on temperature and humidity's impact on well-being in an office environment [18] and an outdoor experiment at Yokosuka Research Park, Japan, where we tested reception performance from groups of nodes over distances up to 1 km, see Fig. 1(b) [19].

In [20], we compared the performance when using APCMA codes with 4 to 6 pulses and up to 2,500 nodes with that of CSMA/CA by simulations, assuming that both systems operate with an equal throughput, but with different packet interarrival times. That study is extended in the current paper to a wider range of scenarios, and a more theoretical approach is followed as well. We have modified the traffic model to have the exact same arrival process on the application layer for APCMA and CSMA/CA, i.e., we generate messages at deterministic interarrival times $\tau$ and then pass them down to the lower layers on the protocol stack.



(a) Setup of an indoor experiment with 15 sensor nodes recording temperature and humidity in an office setting [18] and transmitting these values to a receiver using OOK-APCMA at 315 MHz

(b) Setup of an outdoor experiment at Yokosuka Research Park, Kanagawa Prefecture, Japan. Two trucks were each loaded with 500 nodes using CSS-APCMA at 920 MHz to transmit messages to a receiver antenna located at the top of a seven-storey building [19].

**FIGURE 1. Two implementations of APCMA devices that were developed and tested in experiments to evaluate the applicability of APCMA in real-world scenarios.**

The main contributions of this paper are as follows:

- Extend the models for computer simulation and theoretical analysis of APCMA to include codes with 4 to 10 pulses and up to 30,000 nodes.
- Investigate the sensitivity of APCMA's performance toward its four main input parameters: number of nodes, number of pulses, message length, and message interarrival time.
- Compare the performance of successful transmissions of APCMA to that of CSMA/CA using simulations and analysis.

This paper is organized as follows. In Section II we describe related work on massive IoT. In Section III,

we provide the background on APCMA, show how it uses discrete pulse codes to encode information, and describe how we evaluate its performance in terms of the probability of successful transmissions. In order to compare the performance of APCMA with that of CSMA/CA, we use a CSMA/CA model based on IEEE 802.15.4, which is described in Section IV. Through numerical analysis and simulations we show in Section V how APCMA has a significantly superior performance over CSMA/CA, and we investigate the effects of the key parameters on its performance. Section VI concludes this paper.

## II. RELATED WORK ON MASSIVE IOT NETWORKS

The term Massive IoT refers to applications with a huge number of low-cost, low-power devices [21]. These devices are typically less latency-sensitive and have relatively low throughput requirements. A number of research articles and survey papers have discussed the potential for massive IoT in 5G and 6G networks, as well as their transitional solutions, such as Long Term Evolution (LTE) or Beyond 5G (B5G).

De Andrade et al. [22] discuss the challenges and advancements in LTE networks to support the growing number of IoT devices. They show the necessity of enhancing LTE technology to accommodate the distinctive requirements of IoT applications while addressing the congestion caused when having a large number of concurrent connections. Secure communication when having many IoT devices is also important for Industrial IoT (IIoT). Mumtaz et al. [23] explain some of the challenges with IIoT connectivity in the future of building automation. To support massive Machine-Type Communications (MTC) in Narrowband-Internet of Things (NB-IoT), Narayanan et al. [24] propose an adaptive access mechanism to handle more connections and improve access success.

Akpakwu et al. [25] and Chettri and Bera [26] provide surveys that describe the integration of 5G networks with IoT. They address the diverse communication requirements of IoT devices and the limitations of current 4G LTE networks. The authors discuss emerging standards such as Extended Coverage-Global System for Mobile Communications for the Internet of Things (EC-GSM-IoT), enhanced MTC, NB-IoT, and 5G New Radio (NR), as well as the potential of using 5G technologies to handle the demands of IoT. Chen et al. [27] and Mlika and Cherkaoui [28] investigate the design principles and challenges of implementing massive access in 5G and B5G, focusing on cellular IoT applications. They propose novel concepts, theoretical models, and operational paradigms for future wireless networks. Additionally, [28] examines online user grouping, scheduling, and power allocation using NOMA, proposing an efficient online competitive algorithm. Guo et al. [1] survey the evolution of IoT toward 6G networks, discussing the limitations of 5G networks. The authors propose a 6G network architecture that integrates space, air, ground, and underwater networks, enhanced by edge computing, and examine the inclusion of emerging technologies such as machine learning and blockchains. They present a case study of fully autonomous driving to demonstrate how 6G could facilitate extensive interconnectivity of IoT devices with diverse service requirements.

In [29] and [30] solutions for managing traffic from a vast number of IoT devices and enhancing connectivity in massive MTC networks are proposed. Rodoplu et al. [29] introduce a multiscale algorithm that forecasts future traffic flows using a multilayer perceptron and preallocates the uplink wireless channel based on these forecasts, differing from the usual assumption of having random arrivals of IoT data packets. Simulations demonstrate the algorithm's scalability, supporting up to 6650 IoT devices, and its efficiency in throughput and energy consumption when compared to other protocols. Shoaei et al. [30] present a novel access scheme for massive MTC networks that utilizes a large-scale antenna array at the base station. The scheme optimizes the expected network throughput by dynamically partitioning each frame into grant-based and grant-free parts for devices with varying data throughput requirements. They propose a low-complexity algorithm to allocate the resources and optimize the access probabilities, resulting in higher throughput and lower packet delays, supporting up to 800 devices in their simulations.

The performance and quality of service (QoS) of IoT networks and communication technologies for massive MTC or massive IoT is discussed in [31] and [32]. Gelenbe et al. [31] address the issue of congestion in IoT networks, where numerous devices simultaneously attempt to send data to a gateway. They propose a Quasi-Deterministic Transmission Policy (QDTP) that reduces the probability of IoT data missing its deadline by scheduling packet transmissions at optimal times using diffusion approximation and queuing theory. The efficiency of this method is demonstrated by being able to meet transmission deadlines in congested IoT networks when using real-world IoT data. Furthermore, Stusek et al. [32] focus on the design of efficient communication protocols and devices for 5G and beyond networks, considering the growing number of resource-limited MTC devices. They examine existing communication protocols, including TCP, UDP, CoAP, and MQTT, with a focus on their use in cellular IoT standards such as NB-IoT, and show the importance of protocol overheads and data usage in relation to QoS and the associated costs for data transmission in massive IoT scenarios.

In most of these papers, the term ''massive'' refers to networks consisting of a few thousand nodes. In our proposal, which is described in the next section, we demonstrate that we could provide access to about ten times that number of devices.

## III. ASYNCHRONOUS PULSE CODE MULTIPLE ACCESS

APCMA uses codewords consisting of sparse pulse patterns (Fig. 2). It is designed to handle message collisions by adding redundant pulses to each codeword such that the unique pattern of each codeword can be recognized by a decoder at

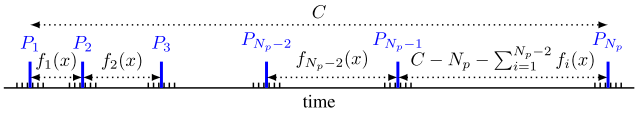



**FIGURE 2.** Generic format of a codeword in APCMA with $N_p$ pulses and length $C$ time slots. Intervals between pulses $i$ and $i+1$ are specified by encoding functions $f_i(x)$ for an encoded message $x$.

the receiver, even if the received message is interspersed with pulses originating from other nodes. Collisions are not really a problem in APCMA due to several factors:

- Each codeword is sparse, i.e., it contains a relatively small number of pulses compared to the codeword length.
- If collisions between messages occur, they can be successfully resolved with a high probability by using pattern recognition that detects valid combinations of pulses and rejects invalid ones.

In APCMA, any sender can initiate its transmission at any time without having to check whether other transmissions are in progress, i.e., without carrier sensing, and without needing any coordination with the receiver. The sparsity of the codewords contributes to the robustness of this multiple access mechanism even when thousands of messages are transmitted simultaneously [9], [33]. However, this also means that codewords need to be carefully designed so that the combinations of interpulse intervals of the codewords are as unique as possible and randomly received pulses from other nodes do not by chance form valid codewords, because that would lead to misdetections at the receiver.

### A. PULSE CODES IN APCMA

The general format of an APCMA codeword of length $C$ containing $N_p$ pulses is shown in Fig. 2. All variables that we use in this paper are summarized in Table 1. The $N_p - 1$ interpulse intervals of successive lengths

$$f_1(x),\ f_2(x),\ \ldots,\ f_{N_p-2}(x),\ C - N_p - \sum_{i=1}^{N_p-2} f_i(x)$$

are used to encode a single integer value $x$. The encoding functions $f_i$ are bijective and are shared in advance between the transmitters and the receiver. The codes considered in this paper have the property that the first and last slots of a codeword are always occupied by pulses. Furthermore, the second pulse of codeword $i$ is always at position $i+2$ and the $(N_p-1)$-th pulse of codeword $i$ is always at position $C-i-1$, for $1 \le i \le N_c$ where $N_c$ is the total number of codewords. The other pulses are determined in such a way that any two different codewords have at most two pulses in common at any relative displacement of the codewords to each other. Typically, the codeword length $C$ increases with the number of pulses $N_p$ and with the number of codewords $N_c$. Examples of APCMA codes with $N_p = 4,\ 5$ and 6 pulses consisting of $N_c = 10$ codewords are shown in Figs. 3(a), (b), and (c), respectively, where each row represents the pulse pattern of a different codeword.

