CANL LoRa: Collision Avoidance by Neighbor Listening for Dense LoRa Networks

Guillaume Gaillard
E2S UPPA, LIUPPA
Universite de Pau et des Pays de l'Adour
Anglet, France
guillaume.gaillard@univ-pau.fr

Congduc Pham
E2S UPPA, LIUPPA
Universite de Pau et des Pays de l'Adour
Pau, France
congduc.pham@univ-pau.fr

***Abstract*—The current medium access in LoRa, involving strategies very similar to early ALOHA systems, does not scale for future denser LoRa networks, subject to many collisions. Semtech's Channel Activity Detection (CAD) feature enables to implement a carrier sense (CS) in LoRa WANs, but its unreliability at short distance dramatically decreases its efficiency for classical CS strategies. We present CANL, a novel LoRa channel access approach based on an asynchronous collision avoidance (CA) mechanism and operating without the CAD procedure. Extensive simulations using an extended LoRaSim confirm the performance of CANL in a wide range of configurations. The results are promising and show that the proposed CA approach can greatly increase the delivery ratio in dense LoRa networks compared to a classical CS strategy while keeping the energy consumption at a reasonable level.**

***Index Terms*—Channel Access, LoRa, Collisions, Dense Networks, Listen-Before-Talk, Carrier-Sense, LoRaSim**

I. Introduction

The Internet of Things (IoT) gets denser [1], while maintaining requirements of long range, low energy use and ISM band sharing. LPWAN (Low-Power WAN) LoRa networks, currently deployed worldwide, still satisfy these requirements thanks to the unprecedented sensitivity levels that can be achieved at receivers. But as traffic explodes and LoRa end devices (EDs) invade the terrain, the band is saturated as collisions make channel usage inefficient [2]–[4].

In dense networks, many EDs share the same collision domain, i.e. for a given channel, two simultaneous transmissions toward one Gateway (GW) would mostly collide and get lost, except in rare cases of overlap (one capturing the other with less power at reception). The packet losses increase due to collisions. The density even impairs Capture Effect (CE) that could spare the most powerful transmissions in a sparse environment. EDs cannot anymore upload their data efficiently enough.

In LoRaWAN specifications [5], the GW improves the performance using the Adaptive Data Rate (ADR) mechanism to dynamically set LoRa channel parameters (spreading factor SF and bandwidth BW) to EDs, according to SNR and/or Packet Error Rate (PER). However, ADR showed a fairly long convergence time as a GW needs to process many packets [6]. Even ADR does not permit to face the increase of density.

Unfortunately, LoRa standards do not define any specific Medium Access Control (MAC) mechanism as opposed to other radio technologies such as 802.15.4 (ZigBee) or WiFi. It means that a LoRa network behaves as an unidirectional ALOHA system with packet transmissions occurring at anytime, impairing its scalability and Data Extraction Rate (DER) [3], [7]–[10].

Besides, the classical Collision Avoidance (CA) mechanisms used in wireless networks do not work for LoRa because they stand over one of the following:

- Listen-Before-Talk [2], [4], [11] using Channel Activity Detection (CAD) [12]: because transmissions are available below noise level for LoRa, CAD becomes unreliable at long range [13], [14];
- closed-loop collision avoidance: implies answers (ACK, CTS, beacon, etc.) from the receiving GW, which is not tractable for dense LoRa (due to higher energy waste and GW duty cycle).

Replications, as in former Sigfox [1], mitigate the risk of collisions but impair the energy consumption and the channel occupation.

We propose CANL LoRa, a CA mechanism in open-loop, taking advantage of the listening capability of the EDs to avoid collisions with neighboring traffic.

This article is inspired from guidelines we designed in a preliminary experimental work [14]. The proposed approach has been largely improved and trialed in much more detail. Our contribution in this article is threefold: **1)** we design CANL, defining its listening phase, a programmed backoff, and a full compatibility with current standards; **2)** we consider an alternative to bypass current limitations of LoRa radio modules at the expense of an increased delay and energy consumption; **3)** we extend LoRaSim, the well-known simulator [15], [16], to model both a more realistic dense scenario and the mechanisms involved in the proposed protocols. We thoroughly model the network and perform extensive simulations to compare the performance of our proposed approach with the classical Carrier Sense (CS) approach in dense LPWAN traffic scenarios.

