2025 IEEE 22nd International Conference on Mobile Ad-Hoc and Smart Systems (MASS) | 979-8-3315-6599-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/MASS66014.2025.00045

SFMAC: Bleeps that enable high-density LoRaWANs

Teresa Blanco Abad*, Vijay Rao**, Nikos Kouvelas†, Venkatesha Prasad*, Kumar Ramamoorthy**, Sujay Narayana*
*Networked Systems, TU Delft, NL / †IMEC Holst Centre, NL / **Cognizant, NL
Contact Author: r.r.venkateshaprasad@tudelft.nl

***Abstract*—LoRaWANs, a widely accepted IoT connectivity solution, adopt a simple (ALOHA-like) MAC layer, enabling low-power communication at the cost of scalability due to packet collisions. Hence, current studies on LoRaWAN conclude that the network does not support dense deployments. Several alternative MACs are proposed but they stumble upon well-known limitations: time division eliminates the asynchrony of LoRa nodes but requires feedback from the gateways; carrier-sensing-based protocols are heavily constrained by the reduced sensing ranges of the devices, thus creating a large number of hidden terminals, leading to collisions.**

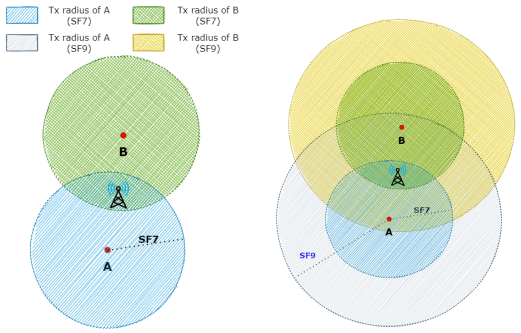
To enhance LoRaWAN to cater to both low- and high-density deployments, in this paper, we propose Spreading Factor MAC (SFMAC), a novel, practical, distributed, and energy-efficient MAC protocol. SFMAC, a channel-sensing-based MAC, takes an unconventional approach to eliminate hidden terminals – by operating with pairs of SFs, wherein the higher SF is used for channel sensing and the lower for data transmission. Bleeps are transmitted in the higher SF as they can be sensed at longer ranges. SFMAC does not require any change in hardware or the LoRaWAN protocol. We demonstrate that the fundamental trade-off made by SFMAC – utilizing two SFs per data transmission instead of using all for data – works extremely well due to the elimination of hidden terminals. Through real-world experiments on 30 SX1261 devices and data-driven *ns-3* simulations, we showcase that SFMAC increases goodput and channel utilization by manifolds over state-of-the-art protocols such as p-CARMA, np-CECADA, and LMAC.

I. Introduction

The proliferation of the Internet of Things (IoT) with miniaturised battery-powered sensing devices for various use cases, such as ambient monitoring in semi-public places, smart cleaning and hygiene solutions, and asset tracking in warehouses, has given rise to a variety of communication technologies [1]. Among these, Low-Power Wide Area Networks (LPWANs), particularly Long Range Wide Area Networks (LoRaWANs), are becoming increasingly popular due to robust low-power communication, less infrastructure needed, and simpler deployment compared to short-range technologies such as WiFi or BLE for these use-cases.

A key metric to economically deploying a LoRaWAN-based IoT solution is the capacity per gateway, i.e., the number of end devices (or nodes) that can successfully communicate with the gateway over a defined period. This should not be taxing the energy-constrained devices nor be compromising the reliability of data transfer.

When deploying LoRaWANs, the most common deployment strategy is to perform extensive site surveys to place gateways. The placing ensures that every node is covered by at least 2 gateways, with an average of 3 gateways coverage [2]. With inexpensive LoRaWAN devices, applications such as asset tracking, asset monitoring, and logistics are transforming as these devices provide long-range and real-time coverage, which seems a much better alternative to the traditional RFID tracking solutions [3], [4], [5]. This simply increases the number of tags per deployment to several thousand, creating high-density deployments. This directly translates to: ❶ increased capital expenditure (CAPEX), with each industrial-grade gateway costing several hundreds to thousands of dollars, and ❷ increased operational expenditure (OPEX) due to the increased number of messages per gateway in network servers. Typically, a third-party network server charges per gateway message processed. While the overall solution is still cheaper than other LPWAN technologies, there is a need to reduce the costs further.



(a) (b)

Fig. 1: (a) Node B is a hidden terminal to node A as sensing and data transmission are on the same SF. (b) Node A's bleep on higher SF can be sensed by node B, and thus B is no longer a hidden terminal for data transmission in lower SF.

The problem. The fundamental reason for deploying multiple gateways is to increase the reliability of message reception by at least one of the gateways as the nodes adopt LoRaMAC, which is Aloha-like, to transmit messages. While this medium access consumes extremely low energy, it is not energy-efficient as it is prone to collisions, with a maximum throughput of 0.18 for an offered traffic load of 0.5, especially for deployments with a high density of nodes. Note that a

2155-6814/25/$31.00 ©2025 IEEE
DOI 10.1109/MASS66014.2025.00045

high offered load (or traffic) can translate to a high density of nodes. Although adding more gateways is the typical solution to support a higher number of nodes in LoRaWAN, it will not always work for a high-density deployment, as it is bounded by the level of the reduction in transmission range that can be applied [6]. The need to reduce the number of collisions, i.e., receiving more frames successfully at a gateway, is critical to the working of the solution, reduction of the overall costs, and LoRaWAN to be more usable and successful.

Carrier-sensing-based medium access (CSMA) has been extended to LoRaWAN using the Carrier Activity Detection (CAD) through LMAC [7], p-CARMA [8], np-CECADA [9] and a lightweight collision avoidance mechanism presented in [10]. However, the improvement is incremental and they fail to scale up due to: ❶ a large number of hidden terminals because of the long transmission ranges of LoRa; and ❷ inaccuracies in carrier sensing on the nodes as they just employ preamble detection, and not payload detection in CAD.

Wastage leads to opportunity. In practical deployments, LoRaWAN devices and gateways are configured to use lower SFs, mostly SF7, due to the lower Time-on-Air (ToA) (implying lower energy consumption by nodes), and the higher SFs (SF9-12) are simply unused. These higher SFs consume more energy per transmission, and due to the increased ToA, their chance of collision is higher, leading to lower goodput.