Figure 3 shows that codes with 5 or more pulses in APCMA tend to exhibit less regularity compared to 4-pulse codes. The codes used in this paper all share the common characteristic that their codebooks have a maximum of one pulse per time slot (represented by a single column in the codebook), except for the first and last pulses, which are present in all codewords. Furthermore, we require by definition that there is always at least one free slot between any two pulses of

**TABLE 1.** List of symbols used in this paper. Subscripts $A$ and $C$ distinguish between variables for APCMA and CSMA/CA, respectively, and an asterisk indicates values that are empirically determined from simulations.

| Symbol | Meaning |
|---|---|
| $b_A$ | probability that an APCMA slot is occupied by a pulse |
| BE | current CSMA/CA backoff exponent |
| $C$ | APCMA codeword length in APCMA slots |
| CW | contention window in CSMA/CA slots |
| $\mathrm{CY}_A$ | interarrival time between packets in APCMA slots |
| $\Delta T_A$ | time slot duration in APCMA (= 10 µs) |
| $\Delta T_C$ | time slot duration in CSMA/CA (= 200 µs) |
| $F^*_{C,k}$ | number of CSMA/CA messages from node $k$ that exceed the maximum number of backoffs in simulation |
| $f_i(x)$ | encoding function, mapping a value $x$ to a number of APCMA slots in the interval between the $i$-th and $(i+1)$-th pulses |
| $M^*_{A,k}$ | number of APCMA messages sent by node $k$ in simulation |
| $M^*_{C,k}$ | number of CSMA/CA messages sent by node $k$ in simulation |
| maxBE | maximum backoff exponent in CSMA/CA |
| maxBO | maximum number of attempted backoffs in CSMA/CA |
| minBE | minimum backoff exponent in CSMA/CA |
| $N_b$ | number of bits used for encoding APCMA codewords ($N_b = \log_2 N_c$) |
| $N_c$ | number of APCMA codewords |
| $N_n$ | number of nodes communicating with the receiver |
| $N_p$ | number of pulses per APCMA codeword |
| NB | current number of backoffs performed in CSMA/CA |
| $\pi(x)$ | steady state probability of state $x$ in CSMA/CA Markov chain model [34] |
| $p$ | probability in CSMA/CA Markov chain model for leaving the IDLE state [34] |
| $p^c_i$ | probability in CSMA/CA Markov chain model that channel is free at first CCA [34] |
| $p^c_{i\|i}$ | probability in CSMA/CA Markov chain model that channel is free at second CCA, given it was free at first CCA [34] |
| $p^n_k$ | probability in CSMA/CA Markov chain model for leaving $k$-th backoff state [34] |
| $\mathrm{ps}_A$ | analytical APCMA success probability |
| $\mathrm{ps}^*_A$ | empirical APCMA success probability from simulations |
| $\mathrm{ps}_C$ | analytical CSMA/CA success probability |
| $\mathrm{ps}^*_C$ | empirical CSMA/CA success probability from simulations |
| $q_A$ | average density of APCMA pulses over time |
| $\tau$ | interarrival cycle time of packets (seconds) |
| $U^*_{A,k}$ | number of APCMA messages sent by a node $k$ that can be unambiguously decoded in simulation |
| $U^*_{C,k}$ | number of CSMA/CA messages that are successfully sent by node $k$ in simulation |

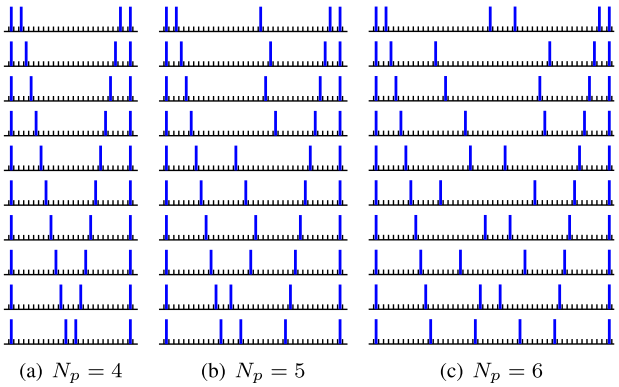



**FIGURE 3.** Examples of three APCMA codes having $N_c = 10$ codewords: (a) $N_p = 4$ pulses with codeword length $C = 25$, (b) $N_p = 5$ with $C = 36$, and (c) $N_p = 6$ with $C = 48$.

the same codeword, so that the beginning and the end of each pulse can be easily identified by a decoder. Under these conditions, we assume in this paper that the codeword length $C$ is given by

$$C = \begin{cases} 2(N_c + 2) + 1 & \text{if } N_p = 4 \\ (N_c + 2)(N_p - 2) & \text{if } N_p > 4 \end{cases}$$

time slots with each of the $N_c$ codewords having $N_p$ pulses.

### B. SCHEDULING OF TRANSMISSIONS

We consider a single-hop network configuration where all sensor nodes operate on discrete time slots and a receiver is constantly listening for transmissions from the nodes. In our simulations, we assume that each message contains only a single codeword as payload without the inclusion of any headers or trailers. Our envisaged application scenario is that of having a very large number of sensors monitoring critical structures, e.g., machinery inside manufacturing plants, sensors along railway lines, bridges, dams, etc. Each sensor node collects its data in a way that can be stored as $N_c$ different discrete values, e.g., $N_c = 1024$ different values of humidity or temperature, and reports this information to the receiver at regular intervals that are sufficiently short to be able to respond in a timely manner to anomalous conditions. Given the type of applications, we consider intervals of a few seconds appropriate, which is why we select intervals of 1 to 4 s. This is much shorter than in typical LPWA systems like LoRaWAN and Sigfox, which are limited by low duty cycle restrictions to at most 10 transmissions per hour.

The transmitting nodes send messages independent of each other according to the schedule illustrated in Fig. 4. Each node starts its schedule after an initial random number of time slots and then performs periodic cycles, each of which includes an active period and an idle period. The active period lasts for $C$ time slots and repeats at a cycle time of $\tau$ seconds, which also serves as the message interarrival time on the application layer. In the simulations of APCMA, each node randomly selects one of the $N_c$ codewords as the next message for transmission.

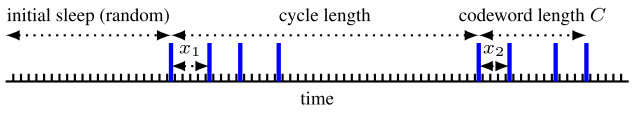



**FIGURE 4.** Scheduling of an arbitrary node in the APCMA simulation. After an initial random offset at the beginning of the simulation, each node transmits a sequence of randomly selected APCMA messages $x_1, x_2, \ldots$ of length $C$ time slots with constant interarrival cycle times $\tau$ until the end of the simulation is reached.

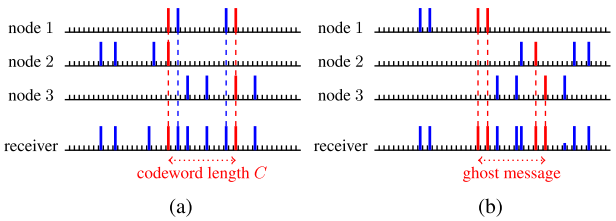



**FIGURE 5.** Three transmitting nodes sending 4-pulse APCMA messages to a receiver. (a) Pulses received from different nodes overlap each other (marked in red), but the corresponding codewords can still be decoded. (b) Detection of a ghost message (marked in red) at the receiver consisting of pulses received from different nodes. Although it was not transmitted by any single node, the ghost message is interpreted as a valid codeword at the receiver.

### C. DECODING IN APCMA AND GHOST MESSAGES

The decoding of messages at the receiver is done by a pattern recognition algorithm using either a finite automaton that operates on pulse sequences [33], [35], or a shift register that detects pulse sequences entering at the left and shifting one cell to the right each time step [36], [37].

In our simulations, we loop over all valid codewords, trying to match them with the pattern of pulses received within the window over the last $C$ time slots. Since the codewords have the structure shown in Fig. 3, they require a pulse in the first and last time slots. We exploit this knowledge to speed up the codeword detection algorithm by skipping all windows without a pulse in either the first or the last time slot. Superimposed pulses are not a concern in APCMA because they do not change the underlying pulse patterns, see Fig. 5(a). In this example, we have three nodes transmitting three different messages to the receiver, whereby the first pulse from node 1 is superimposed with the fourth pulse from node 2, and the fourth pulse from node 1 is superimposed with the third pulse from node 3 (marked in red). However, despite these superpositions of pulses, each of the three messages maintains its pulse pattern at the receiver and can be correctly decoded.