II. CANL: A medium access control protocol

In brief, CANL LoRa is an open-loop collision avoidance mechanism implementing a Listen-Before-Talk strategy by setting EDs in LoRa packet reception mode.

A. *Description of CANL LoRa*

Fig. 1 shows three simultaneous CANL data transmissions towards a GW. For each involved equipment, a time line is drawn. Above the time line, EDs are in transmission mode. Below the time line, blue filled blocks correspond to periods where the EDs are in reception mode. Dashed empty blocks represent full-sized listening periods that would have occurred if channel was considered free.

Fig. 2 draws the states and transitions of the proposed protocol, from the moment an ED has data to be sent, up to the end of the actual transmission of the frame. These steps are presented in detail below.

1) Preparation for a transmission: when a data frame is generated, the ED enters the *Want Transmit* state, and increments a transmission attempt counter. When this counter reaches a maximum, the transmission is aborted and the ED goes back to sleep until next data notification. If a new data is generated while the medium access has not been granted, the old data is dropped.

2) Listening phase: during this phase, the ED's radio is put in reception mode, capable of hearing and detecting preamble symbols in its neighborhood, benefiting from the high sensitivity of LoRa packet reception.

The duration of the *Listen* window is uniformly distributed among the EDs between a minimum and a maximum value, both parameters of the protocol. Transmitting without a listening phase corresponds to an ALOHA-like scenario (with no CA at all). Moreover, the EDs need to listen at least the minimum number of preamble symbols enabling a radio chip to detect an incoming frame. The first ED that stops listening wins the access to the medium. The listen duration trades off latency, saturation, resilience to external traffic and noise.

When enough symbols of a preamble are caught, a valid transmission from another ED is heard. The ED will keep listening the time necessary to receive the LoRa header, plus a margin: the end of the planned listening period is postponed by the necessary time to get the header. If the reception is not impaired, e.g. by a collision, the radio chip-set will indeed trigger a *Valid Header* interrupt and catch the explicit header containing the length of the following payload in bytes [12]. Immediately, the ED stops listening and changes its state to *Network Allocation Vector (NAV)*. When no preamble is caught, the channel is considered free and the ED goes to *Send Data*.

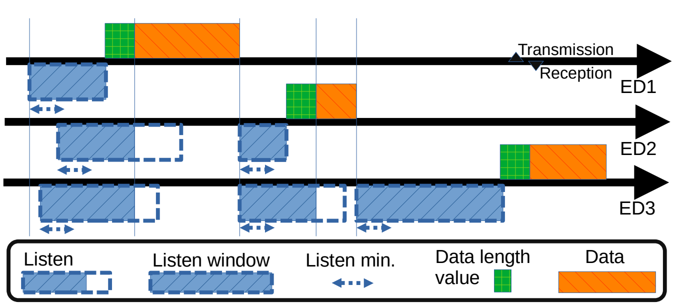

Fig. 1. CANL uses random-sized listening periods and data length fields to avoid collisions.

The use of the RX mode preserves the asynchronous behavior of the communications therefore CANL is well suited for common scenarios with class A EDs. A full CANL transmission individually presents a higher energy consumption than one using the CAD feature [12], but CANL compensates by an increase in reliability and a decrease on the global energy consumption when considering efficiency.

3) Sleeping in NAV state: the ED stays in sleep mode during the whole predicted transmission time. The duration of NAV is set according to the size of data, if known, hence reducing both delay and energy consumption.

If a preamble was heard (no data length has been identified by the listener, but a data frame has been detected), then the duration of NAV is set according to the time on air of a frame with the maximum size, shortened by the duration of a preamble.

After waiting, the ED switches to *Want Transmit* and the complete cycle of access to medium may be started again.

If the payload length was caught by several listening neighbors, e.g. ED2 and ED3 in Fig. 1, their next attempts will be simultaneous. In that case, the randomness of the listening duration makes the ED with the shortest listening phase earn the medium access.