The main deterrence for CSMA, i.e., hidden terminals, can be eliminated if the transmissions from a node are detectable at twice its transmission range. This is achievable if we employ a suitable, unused higher SF for beaconing an ongoing transmission in a lower SF. For example, as shown in Fig. 1b, node A sends very short bleeps on SF9, which can be heard by node B, and then A transmits data on SF7.

SFMAC. In this article, we propose to utilise higher SFs as a *control channel* to broadcast bleeps that reserve transmissions in the *data channel* (e.g., SF7), thereby reducing hidden terminals. This novel MAC, called SFMAC (Spreading Factor-MAC), can work on the current LoRa nodes with a single half-duplex radio and does not require any hardware design modifications. SFMAC avoids collisions as much as possible and packs the transmissions from various nodes to significantly increase the goodput and channel utilisation.

Challenges. While the idea of SFMAC is neat, several practical challenges must be overcome.

① **Bleeps.** Since we consider a COTS LoRa node that is single-duplex and can only transmit on one SF at a time, it is not possible to protect the data channel transmission whilst transmitting a bleep on the control channel.

② **Energy.** Several energy overheads are introduced: CAD operations for carrier sensing, transmitting a bleep in a higher SF, and energy for switching SFs.

③ **Parameters.** The choice of the control channel, the length of the bleep, SF switching overheads, and CSMA parameters such as the duration of channel sensing and the contention window sizes influence the performance of SFMAC, and must be carefully chosen.

④ **Complexity.** The protocol must be lightweight so that it can be implemented on resource-constrained LoRa nodes easily.

⑤ **Interoperability.** The protocol must be interoperable with existing LoRa gateways.

Contributions and Organisation. In this work, we propose SFMAC, a novel, distributed, energy-efficient, carrier-sensing-based MAC protocol for high-density and/or high-traffic deployments of LoRaWAN. To the best of our knowledge, SFMAC is the first-of-its-kind CSMA-based MAC protocol that can significantly reduce hidden terminals due to its dedicated long-range control channel. Specifically, our contributions:

1. Novelty. SFMAC is the first protocol that avoids hidden terminals. Note that SFMAC works with the single RF frontend that exists on today's nodes and gateways, even though it utilises two SFs. Furthermore, SFMAC is energy-efficient as it packs the transmissions on the data channel one after the other neatly, almost similar to a TDMA protocol, albeit without any need for synchronization among the nodes and/or the gateway. Lastly, SFMAC requires no changes to gateways and can be deployed with current LoRa hardware.

2. CAD. While most prior work utilized SX127x devices that support only preamble detection in CAD and are quite inaccurate in sensing the medium, we base our work on the newer chipsets - SX1261 - that support the detection of LoRa symbols even in payloads. In Sec. III, we characterise the performance of CAD with SX1261 and present our findings briefly due to paucity of space. To the best of our knowledge, this is the first time SX126x devices have been characterised.

3. Evaluation. To evaluate SFMAC, we conduct both data-driven simulations on ns3 [11] using Antwerp dataset [12] (in Sec. V) and real-world experiments (in Sec. V-B). Our simulation model also considers the capture effect, which is an important physical phenomenon in LoRa. We compare with the most recent works - LMAC, np-CECADA and p-CARMA. SFMAC outperforms all these works; SFMAC achieves a 2.08× channel utilisation compared to LoRaMAC when the offered load, G, is 2 while saving 0.75× the energy in collisions. Even when compared to np-CECADA, SFMAC gains about 1.5× in Packet Reception Ratio (PRR).

A make-or-break question for SFMAC is "**would the reduction in the number of data channels by half and elimination of hidden terminals provide better goodput than utilising all the SFs for data?**". We show that this indeed is the case for one data-control channel pair of (SF7, SF9), wherein the combined throughput of LoRaMAC on these SFs is far lower than what SFMAC achieves with the pair. Furthermore, SFMAC can be extended with other possible combinations to operate simultaneously.

II. Related work

CSMA-based protocols are well-established in wireless networks, including low-power networks such as wireless sensor networks (WSN) [13], [14]. However, LoRaWAN is quite different from these due to: ❶ the nature of the protocol - In WSNs, adopting RTS/CTS transmissions can eliminate hidden terminals, however, this is not possible in LoRaWAN since the protocol is minimalistic and gateways do not coordinate

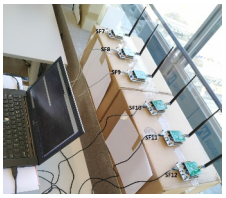


Fig. 2: Placement of receiver LoRa nodes at 22 m and 30 m altitude to perform CAD on different spreading factors.

with the nodes, and thus, collisions can happen. ❷ LoRaWAN uses an unlicensed spectrum, which has duty-cycle constraints in several parts of the world. This limits the number of bytes that can be transmitted by a node. ❸ The longer range of LoRaWAN implies more hidden terminals in a high-density deployment. ❹ Since LoRa is a spread-spectrum-based physical layer, the transmissions can be below the noise floor with low signal-to-noise ratios [15], [16], [17]. Detecting ongoing transmissions can be challenging even with CAD. Therefore, a tailor-made CSMA mechanism is necessary to increase channel efficiency and/or to scale the deployed nodes.

Tobagi and Kleinrock [18] proposed to dedicate a channel for transmitting busy-tone signals to eliminate hidden terminals. Haas [19] proposed Busy Tone Multiple Access (BTMA) with two RF front-ends to achieve this. Though BTMA has inspired us, we have improvised to use a single RF front-end and switch between the control and data channels. Given the constraints of a LoRaWAN network and end devices, BTMA has to be redesigned, which is the core of this work.