In high pulse density scenarios, randomly received pulses from different transmitters may be received in such a way that they form valid codewords, even if these codewords had not been sent by any node. We refer to these messages as ghost messages, see Fig. 5(b). Without additional error detection mechanisms, the receiver has no way of knowing whether a valid decoded codeword was actually sent by one of the transmitters, or whether it is the product of a ghost message. Note that while ghost messages add unwanted information,

they do not remove any legitimately sent information. As we will see later in the simulation results, the effect of ghost messages can be reduced by using codewords with more pulses $N_p$, so that they contain a more unique combination of interpulse patterns.

### D. ANALYTICAL MODEL OF SUCCESS PROBABILITY

We consider a message transmission to be successful if it can be unambiguously decoded by the receiver. This means that out of all the stray pulses falling into a window over the last $C$ time slots, none of these pulses will by chance form a ghost message. Following the argumentation in Sec. IV of [9], the average density of pulses over time is $q_A = N_p/\mathrm{CY}_A$, i.e., the fraction of the time slots occupied by the $N_p$ pulses divided by the total number of APCMA time slots per cycle $\mathrm{CY}_A$. Note that we use subscripts with capital letters A and C to distinguish between APCMA and CSMA/CA. We can now determine the probability $b_A$ that an arbitrary time slot is occupied by a pulse if there are $N_n$ transmitters.

$$b_A = 1 - (1 - q_A)^{N_n} \tag{1}$$

With (8) from [9], we arrive at a theoretical value of the success probability $\mathrm{ps}_A$, which is determined by the probability that no ghost message is formed.

$$\mathrm{ps}_A = \left(1 - b_A^{N_p-2}\right)^{N_c-1} \tag{2}$$

This analytical value for $\mathrm{ps}_A$ in (2) will be used later in Sec. V to verify the results from simulations and also for the comparison between APCMA and CSMA/CA.

### E. SUCCESS PROBABILITY FROM SIMULATION

In the APCMA simulation, we superimpose all transmitted pulse streams from each node at the receiver. The receiver searches at each time step for any detected codeword in this stream and compares any found message to the transmitted ones to determine whether it is correctly received or is an ambiguous message. The success probability in the APCMA simulations is calculated using the following equation

$$\mathrm{ps}_A^* = \frac{1}{N_n} \sum_{k=1}^{N_n} \frac{U_{A,k}^*}{M_{A,k}^*}, \tag{3}$$

where $U_{A,k}^*$ is the number of messages sent by node $k$ that can be unambiguously decoded in the simulation, and $M_{A,k}^*$ is the total number of messages transmitted by node $k$. We use the asterisk as superscript to indicate that a value is empirically determined from simulations.

## IV. CARRIER SENSE MULTIPLE ACCESS WITH COLLISION AVOIDANCE

CSMA/CA is a well-known method for coordinating grant-based access among multiple transmitters that share a common wireless medium. It is widely used in wireless local area networks (WLAN) within the IEEE 802.11 family of protocols and in wireless personal area networks (WPAN)

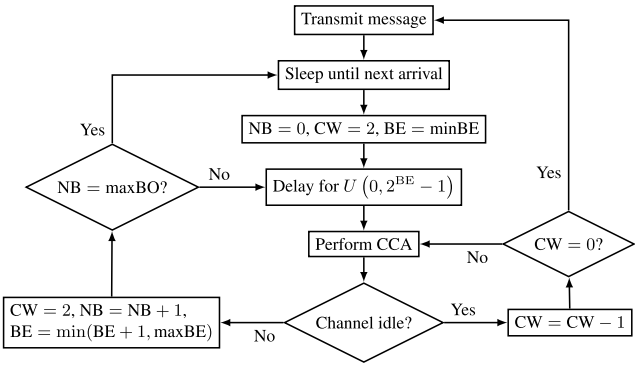



**FIGURE 6. Flowchart of the basic CSMA/CA algorithm in IEEE 802.15.4. Default parameters that we use in our study are the minimum backoff exponent of minBE = 3, the maximum backoff exponent maxBE = 5, and the maximum number of backoffs maxBO = 5.**

using IEEE 802.15.4. CSMA describes an entire family of random access protocols that do not require any preallocation of resources (e.g., time slots, frequencies, or codes), whereby each node can randomly access the communication channel whenever it has data to send. The essential feature of CSMA/CA is that before attempting any transmission, a node first senses the state of the channel to determine if it is free. Unlike its Ethernet counterpart CSMA/CD (CSMA with collision detection), which usually handles fewer nodes and reacts to detected collisions after they have occurred, CSMA/CA attempts to avoid collisions before they happen. Both variants of CSMA typically employ backoffs by waiting a time that is uniformly distributed over a time window that doubles with every backoff until an upper limit is reached.

### A. IEEE 802.15.4 CSMA/CA PROTOCOL

CSMA/CA in IEEE 802.15.4 operates on the link layer and uses discrete units of time called backoff slots. In the following we focus on the simplest case of CSMA/CA, which operates without inactive periods within the MAC layer superframe and without guaranteed time slots. The duration of a unit backoff slot depends on several factors, such as the modulation type and the frequency band.

Whenever a node attempts to transmit a message, it first performs a random backoff within the interval of $\left[0, 2^{\mathrm{BE}} - 1\right]$ backoff slots. The variable BE stands for the backoff exponent, which is initialized as minBE $= 3$ and is doubled each time the channel is detected as busy on each transmission attempt for the same message, until the maximum backoff exponent (maxBE) is reached. After its backoff timer expires, the node performs two clear channel assessments (CCA) in consecutive backoff slots to confirm that the channel is free. The node begins its transmission only if both of these CCAs are cleared successfully. If either CCA fails due to detecting an ongoing transmission by another node, the attempting node performs a new random backoff with an increased backoff window size and attempts to access the channel again when this new backoff timer expires. For both backoff parameters, the maximum backoff exponent maxBE and the maximum backoff limit maxBO, we use their default values

of 5. The flowchart in Fig. 6 summarizes the steps of the CSMA/CA algorithm.

There are three possible outcomes for a message transmission attempt made by a node after it successfully clears two CCAs. If it is the only node attempting to transmit in the next time slot, it will be counted as a success. However, if there is at least one more node also attempting to transmit in that time slot, it will not be counted as a success. If the node is always unsuccessful in getting a free transmission slot during its maxBO attempts, it aborts this message and it is considered a failure to transmit, and thus will not be counted as a success. The node will then subsequently wait and proceed to transmit the next message.

### B. ANALYTICAL MODEL OF CSMA/CA

There are numerous studies that have investigated the performance of the CSMA/CA backoff algorithm in IEEE 802.11 [38], [39], [40] and IEEE 802.15.4 [34], [41], [42], [43], [44] either by simulation or theoretical analysis. For this paper, we follow the analytical model in [34] and we also perform simulations of CSMA/CA to numerically compare its performance with that of APCMA.

Let us look at the model proposed by [34] and consider a CSMA/CA system with $N_n$ nodes transmitting to a single receiver. While we generally mostly stick to the notation in [34], we use a slightly different notation for the number of nodes, since [34] uses $N$ as the message length in backoff slots, which is kept at 1 throughout this paper. Furthermore, unlike [34], our model assumes periodic transmissions with deterministic interarrival times at a cycle length $\tau$, which we also need to take into account. One benefit of the model in [34] is that it uses approximations of uniform random variables by geometric random variables having the same mean. This lets us describe the system in much fewer states than other models using uniform random variables, e.g., [41], [43], while maintaining a sufficient accuracy.

#### 1) STATE TRANSITION DIAGRAM OF THE MARKOV MODEL

The state transition diagram of the Markov model is shown in Fig. 7. State IDLE represents the time that a node waits until the next packet arrives and is the state where the node spends most of its time. Then, following the algorithm in Fig. 6, one or more of the 5 backoff stages are traversed in states $BO_k$, before attempting the two CCA $CS_{k1}$ and $CS_{k2}$, for $k = 1, \ldots, 5$. If the second CCA is successful, the node moves on to the transmit state TX, after which it returns to IDLE. In the case that either of the CCAs from the fifth backoff is not successful, it will be marked as a failure and the node returns to the IDLE state without transmission.

The state transition diagram shown in Fig. 7 is parameterized by probabilities $p_i^c$ and $p_{i|i}^c$, which represent the probabilities that the channel is free at the first CCA and that it is free at the second CCA given it was free at the first CCA, respectively. Probabilities $p_k^n$, with $k = 1, \ldots, 5$, are the probabilities for leaving the backoff state $BO_k$, and they have the same mean values as their respective uniformly distributed counterparts, i.e., $p_1^n = 1/4.5$, $p_2^n = 1/8.5$, and $p_k^n = 1/16.5$ for $k = 3, 4, 5$.