4) Sending the Data: whether successful or not, the transmission ends and the EDs returns in sleep mode until next activity period.

B. *Refinements*

1) We give priority to EDs with already at least one attempt: after the failure of the first transmission attempt, we reduce the upper bound of the *Listen* phase according to the transmission attempt counter, hence statistically favouring the oldest competitor. This gain in fairness decreases the mean latency, hence reducing the drops ans abortions.

The reduction is linear with the number of attempts and a *fair factor* $\mathcal{F}$ expressed in preamble times on air, until the minimum listen duration is reached.

2) We do not perform a CAD before the listen phase: we compared CANL with and without CAD prior to listening in RX mode, and observed that although efficient in very short range scenarios, it negatively impacts CANL in terms of PER, energy and latency in a dense and long range scenario.

C. *CANL_RTS*

To the best of our knowledge, the current Semtech LoRa chip (SX1261/2) does not expose the content of the valid LoRa packet header, despite the existence of the corresponding

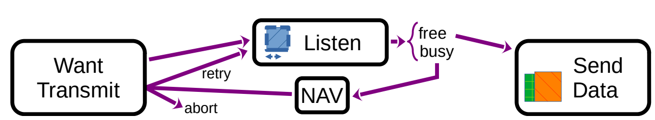

Fig. 2. States of an ED applying CANL's transmission mechanism.

interrupt on the chip. In order to overcome this obstacle that should be tackled in a future version of the firmware, we present *CANL_RTS* that implements an alternative way of obtaining the length of next frame.

Inspired by the RTS/CTS scheme [17], CANL_RTS intercalates an RTS short frame announcing the future data immediately before its transmission and right after the listening period. The RTS includes the size of the next data frame, allowing a neighbor to stop listening after the corresponding *RXDone* interrupt.

Exactly as for CANL, an ED in listening phase that hears an RTS would postpone its transmission, changing to NAV state, with knowledge of the duration of the upcoming data transmission. If the reception of an RTS is interrupted, e.g. by a collision, or if only a preamble is heard, the NAV period is adjusted according to the remaining time on air (max data size, plus RTS).

Although an implicit LoRa header can be used for the RTS, the RTS time on air (including its proper preamble and payload) is higher than the time on air of a data packet header (which is used by CANL), hence increasing both consumption in reception and transmission. In order to reduce the overhead, CANL_RTS only sends an RTS prior to data frames longer than a given size (e.g. 12 payload bytes).

III. A SIMPLE NETWORK MODEL

The dense deployment scenario considers LoRa GWs surrounded by EDs, transmitting at different traffic patterns, coding rates, spreading factors and bandwidth, in presence of varying noise and obstacles.

A. A dense collision domain

We assume that each available *logical* channel (Frequency, SF, BW) works independently of the others. Then, when studying one channel we do not lose generality, but reduce the scale of study (from tens of thousand of EDs to a more reasonable value, e.g. 200 to 1000 EDs) for both simulation and experimental campaigns. In the case of an experimental study, involving e.g. 20 EDs, we anticipate that the change of scale in node density implies a gap in the achieved performance.

We take one single GW for our study, situated at the center of a collision domain. Let it be entirely dedicated to the considered channel. Again, in doing this, we do not lose generality, since the individual performance of several GWs, in a deployment designed for such densely occupied scenarios, could be combined into a global one dedicated to each channel.

We also consider a dense scenario in terms of channel load, where the expected traffic is close to saturate the GW. To emphasize on this aspect, we particularly focus on the worst-case scenario in terms of channel occupation, i.e. SF12BW125: any transmission, whether payload or overhead, takes longer, so the channel is saturated sooner. EDs far from the GW will typically have this setting with ADR and the collision domain will therefore be larger. We chose to maintain the typical 4/5 coding rate though.

We have also considered large and normally distributed payload sizes (e.g. around $60\,\mathrm{B}$). In order to reduce the time on air, one may restrain the maximum data payload size, e.g. to $150\,\mathrm{B}$ instead of the $256\,\mathrm{B}$.