The most relevant works in LoRaWAN are the ones based on carrier-sensing-based MAC protocols. Pham adapted the CSMA of IEEE 802.11 into LoRaWAN by utilizing the carrier activity detection (CAD) feature on the nodes to sense the channel [20]. However, the mechanism is not tailor-made for LoRaWAN - energy consumption to perform many CADs combined with the unreliability of CAD, which makes it impractical to employ in a real-world deployment. Kouvelas et al. proposed p-CARMA [8], a lightweight MAC protocol for LoRaWAN that uses p-persistent CSMA, where each device transmits with a fixed probability after sensing the channel. This approach reduces collisions and improves scalability, enabling support for thousands of devices per gateway without centralized coordination. Further, Kouvelas et al. proceeded with np-CECADA [9], a LoRaWAN MAC protocol that leverages a non-persistent access strategy optimized using the capture effect, enabling greatly improved packet delivery and channel utilization in dense IoT deployments—achieving up to 15× better PRR than standard LoRaWAN and 5× over p-CARMA. Gamage et al. proposed LMAC and its variants [7], which adopt collision-evading mechanisms by choosing a channel that may probably be free, based on partial statistics. LMAC-2 [7] claims to achieve much higher throughput and almost zero losses. However, the only reason they perform well is due to the low traffic rate that is spread across eight channels and multiple SFs. Indeed, this is true even in the case of ALOHA when the offered load is less than 0.2. When normalized, LMAC-2 over a single channel performs worse than p-CARMA and np-CECADA. It is only slightly better than ALOHA. Furthermore, all these protocols, considering LMAC-3 [7] variant of LMAC-2, require assistance from the gateway through downlink messages to perform well.

TABLE I: Percentage of successful CAD over distances on different spreading factors in experiment 1

	2.5 km	4.3 km	5.5 km	6.2 km	7.5 km	8.52 km
SF7	100	56	23	0	0	0
SF8	100	79	47	17	0	0
SF9	100	96	63	37	20	2
SF10	100	100	94	71	49	8

SFMAC, on the other hand, is designed for a high-density deployment of LoRa nodes. SFMAC does not need any assistance from the gateways and hence can operate in existing deployments. SFMAC is the only protocol for LoRaWAN that can avoid hidden terminals to a very large extent.

III. Characterisation of CAD

The working of CAD is described in [8], [9]. Most of the literature employ older SX127x LoRa radios that could only detect the preamble, which is inaccurate and insufficient to design a carrier-sensing MAC. We employ the newer SX1261 radios that can detect payload symbols as well [21]. Since SX1261's efficiency in detecting payload symbols has not been characterized in the literature, we undertake experiments and list the most significant observations.

Experiments: We perform two sets of experiments, both in non-line of sight conditions, by placing LoRa receivers at 30 m and 22 m height, respectively, as shown in Fig. 2. In the first experiment, measurements are made at steps of 100 m and the second at 50 m with a transmitter that was set to send 100 packets at different SFs 7-10, one after the other. To obtain the best performance of CAD, the CAD symbol was set to 2 for SF 7-9, and 4 for SF10 [21]. The preamble size was set to 8 symbols, and the payload size was 20 bytes in all the cases. The frequency was set to 868.1 MHz, coding rate to 4/5, transmission power to 14 dBm, and bandwidth to 125 kHz.

The percentage of successful CAD achieved by the receiver at different spreading factors over distance from experiment 1 is listed in Table I. The ratio of successful CAD detection over distance from experiment 2, which is a more fine-grained experiment, is shown in Fig. 3. While there are several inferences about CAD are already presented in the literature [8], [20], [9], there are three key takeaways from these experiments that have not been reported in the literature: (a) Higher SFs are more robust than lower SFs and so the inaccuracies in CAD are lower even at longer distances; (b) it is sufficient to perform CAD on any part of ongoing transmission to detect it; and (c) it is sufficient to reliably detect with just 2 symbols for SF7-9 and 4 symbols for SF10 and beyond. We utilise this information to design SFMAC in the next section.

IV. Design of the SFMAC protocol

The crux of SFMAC is to utilise a dedicated control SF, one of low data rate (i.e., high SF), to transmit control packets (or

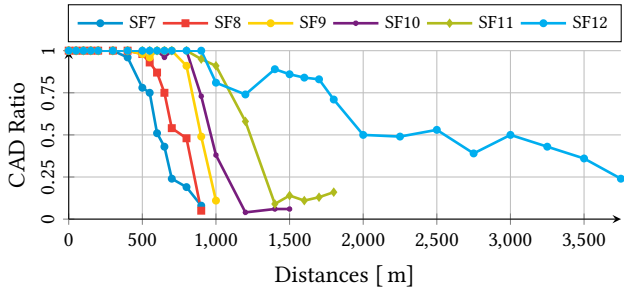


Fig. 3: Ratio of successful CAD detection in experiment 2.

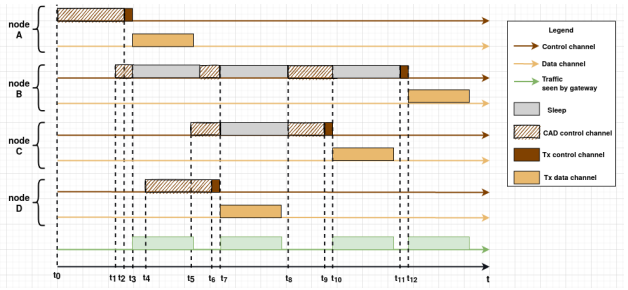


Fig. 4: SFMAC operation.

bleeps), while a high data rate SF (i.e., low SF) is used for data packets. The idea is to find multiple pairs of data and control SFs per frequency band, e.g., (SF7,SF9), (SF8,SF10), etc. A bleep is just random bits and does not contain any identifiable information and is for a short duration.

A. Overview of SFMAC

In the deployment phase of the network, the data and control SF pairs are chosen along with the other parameters, such as the duration of the bleep and CAD, and the contention window size. Since the deployment is for a single IoT solution, we consider that the packet lengths are the same for all transmissions. This is both (a) practical, as most nodes in a deployment perform the same tasks (in our deployment experience), and (b) energy-efficient, as the sleep duration is fixed to the length of the packet and the bleeps need not contain any valid data; therefore, no energy is required to decode bleeps. If the packet lengths vary, they could be either fragmented into multiple fixed-length packets and later assembled at the application server, or assigned different lengths to be used on different SF pairs. The working of SFMAC is simple in the operational phase of the network: When a node has a packet to send, it checks on the control SF for any bleeps. If the control SF is idle, then the node continues to perform CAD for the length of the packet. If still idle, then the node transmits a bleep followed by data on the data SF. However, if a bleep is heard, then the node goes to sleep for the duration of the packet transmission. After waking up, it generates a random sensing duration; if the channel is found empty after the sensing duration, then the node captures it by transmitting a bleep.