#### 2) ESTIMATING THE IDLE DEPARTURE PROBABILITY $P$

Our first step is to determine the probability $p$ of leaving the IDLE state. Since the interarrival time $\tau$ is given in our case, we know the expected number of time slots between two packet arrivals, i.e., $\tau/\Delta T_C$. However, this time also includes the time spent for backoffs and transmission.

If we instantly leave the IDLE state in Fig. 7 by setting $p = 1$, we observe a sequence of states until we either successfully reach state TX or fail after states $CS_{51}$ or $CS_{52}$. Instead of then returning to the IDLE state, we enter an absorbing virtual state, marked as END in Fig. 7. In this way we obtain the average time spent for each path taken through the states from IDLE to END, including the time spent on backoffs, CCAs, and transmission of one packet at a given probability $p_i^c$, etc.; we denote this time as the expected waiting time $W = \sum_{k=0}^{\infty} k\, Q^k_{\mathrm{IDLE}\to\mathrm{END}}$, where $Q^k$ is the $k$-th power of the transition probability matrix of the absorbing system shown in Fig. 7 without the connection from END to IDLE states. Subtracting $W$ from $\tau/\Delta T_C$ then gives the expected time that the system resides in the idle state. Following the same line of thinking as in [34], we achieve this time by setting the probability $p$ to

$$p = \frac{1}{\tau/\Delta T_C - W + 1}.$$

#### 3) STEADY STATE PROBABILITIES AND SUCCESS PROBABILITY

Following [34], we define the probability of the first CCA being successful as $p_i^c$ and that of the second CCA, conditioned that the first CCA was successful, as $p_{i|i}^c$, see (3) in [34].

$$p_{i|i}^c = \frac{2p_i^c - 1}{p_i^c}$$

Considering the channel state model in Sect. V-C of [34], we obtain the probability of a node starting a transmission under the condition that the channel has been idle for two consecutive CCA slots as

$$p_{t|ii}^n = \frac{p_t^n}{2p_i^c - 1}.$$

With the probability $\alpha$ of success in the first CCA in the presence of $N_n$ nodes,

$$\alpha = (1 - p_{t|ii}^n)^{N_n}$$

we get

$$p_i^c = \frac{2 - \alpha}{3 - 2\alpha}.$$

With the steady state probability of a node to be transmitting $p_t^n = \pi(tx)$ and the probability $p$ of leaving the IDLE state, we numerically iterate over the equations for $p_{t|ii}^n$,

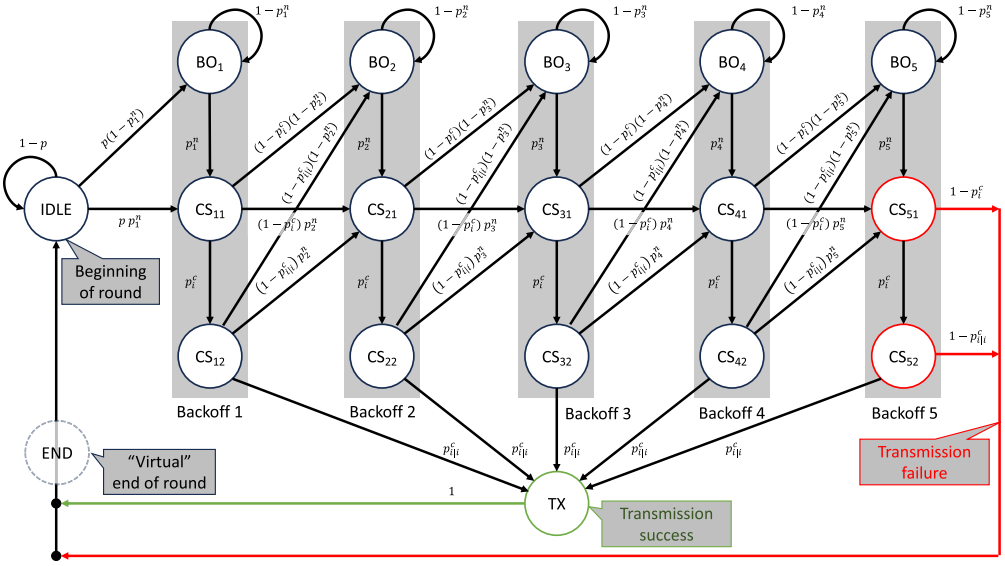



**FIGURE 7.** State transitions in the Markov chain model based on [34] for the CSMA/CA analysis. We use this diagram for two steps in our computation: (1) determining the average number of the backoff steps from the beginning of a round to its end, and (2) computing the steady state success probability.

$\alpha$, and $p_i^c$ until all their values converge, after which we obtain the analytical success probability for CSMA/CA as $\mathrm{ps}_C$ as:

$$\mathrm{ps}_C = \frac{p_t^n}{p_t^n + p_{\mathrm{fail}}} \left(1 - p_{t|ii}\right)^{N_n - 1}. \tag{4}$$

Here, $p_{\mathrm{fail}}$ is the probability that the transmission is aborted by failing either of the two CCAs in the last backoff attempts with their steady state probabilities $\pi(cs_{51})$ and $\pi(cs_{52})$:

$$p_{\mathrm{fail}} = \left(1 - p_i^c\right) \pi(cs_{51}) + \left(1 - p_{i|i}^c\right) \pi(cs_{52}). \tag{5}$$

For further details and explanation on the model and its notations, the reader is referred to the original paper by Ramachandran et al. [34].

We also run simulations of CSMA/CA where we determine the success probability as follows

$$\mathrm{ps}_C^* = \frac{1}{N_n} \sum_{k=1}^{N_n} \frac{U_{C,k}^*}{M_{C,k}^* + F_{C,k}^*}. \tag{6}$$

$U_{C,k}^*$ is the number of messages that node $k$ could successfully send during the simulation, i.e., when it was the only node at the time attempting a transmission, while $M_{C,k}^*$ is the total number of messages that node $k$ could broadcast after clearing the backoffs. The term $F_{C,k}^*$ logs all trials in which node $k$ could not transmit its message after five failed backoff attempts.

## V. NUMERICAL RESULTS

We evaluate the results from simulation and analysis for APCMA with various numbers of pulses $N_p$ and number of nodes $N_n$, and we compare them to the results for CSMA/CA.

### A. SIMULATION PARAMETERS

We assume the same deterministic message arrival process in CSMA/CA from higher layers as for APCMA, where each node generates messages at a constant cycle time of $\tau = 4\,\mathrm{s}$. In both models we transmit 10-bit messages ($N_b = 10$). This corresponds to a codebook with $N_c = 1,024$ codewords in APCMA. We set the width of a time slot in APCMA to $\Delta T_A = 10\,\mu\mathrm{s}$. To allow a fair comparison between the two models, we assume that a *Manchester coding* of bits is used in CSMA/CA, whereby the high part of a bit encoding has the same width as a pulse in APCMA. Together with the low part of a bit with the same width, this yields a bit width of 20 µs. For the width of a 10-bit message in CSMA/CA we thus use $\Delta T_C = 200\,\mu\mathrm{s}$, and this will also be the width of a backoff slot in CSMA/CA. Given that the cycle time of a message is $\tau = 4\,\mathrm{s}$, we then arrive at a cycle duration of $\tau/\Delta T_C = 20,000$ slots. For the sake of simplicity, we assume that a message transmitted in CSMA/CA fits entirely into one backoff slot.

For each participating node we generate 20 message arrivals with constant cycle times $\tau$ and we continue running the simulation until all of these 20 messages have been transmitted for all nodes. Since the arrival times are given in real time and we use discrete time slots in both of our models, we delay the arrival time of a message to the next time slot following the arrival.

### B. SUCCESS PROBABILITY OVER NUMBER OF NODES

While the same term *success probability* is used here for both access schemes, CSMA/CA and APCMA, its interpretation differs slightly for both. In CSMA/CA, we compute success probability as the probability that an attempted transmission of a message by a node is successfully accomplished, i.e.,

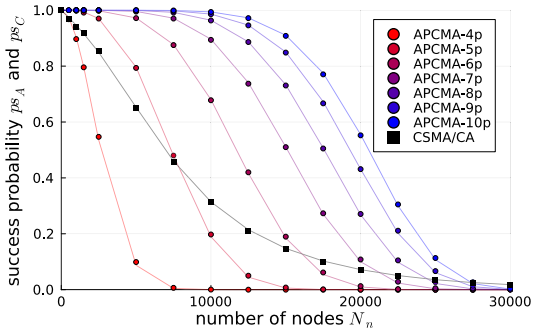



**FIGURE 8.** Success probability over the number of nodes $N_n$ for message interarrival time $\tau = 4\,\mathrm{s}$. Circular markers are from APCMA simulations, black square markers are from the CSMA/CA simulations, and all lines are from the corresponding analyses results.