B. Radio model

We adopted a basic well-known, state-of-art propagation model: transmitted power attenuates with distance according to a path-loss exponent [8].

Compared to the original release of LoRaSim [15], we added in our model a noise Gaussian variable n that alters the transmission within a few dBs, and a Rayleigh distributed variable r affecting the reception power, representing fading on a Rayleigh channel. Every new reception power is computed as an instance of the random variables, depending on the distance from the transmitter d, the transmission power P_{TX}^{dB}, and the current noise n and fading level r:

$$P_{RX}^{dB} = P_{TX}^{dB} + G - Lpld0 - \gamma 10 log_{10}(\frac{d}{d0}) - n - r \quad (1)$$

In Eq. 1, the general hardware gains G, the path-loss exponent (PLE) γ, the default path-loss $Lpld0$ and the reference distance $d0$ are constant parameters, as well as the ED transmission power. It is possible to consider different values for these parameters, first for EDs, then for the GW, that may benefit from its higher position and better antenna gains. We computed n and r directly in dB using Python Numpy normal and Rayleigh distributions [18]. n is clipped to 0 such that it never has positive impact on the transmission, while r is reduced by its mean such that its average impact is null.

Above a given sensitivity threshold, specified according to the hardware specifications (e.g. [12] for current LoRa chips), a receiver detects and receives a transmission if a minimum number of symbols are caught.

In reception mode, a transceiver adjusts its filters to the power level it receives on the antenna. In case of multiple simultaneous transmissions, this behavior results in at least all but one losses of data (collision). A frame is lost if any portion of its data symbols is lost. We assume the Capture Effect only spares the most strongly heard signal, if its reception power is a margin above any other's. We model this margin according to the number of competitors heard above the sensitivity threshold. Indeed, we saw experimentally that the capture margin increases with the number of competitors [14].

C. Device model

We assume data frames of variable payload sizes, with explicit LoRa header, and RTS frames of fixed size with implicit ones. Each frame is preceded by a preamble of given duration enabling correct detection of an upcoming transmission by receivers if a minimum of preamble symbols are well-received according to Semtech's specifications [12].

We focus on the radio energy consumption therefore, EDs can be in 4 states: 1) have its radio chip in sleep mode, i.e. consuming 0 mA; 2) transmit, 45mA; 3) be in receive mode, 5.3mA; 4) perform a CAD, 169nAh [19].

IV. Simulation Study

In this section, we compare CANL and its alternative CANL_RTS with ALOHA and CAD+Backoff. CAD+Backoff is the classical Carrier Sense with backoff method and is illustrated in Fig. 3: before transmitting, each ED listens whether a transmission overcomes the noise level. If channel is listened busy (see ED2), the ED applies a random backoff. A transmission can still be unheard, or may occur at the exact same time as other, leading to an overlap at the GW.

An ideal protocol that would schedule generated frame transmissions in a collision-less FIFO queue is also added to serve as reference. The comparisons are made upon variations of one parameter w.r.t. the reference scenario below.

A. Simulation settings

For each run we have trialed a set of 5 instances of 2D uniform distributions of 500 EDs around a GW. Each ED generates 1000 frames. Consequently, the results presented here sum up 2.5 millions of frame generations for every 160 runs of simulation.

The parameters considered for the density, the traffic load, the physical layer, the channel access protocols and the energetic performance are summed up in Table I.

The propagation model we adopted slightly advantages the GW (see instance in Fig 4). A GW receives around half the signals at a distance of 7.8 km whereas an ED hardly detects any signal after 6.5 km.

We have developed an improved and updated version of LoRaSim [15] that implements CAD+Backoff, CANL and CANL_RTS. In particular, the receive mode in presence of multiple competitors is thoroughly set alongside a full device-to-device attenuation model. We wrapped the simulator within reproducible scenarios, monitoring of a large set of metrics, and automating graph production.

B. Simulation results

In our curves each dot represents the parameters' mean and distribution for the set of instances of 2D topology[1]. We indicate with green zones the *reference scenario* (RS) which uses the set of parameters mentioned in Table I, that are common for Fig. 5, 6 and 7.