Bootstrapping scenario. To better understand the working, consider the example shown in Fig. 4. Consider node A is the first and the only node in the network to have a packet to send. Node A switches to control SF and performs CAD to identify any bleep for a duration equal to the duration of the ToA of a packet. On finding none, node A transmits a bleep and begins to transmit the packet on the data SF.

Long-run scenario. While node A was busy capturing the control SF, node B generated a packet to send at t_1. It finds the bleep from node A and therefore goes to sleep for the duration of ToA of the packet. During the packet transmission of A, nodes D and C generated packets too at t_4 and t_5. Now nodes B, C, and D are contending for the channel. Since nodes C and D did not hear the bleep of A, they began to listen for ToA of the packet. Node B would wake at the end of A's transmission and join C and D to listen to the control SF. Note that B would have generated a random value for the length of the CAD period, chosen between a minimum ($C_W^{\min}$) and maximum ($C_W^{\max}$). Node D would complete the CAD successfully and capture the channel by transmitting a bleep at t_6. At this point, both B and C would sleep until D finishes its transmission. Now, both B and C would wake up, generate random values for CAD duration and begin sensing. As node C would have the least duration for sensing it wins the channel at t_9. At this point, node B sleeps and wakes up after C completes its transmission. The steps repeat - generate a random CAD duration, sense and finally capture the channel.

As mentioned in the example, a node that could not capture the channel at the first instance would generate a random value within the CAD duration window. Upon expiry of this CAD duration successfully, i.e., the control SF was idle for this duration, the node will transmit a bleep and occupy the data SF. This is to prevent any synchronisation among two or more contending nodes that would result in a collision. Note that the switching from control SF to data SF requires only about 20μs which is negligible and not considered in our work.

B. Choosing the control SF

The critical requirement to select a corresponding control SF for a data SF is that the range of CAD detection in the control SF must be close to double the range of the data SF. For instance, if SF7's maximum transmission range is 500 m, the control SF must provide a range of at least 1 km. Such a distance will allow for minimizing the potential hidden terminals to transmission on the data SF. A viable list of candidate control SFs can be generated based on this requirement. The next important parameter is energy consumption. The higher the SF, the higher the energy consumed for both CAD and bleep transmissions. Based on the experiments in Sec. III, we found that 2 symbols are sufficient for SF7-9 and 4 symbols for SF10-12. The energy for bleep transmission depends on the length of the bleep and the SF. Choosing the length of the bleep will be discussed in Sec. IV-C below.

From an energy perspective, the lowest SF in the viable list of control SFs must be chosen. This is subject to the condition that the SF is not already utilised in another SF pair.

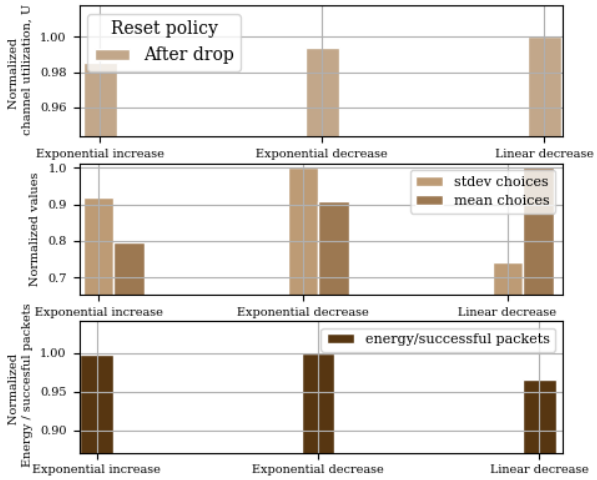


Fig. 5: Evaluation of update policies. The standard deviation, mean choices, collided, and dropped packets are normalized with respect to the highest value per metric considered.

C. Duration of bleep

To keep the energy overheads low, the duration of the bleep must be kept low. Therefore, the minimum length of the bleep must at least be equal to the minimum number of symbols for a highly accurate CAD (i.e., 2 for SF7-9 and 4 for SF10-12).

If a node hears a bleep from another node, the listening node knows how long to sleep as the packet size is preset. Furthermore, a node waking in the middle of a data transmission would sense the control SF to hear a bleep for the duration of the packet ToA. This eliminates the need for a continuous bleep transmission, as in the classical BTMA [18], [19], and the minimum length for bleep suffices.

The design choice of using a short bleep and long CAD duration arises from the fact that the energy to transmit a bleep for the entire packet length would be more expensive than performing CAD for the same duration.

D. Duration of channel sensing

As stated earlier, we consider that the data packet size is predetermined and would be the same across the deployment. Given the time-on-air (ToA) of the data packet, D, on the data SF, and the ToA of the bleep, d, we can calculate the ratio, N, between D and d. N is rounded off to the nearest integer. Therefore, the duration of data transmission in the control SF will be equal to Nd. To prevent any overlaps due to round-off errors, the maximum duration of channel sensing, X, should be equal to $(N+1)d$. The minimum duration of channel sensing is 1, considering the case that a bleep is sensed in the first CAD cycle.

E. CAD duration window and number of attempts

In SFMAC, CAD duration windows are used when a node fails to capture the channel initially. A random number is chosen between $C_W^{\min}$ and $C_W^{\max}$; if the channel remains idle during this period, the node captures it.

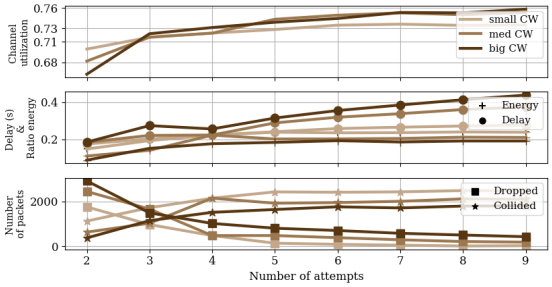


Fig. 6: Channel utilization based on the number of attempts and window size. Small, medium and big window size is respectively defined as $C_W^{\max} = 5$, $C_W^{\max} = 10$ and $C_W^{\max} = 15$.