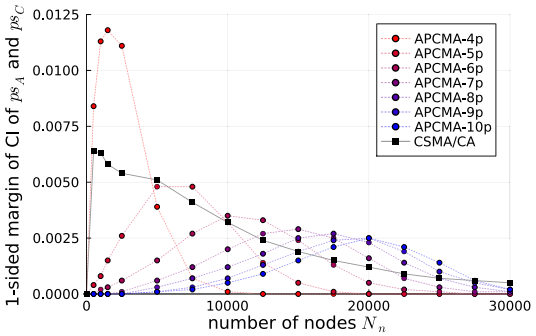



**FIGURE 9.** One-sided confidence interval lengths of the simulated success probability averaged over the number of nodes $N_n$ for message interarrival time $\tau = 4\,\mathrm{s}$.

that the node's transmission does not collide with those of other nodes, and its attempt to transmit is not aborted due to having too many backoffs. On the other hand, in APCMA we interpret success probability as the probability that stray pulses from other nodes don't end up to be interpreted as ghost messages. While these two metrics may seem slightly different at first glance, they both characterize the amount of data that is successfully transmitted from the transmitters to the receiver.

The success probability plotted against the number of nodes is shown in Fig. 8. Our first observation is that all analytical models (solid lines) fit rather well with their respective simulated values (discrete markers) for both CSMA/CA and APCMA. We also see that the curve for CSMA/CA intersects with APCMA with $N_p = 5$ pulses at about 8,000 nodes and with $N_p = 6$ pulses at about 16,000 nodes. Note that the success probability values at these intersection points are all less than 0.5, which already represent impractical scenarios in both cases. For more practical values of success probability, e.g., above 0.8 or 0.9, CSMA/CA performs much worse than APCMA, except for the 4-pulse case.

To demonstrate the statistical reliability of our simulation results, we compute the 95% confidence intervals over the success probability averaged over all nodes in each scenario. Since we are dealing with very large numbers of nodes and our metric takes values that are limited to the interval [0, 1], the confidence intervals that we obtain are very small in size. Therefore, we do not include confidence intervals in Fig. 8, but instead illustrate the one-sided confidence interval lengths for each APCMA version and CSMA/CA separately in Fig. 9 as a function of the number of nodes. The largest confidence intervals for each APCMA variant occur when the corresponding success probability is close to 0.5 in Fig. 8. In all of the considered cases, the confidence intervals remain below 0.0125.

We also ran simulations with $\tau = 1\,\mathrm{s}$ and $2\,\mathrm{s}$ and found that the number of nodes at which a certain success probability is achieved scales approximately linearly with $\tau$. The exact values of the intersection points of nodes $N_n$ between CSMA/CA and APCMA with $N_p = 4, \ldots, 10$ are shown in Table 2 for all three values of $\tau = 1\,\mathrm{s}$, $2\,\mathrm{s}$ and $4\,\mathrm{s}$, where a consistent linear trend can be recognized.

**TABLE 2.** Number of nodes $N_n$ at which the analytical success probabilities for CSMA/CA and APCMA intersect for $\tau = 1\,\mathrm{s}$, $2\,\mathrm{s}$ and $4\,\mathrm{s}$, i.e., $N_n$ at which $\mathrm{ps}_A = \mathrm{ps}_C$. Below $N_n$, the success probability for CSMA/CA is worse than for APCMA, while above it CSMA/CA is better. The success probability at this intersection point is called $\mathrm{ps}_*$ and it is shown in the last column. Because of the observed linear scalability of $N_n$ with $\tau$ the success probability happens to have the same value for all values of $\tau$, so only a single column is included for $\mathrm{ps}_*$. The only practical value of $\mathrm{ps}_*$ where CSMA/CA outperforms APCMA is for $N_p = 4$.

| $N_p$ | $N_n \vert \tau = 1\,\mathrm{s}$ | $N_n \vert \tau = 2\,\mathrm{s}$ | $N_n \vert \tau = 4\,\mathrm{s}$ | $\mathrm{ps}_*$ |
|---|---|---|---|---|
| 4 | 128.32 | 257.75 | 516.62 | 0.97 |
| 5 | 1,965.77 | 3,932.30 | 7,865.36 | 0.43 |
| 6 | 4,083.39 | 8,166.90 | 16,333.91 | 0.10 |
| 7 | 5,389.30 | 10,778.64 | 21,557.33 | 0.04 |
| 8 | 6,196.04 | 12,392.16 | 24,784.40 | 0.03 |
| 9 | 6,686.34 | 13,372.81 | 26,745.75 | 0.02 |
| 10 | 6,973.00 | 13,946.18 | 27,892.54 | 0.02 |

Figure 8 also shows that when the number of nodes $N_n$ increases, the APCMA success probability decreases. The reason for this is that the number of ghost messages increases with $N_n$, because having more nodes leads to more time slots being occupied with pulses, so that it becomes more difficult to unambiguously extract messages by pattern recognition at the receiver. Increasing $N_p$ tends to compensate for this fact, because a larger number of pulses leads to a smaller probability that the recognition of the unique pulse pattern of a codeword is caused by a ghost message. However, the increase in success probability becomes smaller with each successive increase in pulses $N_p$ due to the higher pulse density.

Increasing the number of pulses even further would not necessarily lead to an increase in the number of nodes that can be supported in a system at an acceptable success probability. To get a rough estimate of the maximum number of pulses that can be used in practical applications, we compute the success probabilities for up to 30,000 nodes and up to 50 pulses with the analytical model (Fig. 10). When $N_p$ is increased beyond 19 pulses at $\tau = 4\,\mathrm{s}$, the pulse density increases so much that more ghost messages appear, leading to a decrease in success probability. Thus, assuming a target success probability of

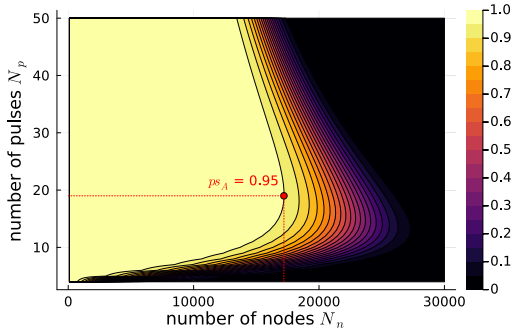



**FIGURE 10.** Analytical APCMA success probability $ps_A$ versus the number of nodes $N_n$ on the horizontal axis and the number of pulses $N_p$ on the vertical axis for message interarrival time $\tau = 4\,\mathrm{s}$. The maximum number of pulses is highlighted for $ps_A = 0.95$ at $N_n = 17,200$ and $N_p = 19$. When the number of pulses increases beyond 19, the high pulse density leads to increased ghost messages that lower the success probability.

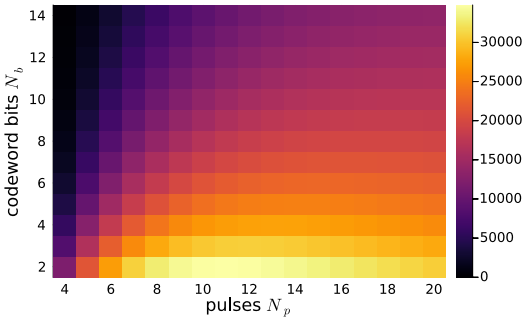



**FIGURE 11.** Analytical computation of APCMA capacity in number of nodes $N_n$ when the acceptable success probability limit is $ps_A = 0.95$, shown as function of the number of pulses $N_p$ and the number of bits for codeword $N_b$ when $\tau = 4\,\mathrm{s}$. The corresponding number of nodes for CSMA/CA at a fixed number of $N_c = 1,024$ codewords are $N_n = 933.58$.

0.95 and $\tau = 4\,\mathrm{s}$, it is not beneficial to increase the number of pulses $N_p$ beyond 19. As expected from our previous results, for $\tau = 1\,\mathrm{s}$ and $2\,\mathrm{s}$ we obtain the same limit on the number of pulses, however, at a quarter and a half of the number of nodes, respectively.

### C. SENSITIVITY OF SUCCESS PROBABILITY TO APCMA PARAMETERS

The analytical model of APCMA permits a direct evaluation of the impact of the APCMA parameters on performance. The simulations, on the other hand, do not allow such an evaluation in an easy way due to excessive computation times that are needed for such large scenarios. In this section we analytically evaluate the sensitivity of $ps_A$ to $N_b$, $N_n$, $N_p$, and $\tau$.