1) Response to an increase of traffic density: Fig. 5 shows that CANL listening method is more energy efficient than classical ones. For RS, an ED using CANL uses in average 610 mJ per successful 60 B frame, *CAD+Backoff 18% more,* for an average generation rate of 1 frame every 6.4 s network-wise. We confirm that over a disk with radius 2.5 km, the CAD is inefficient so CAD+Backoff remains close to ALOHA. On higher load, the cost of a successful transmission increases for CAD+Backoff faster than for CANL (CAD+Backoff 32% more than CANL for twice the traffic). On lower loads, as expected, both approaches converge.

2) Response to an increase of distances: Reliability of both CAD and frame reception decreases when distances increase. Even the ideal scheduler is impacted in terms of PDR, since frames start to get lost at 4 km (Fig. 6). For CANL, the PDR is impacted at a smaller distance (85% at 2 km, 82% at 2.5 km) because EDs could be at the opposite side of the GW (twice the distance), and are assumed less sensible (-133.25 dBm) than the GW (-138 dBm). Still, the gain of ideal over CANL and CAD+Backoff is respectively 18% and 45%.

For a radius of less than 2 km, the CAD mechanism becomes relevant and CAD+Backoff diverges from ALOHA. At 50 m, it outperforms CANL with a PDR of 93.5%. This outcome at very short range often hides CAD's poor performance at long range for LoRa.

3) Response to an increase of payload size: Collision avoidance goes with additional waiting delays. CANL increases CAD+Backoff's latency of a successful transmission by about 9. s, 11.7 s instead of 2.8 for RS, see Fig. 7. With an inter-frame time of 3200s, we believe this is negligible (0.28%). The increase is linear with the payload size, because as the EDs do not hear all the traffic on air, collisions occur and congestion is "naturally" avoided. On the contrary, the ideal scheduler's FIFO queue increases geometrically.

4) Overhead of CANL_RTS alternative: Adding an RTS frame increases overhead which impact remains stable upon the considered variations: less than +35% (+176 mJ at RS, +29%, 10% more than CAD+Backoff) in Fig. 5, less than 4.8% less in Fig. 6, less than +3.2 s in Fig. 7. Despite the impairment of the energy efficiency compared with CAD+Backoff, the gains in PDR and a reasonable latency are maintained and validate that the alternative approach makes sense.

[1] see detailed results on github.com/Guillaumegaillard/CANL-LoRa/

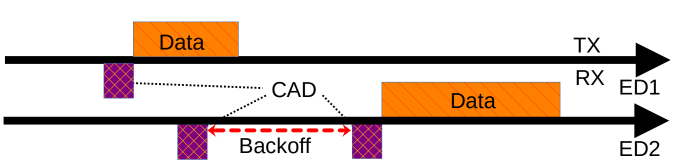

Fig. 3. A simple Carrier Sense strategy using LoRa CAD and random binary exponential backoff procedure.

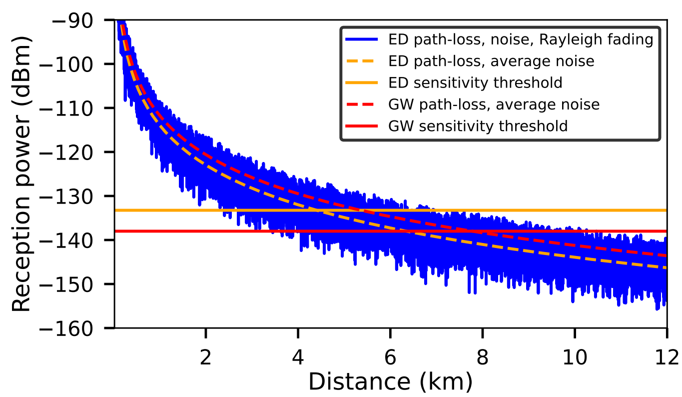

Fig. 4. In our scenario, the reception power is impaired by path loss, Gaussian noise, and Rayleigh fading. GW and EDs have different sensitivities and propagation characteristics, thus different performance. For clarity, we do not display the full instance at the GW, but only the path loss with average noise.