SFMAC aims to serialize transmissions and reduce collisions. If multiple nodes pick the same CAD duration, they are allowed to transmit, leveraging LoRa's capture effect and multi-gateway reception. Each node can attempt to capture the channel up to M times, after which packets are dropped. Several CAD duration window policies are possible: ❶ $C_W^{\max}$ is fixed and preset. ❷ $C_W^{\max}$ increases linearly or exponentially each time the node fails to capture the channel. ❸ $C_W^{\max}$ is initially set to the maximum value and decreases linearly or exponentially on each failed attempt. Fixed values perform poorly; thus, we evaluate linear and exponential updates in Fig. 5.

Simulations using ns3 [22] assume $C_W^{\min} = 2$ and $M = 4$, with $C_W^{\max} = 16$ for exponential policies. For linear decrease, $C_W^{\max} = 10$ and $C_W^{\min} = 4$ ensure comparable windows at $m = 2$. Linear decrease yields the lowest normalized standard deviation due to a smaller initial window, reducing collisions. Thus, it is chosen based on energy per successful packet—4% lower than with exponential decrease (Fig. 6).

Fig. 6 also shows that channel utilization saturates after 5 attempts. While larger windows improve utilization, they increase delay. To balance energy and delay, the retry limit is set to 5; packets are dropped after that.

V. Performance Evaluation

To obtain insights into the performance of SFMAC, we undertake evaluation using both simulations and real-world experiments. We utilise the offered load, G, to introduce varying traffic loads. *Density can be seen as a converse of traffic load*, i.e., the higher the density, the higher the load on the channel. A value of G = 2 implies 18 packets per second are generated. We compare SFMAC with the state-of-the-art protocols LoRaMAC, p-CARMA [8], LMAC [7] and np-CECADA [9]. Along with these, we also introduce CADMAC for comparison; a node with CADMAC does a channel assessment when a packet is to be sent. If the channel is free, the packet is transmitted; otherwise, the node backs off to a pre-determined window and retries again. After two retries, the packet is simply transmitted regardless of the channel state. This is a best-effort transmission, hoping

TABLE II: Parameters of simulation

Parameter	value	Parameter	value
Frequency of channel (MHz)	868.1	Radius (m)	500
Number of devices, N	500	Topology	Circle
Number of gateways	1	Packet size (B)	49
Coding rate, CR	4/5	SF of transmissions	7
Poisson rate (1/s)	λ = G / ToA	SF of control channel	9
Transmission rate per node (1/s)	r = λ / N	ToA (s)	0.0975

the packet is received due to the capture effect. Note that CADMAC works per SF and does not consider pairs.

A. Simulation based evaluation

We use the well-known open-source ns3 simulator with LoRaWAN implementation [11]. Our parameters of the simulation are listed in Table II. The LoRa devices are distributed uniformly around the gateway in a circle. Simulations are run for 1000 rounds with each round generating more than 64,000 packets.

For all SFMAC evaluations in the rest of the paper, we consider SF7 as the data SF and SF9 as the control SF, which are chosen based on the steps in Sec. IV. The bleep duration is set to 2 symbols. Given the orthogonality of the SF, the results can easily be extrapolated when using other SF pairs within the same frequency band and bandwidth, and across other bands/bandwidth combinations.

1) Evaluation without capture effect: The protocols are evaluated by varying the offered load, G. As seen in Fig. 7a, SFMAC achieves a $10\times$ improvement in channel utilization over LoRaMAC at $G = 2$. The corresponding PRRs are shown in Fig. 7b. At $G = 0.5$, LoRaMAC achieves 0.18 channel utilization—the theoretical ALOHA limit [23]. With a low load, CADMAC also performs well.

At $G \geq 1$, the performance of CADMAC and LoRaWAN degrades. $G = 1$ has an average rate of 10 pkts/s and 97.5 ms time-on-air, and SFMAC negins to drop packets to reduce congestion. Note that perfect TDMA is needed to achieve 100% utilization at $G = 1$. Goodput trends similarly to utilization in Fig. 7a.

SFMAC's fairness is shown by the compact PRR quartiles in Fig. 7b, with a max-min PRR difference of 0.21 at $G = 1.5$. Even at $G = 0.5$, SFMAC achieves $1.12\times$ higher PRR than CADMAC due to reduced collisions. Fig. 7c shows that SFMAC has better energy efficiency than CADMAC and LoRaMAC, despite CAD and bleep overheads, thanks to strategic back-offs. For high G, most packet losses stem from collisions or lack of transmission opportunities.

Table III reveals a key insight: **halving the number of spreading factors does not reduce goodput**. SFMAC consistently outperforms the combined goodput of LoRaWAN using both SF7 and SF9. This finding highlights a significant design opportunity—simplifying the network by limiting SF usage and addressing hidden terminal issues can lead to more efficient resource utilization without compromising performance. Such an approach not only enhances scalability but also challenges the prevailing assumption that maximizing SF diversity is essential for optimal throughput.

TABLE III: Comparison successful packets per second in LoRaWAN using both SF7 and SF9 for data transmission and SFMAC only SF7.

	SF7		SF9	
	LoRaWAN	SFMAC	LoRaWAN	SFMAC
G = 0.5	2.95	4.77	0.87	NA
G = 1	3.67	7.025	1.08	NA
G = 1.5	3.69	7.08	1.09	NA
G = 2	3.57	6.38	1.05	NA

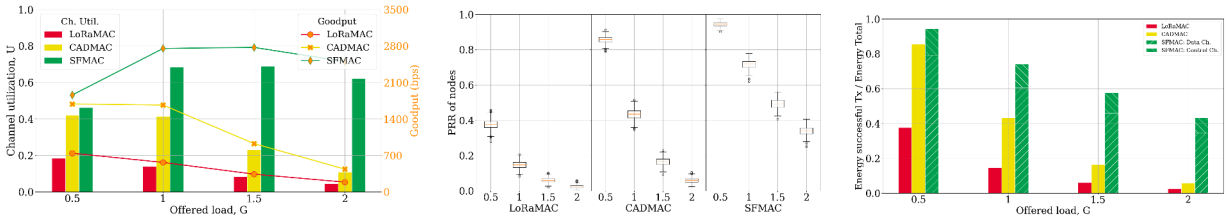
2) Evaluation with capture effect: The capture effect is vital in LoRaWANs where concurrent transmissions occur [24]. To model this realistically, we enhance our ns3 simulation with power and delay captures using a data-driven approach based on the Antwerp dataset [12] and [9].