We have already seen in Figs. 8 and 10 that with an increasing number of nodes $N_n$ the probability of success $ps_A$ drops. Let us now investigate how many nodes can be accommodated in the system if our objective is to reach a given target success probability $ps_A$. By numerically inverting (2) and varying over the number of pulses $N_p$ and the number of bits $N_b = \log_2 N_c$ used to encode the $N_c$ codewords, we obtain Fig. 11 for $ps_A = 0.95$. This matrix

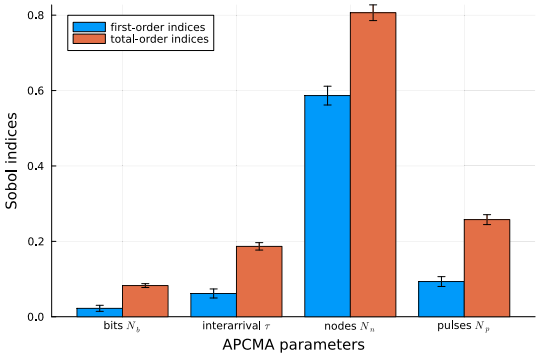



**FIGURE 12.** Results from the sensitivity analysis with the Sobol method show the first and total-order indices with 95% confidence intervals for the four parameters $1 \le N_n \le 30,000$, $4 \le N_p \le 10$, $8 \le N_b \le 12$, and $1\,\mathrm{s} \le \tau \le 4\,\mathrm{s}$. The number of message bits $N_b$ has the smallest, the number of nodes $N_n$ the highest, and interarrival time $\tau$ and number of pulses $N_p$ are nearly equal in Sobol indices.

lets us determine the best settings of APCMA when the target parameters $ps_A$, $N_b$, and $N_p$ are given.

If we change the interarrival time to $\tau = 1\,\mathrm{s}$ and $2\,\mathrm{s}$, the only difference that we would see is that the z-axis with the number of nodes scales down accordingly. Numerically inverting the curve of CSMA/CA success probability at $\tau = 4\,\mathrm{s}$ and $N_b = 10$ yields $N_n = 933.58$ at $ps_C = 0.95$. This value also scales linearly with $\tau$ and stays at about the value of APCMA with $N_p = 4$ pulses and $N_c = 1,024$ codewords as we could already see in Fig. 8.

We also perform a variance-based sensitivity analysis based on the method by Sobol [45], Saltelli [46], Saltelli et al. Saltelli2010 using a Julia implementation [48] of the Python library SAlib [49], [50]. For the four input parameters, the number of nodes $1 \le N_n \le 30,000$, the number of pulses $4 \le N_p \le 10$, the number of codeword bits $8 \le N_b \le 10$, and the interarrival time $1\,\mathrm{s} \le \tau \le 4\,\mathrm{s}$, we find the contribution of the uncertainty of these parameters to the uncertainty of the resulting analytical success probability $ps_A$. The results are summarized as colored bars in Fig. 12, with error bars indicating the 95% confidence intervals. The higher the Sobol indices are, the larger is the influence of a specific variable on the variance of the success probability. Figure 12 shows that the variance of the number of bits $N_b$ has the smallest influence, followed by the interarrival time $\tau$ and the number of pulses $N_p$. The highest influence is observed for the number of nodes $N_n$ when dealing with direct contributions of the parameters (first-order sensitivity index). The same tendency is also observed when including all contributions to the output variance, including those caused by interactions among the parameters (total-order sensitivity index). Note, that while in general we do not consider changing the APCMA parameters during operation, it would be nevertheless possible to do so, e.g., in order to accommodate for a sudden increase in traffic from the nodes.

Increasing the range of the interarrival time to $1\,\mathrm{s} \le \tau \le 3600\,\mathrm{s}$ changes the roles of the variables significantly, see Fig. 13. We can see that now the effect of variations of

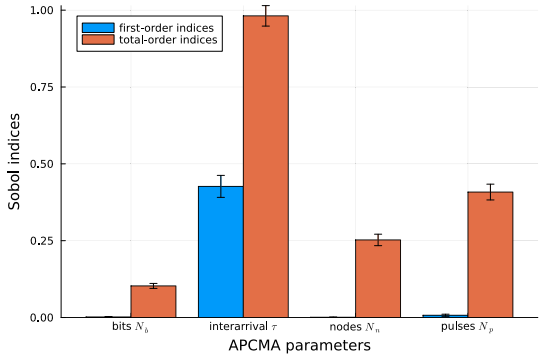



**FIGURE 13.** Results from the sensitivity analysis with the Sobol method when the range of the interarrival $\tau$ lies between 1 s and 3600 s. The first-order Sobol indices all become very small, except for the interarrival time $\tau$, which has now the biggest effect on the sensitivity of success probability. The number of nodes $N_n$ becomes less important and drops below the number of pulses $N_p$, while the number of bits $N_b$ remains the smallest in both scenarios.

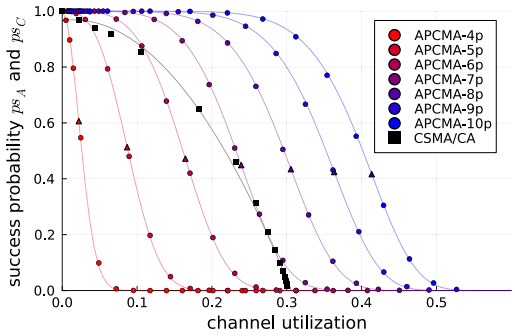



**FIGURE 14.** Success probability over channel utilization for interarrival time $\tau = 4$ s. Circular markers are from APCMA simulations, black square markers are from the CSMA/CA simulations, lines are from the APCMA analysis, and the black line is from the numerical fit of the CSMA/CA results with the parabola function. The triangle markers indicate the locations of the inflection points shown in Table 3 for each APCMA curve.

the interarrival time $\tau$ will impact the success probability strongest, followed by the number of pulses $N_p$. The number of nodes $N_n$ drops in importance and the number of bits $N_b$ remains the least sensitive. Except for the interarrival time $\tau$, all first-order influences of the input variables disappear, leading to only the interaction of the input parameters playing a key role.

### D. SUCCESS PROBABILITY OVER UTILIZATION

In each simulation run, we log the number of time slots that are occupied by transmissions, and we define the ratio of this number over the total number of time slots during the simulation as the *channel utilization*. We plot this utilization value against the success probability in Fig. 14 as discrete markers for all pairs of data points obtained from all simulations with the same cycle time $\tau$. Figure 14 also shows the analytical success probability from (2) as solid lines.

#### 1) ANALYTICAL EXPRESSION FOR APCMA

We can also show this relationship analytically from (2), which already gives us the success probability as a function

**TABLE 3.** Inflection point of success probability over utilization for $\tau = 4$ s. The success probability at this point lies near 0.5, but decreases with increasing $N_p$.

| $N_p$ | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|
| $\tilde{b}_A$ | 0.02 | 0.09 | 0.16 | 0.24 | 0.31 | 0.36 | 0.41 |
| $\mathrm{ps}_A(\tilde{b}_A)$ | 0.61 | 0.51 | 0.47 | 0.45 | 0.43 | 0.42 | 0.42 |

of the channel utilization $b_A$ and where we take its second derivative as

$$\frac{\partial^2 \mathrm{ps}_A}{\partial b_A^2} = (N_c - 1)\left(N_p - 2\right) b_A^{N_p-4}\left(1 - b_A^{N_p-2}\right)^{N_c-3} \times \left[(N_c-2)\left(N_p-2\right) b_A^{N_p-2} - \left(N_p - 3\right)\left(1-b_A^{N_p-2}\right)\right].$$

This function has trivial roots at $\tilde{b}_A = 0$ and $\tilde{b}_A = 1$ and a non-trivial root at

$$\tilde{b}_A = \left[\frac{N_p - 3}{(N_p - 2)(N_c - 1) - 1}\right]^{\frac{1}{N_p-2}}.$$

This point $\tilde{b}_A$ is the inflection point of the utilization at which the success probability curve switches from being concave to convex. The success probability at this point becomes

$$\mathrm{ps}_A(\tilde{b}_A) = \left[\frac{(N_p - 2)(N_c - 2)}{(N_p - 2)(N_c - 1) - 1}\right]^{N_c-1}. \tag{7}$$

This inflection point $\tilde{b}_A$ changes depending on $N_p$ (see Table 3 and the triangular markers in Fig. 14), but its success probability always stays near 0.5 and slightly decreases with increasing $N_p$. Since $\mathrm{ps}_A(b_A)$ is a monotonously decreasing function over $b_A$, this inflection point represents the point with the steepest decline, and it shifts slightly downwards with every increase in $N_p$. We see from Fig. 14 that $\mathrm{ps}_A(b_A)$ remains high for as long as possible before dropping steeply.

#### 2) NUMERICAL FIT FOR CSMA/CA

Similar to the utilization of time slots by pulses in APCMA, we can also determine the utilization of occupied backoff slots in CSMA/CA simulations. However, due to the higher complexity of the backoff process, we only numerically fit the relationship between utilization and success probability from simulations. A good fit happens to be given by a parabola-shaped function:

$$par(x, p_1, p_2, p_3) = p_1 x^2 + p_2 x + p_3.$$

Then, for CSMA/CA we collect all pairs of success probability and channel utilization obtained over all simulations with interarrival time $\tau = 4$ s. We plot the success probability against the channel utilization and numerically fit the parameters $p_1$, $p_2$ and $p_3$ of the parabola function using a Levenberg-Marquardt algorithm for non-linear fitting implemented in Julia [51]. The parameters for the parabola fit for CSMA/CA are $p_1 = -11.37 \pm 1.60$, $p_2 = 0.33 \pm 0.53$, and $p_3 = 0.97 \pm 0.03$, where the margins indicate the

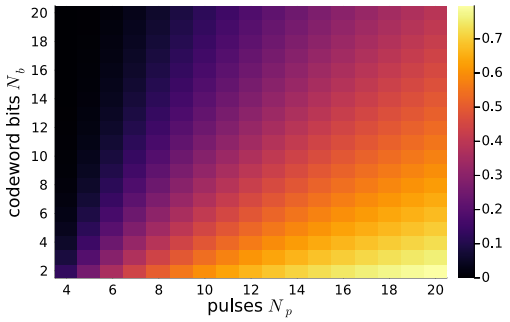



**FIGURE 15.** Analytical capacity of APCMA system at $\tau = 4$ s in terms of channel utilization $b_A$ when the success probability is $\text{ps}_A = 0.95$, shown as function of number of pulses $N_p$ and number of bits for codeword $N_b$.