TABLE I
REFERENCE SETTINGS IN SIMULATION

Setting Category:	Description:		
Reference traffic	exponential distribution, no buffering	mean inter frame generation time:	3200 s each ED, 6.4 s network-wise
Reference payload sizes	normal distribution, clipped into [0,max]	mean 60 B, std. dev. 10 B, max. 150 B	in short: (60 B, 10 B, [0 B, 150 B])
Reference ED distribution	uniform 2D locations, spread on disk around GW	maximum radius: 2500 m	minimum inter ED distance: 40 cm
LoRa PHY parameters	SF12, BW125, CR4/5, frequency 868 MHz	data with explicit header	preamble duration: 12.25 symbols
Times on air	symbol: 32.8 ms; preamble: 401 ms	RTS: 827 ms; data (10 B): 991 ms	data 60 B: 2630 ms; 150 B: 5579 ms
ED log-distance model	Tx: 14 dBm; PLE: 3; Lpl-d0: 83 at 40 m; Gains: 0 dB	normal noise: (3 dB, 3 dB, [0 dB, 6 dB])	unbiased Rayleigh fading average: 4 dB
GW log-distance model	PLE: 2.95; Lpl-d0: 83 at 40 m; Gains: 1.5 dB	normal noise: (3 dB, 3 dB, [0 dB, 6 dB])	unbiased Rayleigh fading average: 4 dB
Receiver sensitivity	ED: -133.25 dBm, GW: -138 dBm	preamble detection:	minimum 3 symbols to detect a preamble
Capture power threshold	linear w.r.t. the h channel competitors	margin over other frames:	$\mathcal{M}_h = 6 + 2 \times (h - 2)$ dB
ED energy consumption	voltage supply: 3.3 V	TX, RX current: 45 mA, 5.3 mA	power in CAD: 169.54 nAh [12], [19]
CANL listen parameters	Listen window: uniform size in [4,20] preambles	fair reduction factor $\mathcal{F}$, retries:	4 preambles/retry, maximum 5 times
CANL_RTS parameters	fixed length: 5 B (4 header, 1 data length)	with implicit LoRa PHY header	RTS only for data payloads $\geq$ 12 B
CAD parameters	CAD sampling duration:	4 active symbols (131 ms)	backoff and retry at most 5 times
CAD reliability model	uniform success probability w.r.t distance	0 m: 100%; linear, down to 300 m: 95%	then log. decrease, 400 m: 20%, 420 m: 0%
Backoff parameters	uniform; minimum duration: 1 preamble	initial value 2^3 preambles	maximum exponent: 6, i.e. 2^6 preambles

V. RELATED WORK

To the best of our knowledge, CANL is the first approach using the reception mode of LoRa devices to listen neighboring transmissions before talking.

For many authors, the *Listen-Before-Talk* paradigm is not even considered an efficient enough solution to the growth in density and the issue of collisions in LoRa [3], [4], [16], [20]. In recent works [2], [21]–[23], well-known CS approaches have been studied and adapted for LoRa networks. They leverage Semtech's CAD mechanism [12] without considering the cases where the CAD fails to detect an ongoing transmission.

But Semtech's CAD has reliability issues [13], [14], [24]. Experiments showed that CAD gets unreliable at distances less than 400 m in dense urban environments. Although clearly explaining the limitations of the CAD, Kouvelas et al. does not discard its use in *np-CECADA* [13]. The authors considered CAD ranges measured in good line-of-sight condition therefore got an optimistic performance evaluation. Similarly, O'Kennedy's extensive outdoor campaign [24] obtained 4 km for CAD, but one of the device was set on top of a building, at 22 m high, thus being similar to a GW. This setting greatly differs from our ED-to-ED case.