Network-wide Metrics. SFMAC improves LoRaMAC performance by $2.08\times$ under high load (Fig. 8a, b). At $G = 2$, SFMAC achieves $2.3\times$ higher PRR. While LoRaMAC stabilizes at all G, CADMAC converges to a utilization of 0.36. For $G \leq 0.5$, CADMAC and SFMAC show similar utilization, though CADMAC uses $0.8\times$ the energy per transmission. As traffic increases, SFMAC outperforms CADMAC by up to $1.78\times$, stabilizing beyond $G = 1.5$.

Per-node Metrics. Packet Transmission Ratio (PTR), the ratio of transmitted to generated packets, reflects protocol fairness. Fig. 8c shows that SFMAC drops packets under contention, unlike LoRaMAC and CADMAC which transmit all packets. In LoRaMAC/CADMAC, nearby nodes achieve PRR = 1, while distant nodes drop to 0.1. In contrast, SFMAC ensures a minimum PRR of 0.22 at $G = 2$.

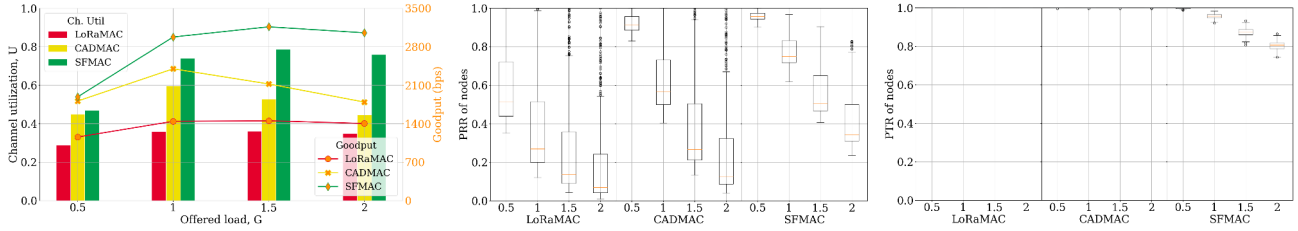
Energy Overhead. Fig. 9a shows SFMAC consumes 37% more energy than LoRaMAC per transmission in worst-case scenarios. However, this excludes energy wasted in collisions. Fig. 9b shows SFMAC reduces collision energy by $6.25\times$ at $G = 0.5$, highlighting the capture effect's benefit. Overall, energy overhead in SFMAC remains minimal.

Comparison with state-of-the-art. Gamage et al., propose a series of protocols - LMAC-1, LMAC-2 and LMAC-3, for LoRaWAN with CAD as a carrier sense mechanism, wherein each protocol is an improvement over the previous one [7]. LMAC-1 adapts the IEEE 802.11's distributed coordinated function for LoRaWAN, wherein the channel is sensed before capturing it. While LMAC-1 chooses a random SF/channel to transmit, LMAC-2 enhances LMAC-1 by probing and selecting the best SF or channel. LMAC-2 uses 16 combinations of 8 channels and 2 SFs, wherein any node may use one from the combination. LMAC-3 is further enhanced by using gateway inputs on the global occupancy matrix to choose the SF/channel to transmit. We will compare it to LMAC-2 since it reports the best performance for class A devices. To make fair comparisons with SFMAC, we scale down LMAC-2 by a factor of 8 as we use two SFs for transmissions. While LMAC uses 16 B transmissions, 50 nodes, and could use in extreme cases 450 CAD, SFMAC uses 49 B payload, 500 nodes, and up to 5 CAD attempts.



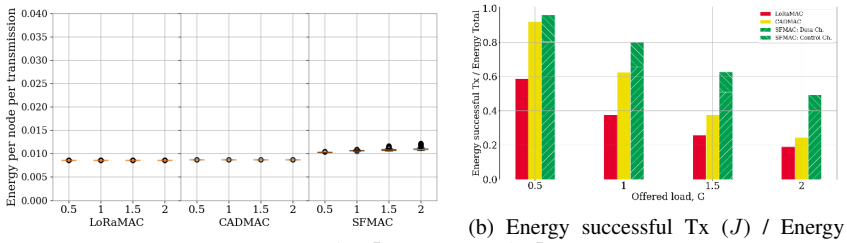
(a) (b) (c)

Fig. 7: Performance of SFMAC measured with various metrics: (a) Channel Utilization and Goodput. (b) PRR, (c) Energy for successful packets/total energy.



(a) (b) (c)

Fig. 8: Comparison between MAC protocols including capture effect: (a) Channel utilization and Goodput, (b) PRR (c) PTR.



(a) Energy per transm. per node (J)

(b) Energy successful Tx (J) / Energy Total (J)

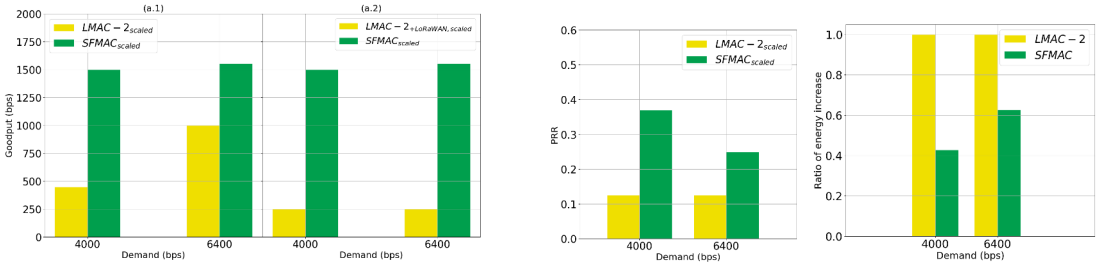
Fig. 9: Comparison of energy between existing MAC protocols considering the capture effect.