95% confidence intervals from the fit. The coefficient of determination is $R^2 = 0.998$, indicating that the numerical fit matches very well with the simulated values. We also see that the curve approaches the x-axis nearly perpendicular at a utilization of about 0.33, indicating the existence of a hard upper utilization limit for CSMA/CA.

While the shapes of Figs. 8 and 14 are basically similar for APCMA, the channel utilization variable aggregates the parameters $C$ and $N_n$ into a single value, which we can arbitrarily scale up. In Fig. 8, we could not increase the codeword bits $N_b$ beyond a certain limit, since we would then have messages that become longer to transmit than the interarrival time $\tau$. However, since the same utilization value can be reached by several different combinations of the parameters, we can now avoid this problem. The surface plots of the target utilization at the success probability $\text{ps}_A = 0.95$ is shown in Fig. 15. This plot gives us a more general view of the interactions between the key parameters of APCMA when the parameters $\text{ps}_A$, $N_b$, and $N_p$ are given.

## VI. DISCUSSION AND CONCLUSION

Sparse pulse-based APCMA codes have great potential for providing massive multiple access in IoT, but investigations on their performance at various parameter settings, especially the number of pulses per codeword, had been lacking until now. Our comparison with CSMA/CA shows that with a sufficient number of pulses, APCMA can achieve a significantly higher probability of successful transmission when the number of nodes is very large. In contrast to CSMA/CA, which requires each node to sense whether the channel is free before transmissions, APCMA is a grant-free access scheme that does not require carrier-sense. The results in this paper suggest that APCMA is a viable multiple access scheme for high-density applications of IoT, in which high connectivity rather than high data rates are the main objective.

Typically, duty cycles of messages in APCMA are very low. In this paper we assume a time slot duration of $\Delta T_A = 10\,\mu$s and interarrival time at the application layer of $\tau = 4$ s. Thus, one APCMA cycle consists of $\text{CY}_A = \tau/\Delta T_A$ time slots of which $N_p$ slots are actively used for transmitting pulses, resulting in a duty cycle of $N_p\,\Delta T_A/\tau$. For $N_p = 10$, we then have a duty cycle of $2.5 \times 10^{-5}$. In comparison, other schemes like LoRa usually operate at much higher duty cycles than this [52], [53]. As a result, LoRa is limited in the number of messages that can be sent per time unit in a regime without carrier-sense. For example, given a duty cycle of 1%, LoRa can typically only send 10 messages per hour. APCMA is much less sensitive to such restrictions.

A new phenomenon related to the particulars of APCMA is the occurrence of ghost messages. These are caused by the ambiguities that arise when multiple messages are combined on a single radio channel. As the number of received pulses within a message duration increases, so too does the percentage of ghost messages. Based on our analytical and simulation results with up to 10 pulses per codeword, we have found that increasing the number of pulses per codeword significantly improves the probability of successful communication. Even though this leads to a higher pulse density, this is more than compensated for by the higher reliability of using more unique patterns of pulses. There is a diminishing effect, however, so increasing the number of pulses comes at a lower gain in success probability up to 19 pulses, beyond which the success probability decreases due to having too many ghost messages.

The current implementation of APCMA encodes its payload information as unary-encoded distances between two pulses. This leads to codewords becoming much longer than their binary-encoded counterparts and requiring more time for transmission. However, since time slots are short (10 µs in this paper), even for long codewords the frequency at which transmissions can take place is in the range of a few to tens of seconds.

This paper only considers the transmission of messages consisting of a single frame of payload data. The concatenation of multiple frames to form multi-frame messages in order to increase the transmission rate has yet to be investigated. When a successful transmission is defined as the correct transmission of all frames, the probability of success will decrease when the number of frames per message increases. However, the use of multiple frames per message opens up the possibility of using some of these frames for error correction, thus improving the overall success probability.

APCMA is expected to be much more energy-efficient than conventional systems, since no retransmissions are needed and each message only consists of a rather small number of emitted pulses. In our future work, we plan to experimentally measure the amount of energy needed for each sensor node, to provide a detailed mathematical model of the battery lifetimes in a typical sensing environment.

## REFERENCES

[1] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 11891–11915, Aug. 2021.

[2] M. Pons, E. Valenzuela, B. Rodríguez, J. A. Nolazco-Flores, and C. Del-Valle-Soto, "Utilization of 5G technologies in IoT applications: Current limitations by interference and network optimization difficulties—A review," *Sensors*, vol. 23, no. 8, p. 3876, Apr. 2023.

[3] J. Che, Z. Zhang, Z. Yang, X. Chen, and C. Zhong, "Massive unsourced random access for NGMA: Architectures, opportunities, and challenges," *IEEE Netw.*, vol. 37, no. 1, pp. 28–35, Jan. 2023.

[4] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1805–1838, 3rd Quart., 2020.

[5] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *Proc. IEEE 77th Veh. Technol. Conf. (VTC Spring)*, Jun. 2013, pp. 1–5.

[6] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1841–1876, 3rd Quart., 2023.

[7] 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.*, New York, NY, USA, Nov. 2016, pp. 59–67.

[8] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35816–35825, 2019.

[9] F. Peper, K. Leibnitz, C. Tanaka, K. Honda, M. Hasegawa, K. Theofilis, A. Li, and N. Wakamiya, "High-density resource-restricted pulse-based IoT networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1856–1868, Dec. 2021.

[10] Y. Zhu and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2005, pp. 140–147.

[11] Y. P. Chen, D. Wang, and J. Zhang, "Variable-base tacit communication: A new energy efficient communication scheme for sensor networks," in *Proc. 1st Int. Conf. Integr. Internet Ad Hoc Sensor Netw. InterSense*, New York, NY, USA, 2006, p. 27–es. [Online]. Available: https://dl.acm.org/doi/10.1145/1142680.1142715

[12] D. Feng, S. Das, F. Hajiaghajani, Y. Shi, and S. Biswas, "Pulse position coded medium access in energy-starved networks," *Comput. Commun.*, vol. 148, pp. 62–73, Dec. 2019.

[13] *920 MHz-Band Telemeter, Telecontrol and Data Transmission Radio Equipment*, Association of Radio Industries and Businesses (ARIB), Tokyo, Japan, Apr. 2021.

[14] K. Honda, A. Nakamura, F. Peper, K. Leibnitz, N. Wakamiya, and M. Hasegawa, "Performance evaluation of CSS-APCMA by experiments using 500 devices for massive IoT," in *Proc. Int. Symp. Nonlinear Theory Appl. (NOLTA)*, Dec. 2022, pp. 343–346, doi: 10.34385/proc.71.B3L-D-04.

[15] N. Wakamiya, K. Leibnitz, F. Peper, and M. Hasegawa, "Evaluation and optimization of asynchronous pulse code multiple access," *Nonlinear Theory Appl., IEICE*, vol. 13, no. 2, pp. 318–323, 2022.

[16] A. Nakamura, K. Honda, F. Peper, K. Leibnitz, N. Wakamiya, and M. Hasegawa, "Performance evaluation of CSS-APCMA for massive IoT using GNU Radio/USRP," *IEICE Commun. Exp.*, vol. 12, no. 4, pp. 120–125, 2023.

[17] K. Shiotsuka, K. Honda, K. Leibnitz, F. Peper, N. Wakamiya, Y. Maeda, and M. Hasegawa, "Experimental demonstration of APCMA's scalability advantage over current LPWA," in *Proc. RISP Int. Workshop Nonlinear Circuits, Commun. Signal Process. (RISP NCSP)*, 2023, pp. 13–16.

[18] B. Wutzl, K. Leibnitz, D. Kominami, Y. Ohsita, M. Kaihotsu, and M. Murata, "Analysis of the correlation between frontal alpha asymmetry of electroencephalography and short-term subjective well-being changes," *Sensors*, vol. 23, no. 15, p. 7006, Aug. 2023.