Comparison with LMAC. Fig. 10a shows SFMAC achieves 6.25× higher goodput at 4 kbps compared to LMAC-2, confirming the benefit of a control channel under heavy load. SFMAC also delivers 3× higher PRR (Fig. 10b) and consumes only 0.42× more energy than LMAC-2 (Fig. 10c), justifying the SF reduction in SFMAC.

Comparison with p-CARMA and np-CECADA. Fig. 11a shows SFMAC achieves 8.7× better channel usage over p-CARMA at $G = 1.5$, due to direct sensing via a control channel. It also outperforms np-CECADA with 1.4× higher utilization at $G = 1$. In PRR (Fig. 11b), SFMAC achieves 8.5× higher PRR than p-CARMA at $G = 1.5$, and 1.5× higher than np-CECADA at $G = 1$, thanks to efficient backoffs and serialized transmissions. While SFMAC incurs slightly higher switching energy, its total per-byte energy remains modest—only 1.17× that of p-CARMA (37.5 μJ; Fig. 11c). Though np-CECADA saves more energy via reduced transmission power and politeness, it suffers in PRR and PTR due to excessive backoff, dropping many generated packets.

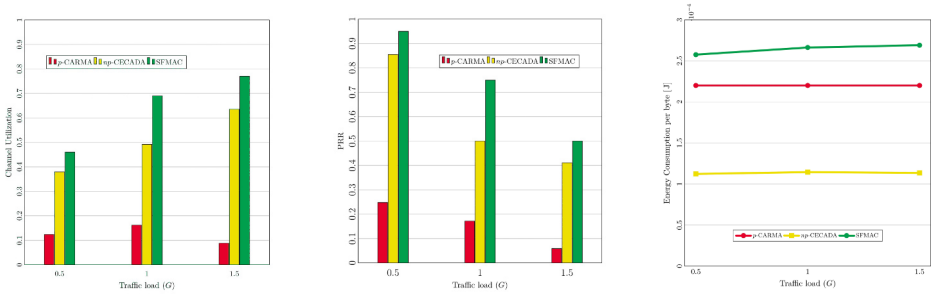
B. Evaluation on testbed

To evaluate SFMAC in actual LoRa deployments on the field, we built a small testbed of 30 LoRa SX1261 *class A* devices (with STM32 Nucleo-F446RE MCU) and one gateway in an indoor environment (with very low power transmissions), emulating a LoRaWAN without interference from commercial operations. As shown in Fig. 12, the devices are deployed in six clusters of five devices in a campus building. We fixed the gateway and extensively measured the RSSI and successful reception of frames as they were transmitted from different locations, mimicking a LoRa network with hidden and non-hidden terminals. We set the location such that each group is visible only by a subset of other groups in terms of SF7-CAD, see Table IV), but at the same time visible by every other group regarding SF9-CAD as it works as the control channel. The above environment, due to its indoor nature, introduces higher fading because of walls and signal attenuation, similar to an open field LoRaWAN with (non-)hidden devices positioned at large distances. To emulate the high-density LoRaWAN, *in*



(a) Goodput (b) PRR (c) Energy

Fig. 10: Comparison between SFMAC and LMAC.



(a) (b) (c)

Fig. 11: Comparison between SFMAC, p-CARMA, and np-CECADA: (a) Channel Utilization, (b) PRR, (c) Energy.

effect, the idea here is to increase traffic as in our simulations. We increased the duty cycle (above 1%) and frame sizes (i.e., Time on Air). We tested one hour of operation under SFMAC and LoRaMAC for G $\in [0.5, 1.0, 1.5, 2.0]$. Several experiments were carried out in order to reach more than 90% level of confidence. We used frames of 220 B and transmission power of 14 dBm for SF7. The rest of the parameters are unchanged.

The PRR of each device is presented in Fig. 13. As observed, when SFMAC is used, at least 28 devices out of 30 for each traffic load report better results regarding PRR. Further, SFMAC distributes the transmissions fairly among the devices as shown by the lesser variations in PRR. On the contrary, the vast majority of devices employing LoRaMAC stay below 40% PRR, and at high traffic loads, i.e., G=2, half of them report almost 0% PRR. The overall PRR is presented in Fig. 14a. SFMAC outperforms LoRaMAC by 2× for G=1 and by 4× for G=2.

In Fig. 14b, the energy consumption of the devices is depicted, showing also the parts of energy spent for successful transmissions and collided frames. SFMAC consumes energy not only to transmit but also for the employment of CAD both at the control and the data channel. In particular, LoRaMAC spends up to 4 times more energy than SFMAC for a traffic load of G=2 for lost packets, since the attempted transmissions are not regulated by the MAC layer. From this energy, the portion that is consumed for successfully received frames is 50% for G=0.5 and 11.4% for G=2.0; the rest is wasted. On the other hand, SFMAC manages 66.4% and 52.6% useful energy consumption for G=0.5 and G=2, respectively.

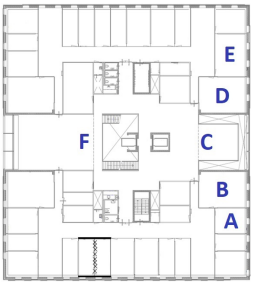


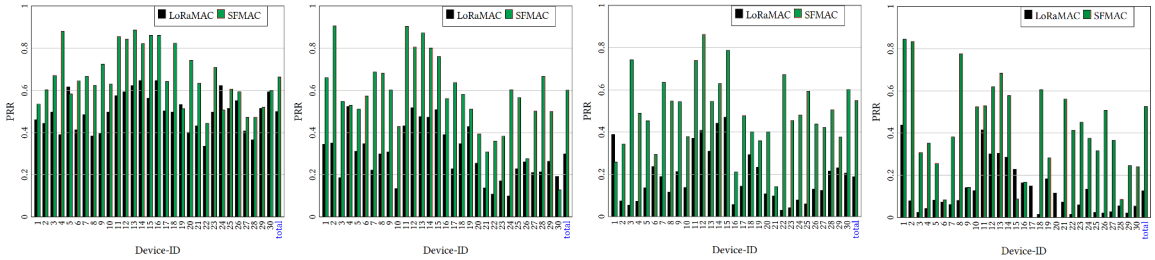
Fig. 12: Thirty devices distributed per five devices per group in six groups. Floor-plan of building with group-positions according to their SF. Gateway positioned with group C.

TABLE IV: Groups of non-hidden devices for each group regarding SF7

Group	Non-hidden	Group	Non-hidden	Group	Non-hidden
A	B, C	B	A, C, D	C and GW	A, B, D, E, F
D	B, C, E, F	E	C, D, F	F	C, D, E

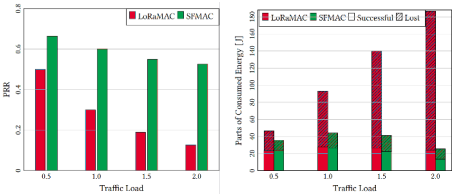
VI. Conclusions and Discussions

With LoRaWAN gaining popularity, scalability per gateway has to increase due to (a) the traditional solution of adding more gateways will not actually help increase packet reception ratio (PRR) beyond a certain point; (b) the cost increases as the number of messages received/gateway increases. Neither the Aloha-like LoRaMAC nor the proposed CAD-based protocols are effective due to the hidden terminals. Therefore, in this



(a) G=0.5 (b) G=1 (c) G=1.5 (d) G=2

Fig. 13: PRR per device: SFMAC versus LoRaMAC and in total for different traffic loads.



(a) (b)

Fig. 14: On field experiments. (a) Overall PRR comparison between SFMAC and LoRaMAC. (b) Energy consumption for SFMAC and LoRaMAC for different traffic loads.

paper, we proposed Spreading Factor MAC (SFMAC), a novel, simple, distributed, and energy-efficient MAC protocol for LoRaWAN wherein higher-SF channels are dedicated strictly to channel sensing (using bleeps), while low-SF channels are used for data transmission for low time on air. SFMAC is a low-complexity protocol, and we demonstrated through simulations and real-world experiments that it outperforms state-of-the-art protocols by several folds. The gains of SFMAC outweigh the main tradeoff: although one SF is dedicated to bleeps, the overall gains in PRR and channel utilization on data SFs, when the channel is saturated, are higher than SF7+SF9 throughputs put together with LoRaMAC. While there are energy overheads in SFMAC, they are demonstrated to be insignificant due to the short bleep transmissions. To achieve significant gains, several data-control SF pairs can be created over orthogonal channels and bandwidths. This would increase the gains of SFMAC simply by several orders of magnitude more than existing MAC protocols in the literature.

ACKNOWLEDGMENT

We acknowledge the EU Project ENACT (https://enact-he.eu/), which has received funding from the European Union under the grant agreement No GA 101157151. Views and opinions expressed are, however, those of authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] Semtech. (2020) Lora technology is connecting our smart planet. [Online]. Available: https://www.semtech.com/lora/lora-applications

[2] N. Matni *et al.*, "Lorawan gateway placement model for dynamic internet of things scenarios," *Sensors*, p. 13, 04 2018.

[3] Abeeway. (2025) Abeeway track and trace. [Online]. Available: https://www.abeeway.com/track-trace/

[4] tektelic. (2025) Orca industrial gps tracker. [Online]. Available: https://tektelic.com/products/sensors/orca-industrial-gps-asset-tracker/

[5] CubeWorks. (2025) Nanotags. [Online]. Available: https://www.cubeworks.io/product/nanotag/

[6] B. Ghena *et al.*, "Challenge: Unlicensed lpwans are not yet the path to ubiquitous connectivity," in *ACM MobiCom*, 2019, pp. 1–12.

[7] A. Gamage *et al.*, "Lmac: Efficient carrier-sense multiple access for lora," in *ACM MobiCom*, 2020.

[8] N. Kouvelas *et al.*, "P-carma: Politely scaling lorawan," in *EWSN*, 2020.

[9] ——, "np-cecada: Enhancing ubiquitous connectivity of lora networks," in *IEEE MASS*, 2021, pp. 374–382.

[10] M. E. Congduc Pham, "Dense deployment of lora networks: Expectations and limits of channel activity detection and capture effect for radio channel access," *Sensors*, 2021.

[11] "ns-3," https://www.nsnam.org/.

[12] M. Aernouts *et al.*, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, p. 13, 04 2018.

[13] S. B. Eisenman and A. T. Campbell, "E-csma: Supporting enhanced csma performance in experimental sensor networks using per-neighbor transmission probability thresholds," in *IEEE INFOCOM*, 2007, pp. 1208–1216.

[14] Y. Tay *et al.*, "Collision-minimizing csma and its applications to wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1048–1057, 2004.

[15] J. Petäjäjärvi *et al.*, "Performance of a low-power wide-area network based on LoRa technology: Doppler robustness, scalability, and coverage," *Int. Journal of Distributed Sensor Networks*, vol. 13(3), 2017.

[16] J. C. Liando *et al.*, "Known and Unknown Facts of LoRa: Experiences from a Large-Scale Measurement Study," *ACM Trans. Sen. Netw.*, vol. 15, no. 2, Feb. 2019.

[17] R. Sanchez-Iborra *et al.*, "Performance Evaluation of LoRa Considering Scenario Conditions," *Sensors*, vol. 18, p. 772, 03 2018.

[18] F. Tobagi and L. Kleinrock, "Packet switching in radio channels: Part ii - the hidden terminal problem in carrier sense multiple-access and the busy-tone solution," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1417–1433, 1975.

[19] Z. Haas and J. Deng, "Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks," *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 975–985, 2002.

[20] C. Pham, "Investigating and experimenting CSMA channel access mechanisms for LoRa IoT networks," 04 2018, pp. 1–6.

[21] Semtech. (2019) Application note: Sx1261 cad performance evaluation. [Online]. Available: https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R000000Q1ES/SPexo9njbhEQLcJVUg1i0Su8p3tpAtwX1jhMBGlXsQI

[22] D. Magrin *et al.*, "Performance evaluation of lora networks in a smart city scenario," in *IEE ICC*, 2017, pp. 1–7.

[23] N. Abramson, "The ALOHA System-Another Alternative for Computer Communications," in *Managing Requirements Knowledge, International Workshop on*, vol. 1. IEEE Computer Society, 11 1970, p. 281.

[24] M. C. Bor *et al.*, "Do LoRa Low-Power Wide-Area Networks Scale?" in *ACM MSWiM*, 2016, pp. 59–67.