[19] M. Hasegawa, A. Nakamura, K. Honda, F. Peper, K. Leibnitz, and N. Wakamiya, "Experimental demonstration of high density wireless commutation realized by asynchronous pulse code multiple access using 1000 devices," in *Proc. IEICE Soc. Conf.*, 2023. [Online]. Available: https://confit.atlas.jp/guide/event/society2023/subject/N-2-6/advanced

[20] K. Leibnitz, F. Peper, K. Theofilis, M. Hasegawa, and N. Wakamiya, "Evaluating multiple-access protocols: Asynchronous pulse coding vs. carrier-sense with collision avoidance," in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, T. Hara and H. Yamaguchi, Eds., Cham, Switzerland: Springer, 2022, pp. 693–706.

[21] C. Lundqvist, A. Keränen, B. Smeets, J. Fornehed, C. R. B. Azevedo, and P. von Wrycza, "Key technology choices for optimal massive IoT devices," *Ericsson Technol. Rev.*, vol. 98, no. 1, pp. 48–59, 2019.

[22] T. P. C. De Andrade, C. A. Astudillo, L. R. Sekijima, and N. L. S. Da Fonseca, "The random access procedure in long term evolution networks for the Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 124–131, Mar. 2017.

[23] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIoT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.

[24] S. Narayanan, D. Tsolkas, N. Passas, and L. Merakos, "ADAM: An adaptive access mechanism for NB-IoT systems in the 5G era," *IEEE Access*, vol. 9, pp. 109915–109931, 2021.

[25] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[26] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[27] X. Chen, D. W. K. Ng, W. Yu, E. G. Larsson, N. Al-Dhahir, and R. Schober, "Massive access for 5G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 3, pp. 615–637, Mar. 2021.

[28] Z. Mlika and S. Cherkaoui, "Massive IoT access with NOMA in 5G networks and beyond using online competitiveness and learning," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13624–13639, Sep. 2021.

[29] V. Rodoplu, M. Nakip, D. T. Eliiyi, and C. Guzelis, "A multiscale algorithm for joint forecasting–scheduling to solve the massive access problem of IoT," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8572–8589, Sep. 2020.

[30] A. D. Shoaei, D. T. Nguyen, and T. Le-Ngoc, "A reconfigurable access scheme for massive-MIMO MTC networks," *IEEE Access*, vol. 9, pp. 65547–65559, 2021.

[31] E. Gelenbe, M. Nakip, D. Marek, and T. Czachorski, "Diffusion analysis improves scalability of IoT networks to mitigate the massive access problem," in *Proc. 29th Int. Symp. Model., Anal., Simul. Comput. Telecommun. Syst. (MASCOTS)*, Nov. 2021, pp. 1–8.

[32] M. Stusek, K. Zeman, P. Masek, J. Sedova, and J. Hosek, "IoT protocols for low-power massive IoT: A communication perspective," in *Proc. 11th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, 2019, pp. 1–7.

[33] F. Peper, K. Leibnitz, K. Theofilis, M. Hasegawa, N. Wakamival, C. Tanaka, J.-n. Tcrarnae, S. Sekizawa, and A. Li, "On high-density resource-restricted pulse-based IoT networks," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

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

[35] F. Peper, K. Leibnitz, M. Hasegawa, and N. Wakamiya, "Spike-based communication networks with error correcting capability," *Brain Neural Netw.*, vol. 25, no. 4, pp. 157–164, 2018.

[36] C. Tanaka, A. Li, F. Peper, K. Leibnitz, K. Theofilis, N. Wakamiya, and M. Hasegawa, "Implementation of pulse-based multiplexing protocol for massive IoT," in *Proc. Int. Symp. Nonlinear Theory Appl. (NOLTA)*, 2020, pp. 346–349.

[37] C. Tanaka, K. Honda, A. Li, F. Peper, K. Leibnitz, K. Theofilis, N. Wakamiya, and M. Hasegawa, "Performance evaluation of pulse-based multiplexing protocol implemented on massive IoT devices," *Nonlinear Theory Appl., IEICE*, vol. 12, no. 4, pp. 726–737, 2021.

[38] 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.

[39] E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: Throughput and delay analysis," *Comput. Commun.*, vol. 25, no. 3, pp. 313–321, Feb. 2002.

[40] C. H. Foh, M. Zukerman, and J. W. Tantra, "A Markovian framework for performance evaluation of IEEE 802.11," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, p. 1276, Apr. 2007.

[41] J. Mišić, S. Shafi, and V. B. Mišić, "The impact of MAC parameters on the performance of 802.15.4 PAN," *Ad Hoc Netw.*, vol. 3, no. 5, pp. 509–528, Sep. 2005.

[42] K. Leibnitz, N. Wakamiya, and M. Murata, "Modeling of IEEE 802.15.4 in a cluster of synchronized sensor nodes," in *Proc. Int. Teletraffic Congr. (ITC)*, Sep. 2005, pp. 1345–1354.

[43] S. Pollin, M. Ergen, S. C. Ergen, B. Bougard, L. V. Der Perre, I. Moerman, A. Bahai, P. Varaiya, and F. Catthoor, "Performance analysis of slotted carrier sense IEEE 802.15.4 medium access layer," *IEEE Trans. Wireless Commun.*, vol. 7, no. 9, pp. 3359–3371, Sep. 2008.

[44] F. Shu and T. Sakurai, "A new analytical model for the IEEE 802.15.4 CSMA-CA protocol," *Comput. Netw.*, vol. 55, no. 11, pp. 2576–2591, Aug. 2011.

[45] I. M. Sobol, "Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates," *Math. Comput. Simul.*, vol. 55, nos. 1–3, pp. 271–280, Feb. 2001.
[46] A. Saltelli, "Making best use of model evaluations to compute sensitivity indices," *Comput. Phys. Commun.*, vol. 145, no. 2, pp. 280–297, May 2002.
[47] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola, "Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index," *Comput. Phys. Commun.*, vol. 181, no. 2, pp. 259–270, Feb. 2010.
[48] (2023). *Global Sensitivity Analysis Package (v1.2.0)*. Accessed: May 2, 2024. [Online]. Available: https://docs.juliahub.com/General/GlobalSensitivityAnalysis
[49] J. Herman and W. Usher, "SALib: An open-source Python library for sensitivity analysis," *J. Open Source Softw.*, vol. 2, no. 9, p. 97, Jan. 2017.
[50] T. Iwanaga, W. Usher, and J. Herman, "Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses," *Socio-Environmental Syst. Model.*, vol. 4, p. 18155, May 2022.
[51] (2018). *LsqFit Package (v0.15.0)*. Accessed: May 14, 2024. [Online]. Available: https://julianlsolvers.github.io/LsqFit.jl/latest/
[52] 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.
[53] R. Verhoeven, S. Kempinski, and N. Meratnia, "Performance evaluation of Wi-Fi HaLow, NB-IoT and LoRa for smart city applications," in *Proc. 19th ACM Int. Symp. Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw. 19th ACM Int. Symp. Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw.*, New York, NY, USA, Oct. 2022, pp. 17–24.

**KENJI LEIBNITZ** received the M.Sc. and Ph.D. degrees in information science from the University of Würzburg, Germany, in 2003. In 2004, he joined Osaka University, where he became a specially appointed Associate Professor, in 2006. In 2010, he moved to the National Institute of Information and Communications Technology (NICT), as a Senior Researcher, and he has been the Principal Investigator with the Center for Information and Neural Networks, NICT, and Osaka University, since 2013. His research interests include modeling and performance analysis of communication networks, especially biologically and brain-inspired mechanisms for self-organization in future networks.

**FERDINAND PEPER** received the M.Sc. degree in mathematics and the Ph.D. degree in computer science from Delft University of Technology, The Netherlands, in 1985 and 1989, respectively. In 1990, he joined the National Institute of Information and Communications Technology (NICT), Japan, (then named Communications Research Laboratory), as a Researcher, and he has been with NICT, since then. His research interests include artificial neural networks, cellular automata, nanocomputer architectures, asynchronous systems, distributed computing, noise and fluctuations, wireless sensor networks, and error-correcting codes.

**MIKIO HASEGAWA** (Member, IEEE) received the B.Eng., M.Eng., and Dr.Eng. degrees from Tokyo University of Science, Tokyo, Japan, in 1995, 1997, and 2000, respectively. From 1997 to 2000, he was a Research Fellow at Japan Society for the Promotion of Science. From 2000 to 2007, he was at the Communications Research Laboratory, Ministry of Posts and Telecommunications, which was reorganized as the National Institute of Information and Communications Technology, in 2004. He is currently a Professor with the Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science. His research interests include mobile networks, cognitive radio networks, chaos, neural networks, machine learning, and optimization techniques.

**NAOKI WAKAMIYA** (Member, IEEE) received the M.E. and Ph.D. degrees from Osaka University, Osaka, Japan, in 1994 and 1996, respectively. He has been a Research Associate with the Graduate School of Engineering Science and the Educational Center for Information Processing, an Assistant Professor with the Graduate School of Engineering Science, and an Associate Professor with the Graduate School of Information Science and Technology, Osaka University, where he became a Professor, in 2011. His research interests include biologically and brain inspired information and communication technology and self-organizing network control.