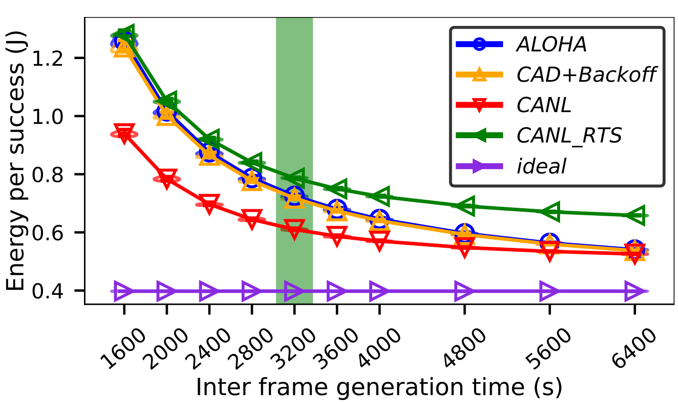

Fig. 5. Traffic is exponential, with a mean varying horizontally. E.g. for RS, each ED generates a frame every 3200 s in average. The vertical axis represents the ratio between the energy used by an ED and the amount of frames successfully uploaded to the GW.

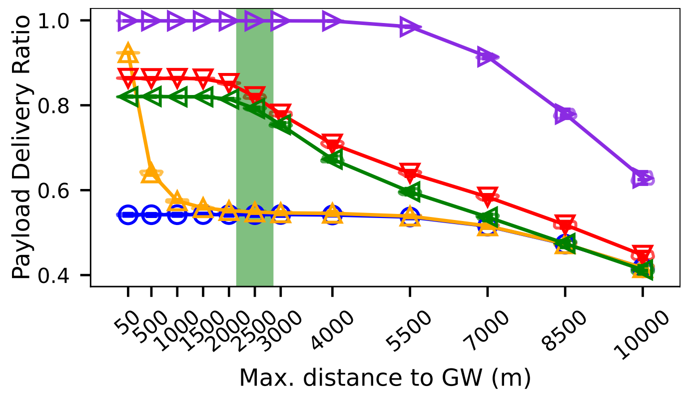

Fig. 6. The Payload Delivery Ratio (PDR) is the ratio between payload bytes in frames fully received at the GW and the payload bytes generated at the EDs. Horizontally, the five same 2D distributions of EDs are trialed while increasing their scale. E.g. for RS, the 500 EDs spread over a disk with radius 2.5 km around the GW.

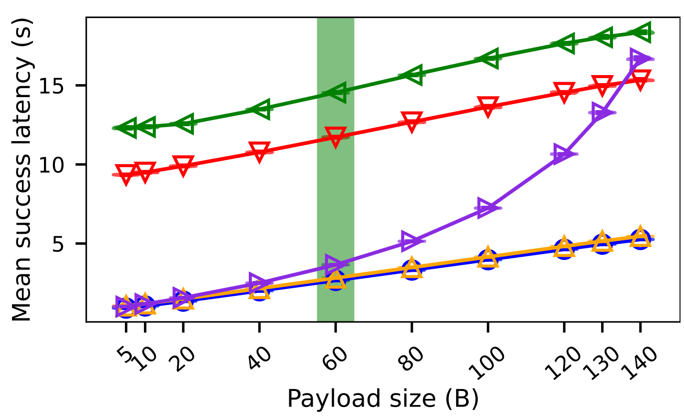

Fig. 7. The frame success latency is the time difference between the generated time and the end of its reception at GW, averaged on all successful transmissions. The payload is normally distributed, with a mean varying horizontally, a standard deviation of 10 B, and clipped into interval 0, 150 B. E.g. for RS, 68% of data frames have their payload in 50 B, 70 B.

Besides ADR [6], alternative approaches have been investigated in [25]–[27] to better assign transmission parameters (SF and TX power for instance) to devices. These approaches reduce the network density by dividing it into several contention domains. However, when a domain get saturated the access to medium will still remain difficult.

Slotted ALOHA [22], [28] or TDMA-like scheduling [20], [29], [30] can reduce the risk of collisions. Unfortunately, these solutions are hard to scale and can only be conceived in low-density scenarios as they definitely need a high level of synchronization that is very costly with duty-cycle limitations.

Another promising technique consists in exploiting the fact that CE allows a packet to be successfully decoded even under the presence of interfering devices. Early works study and leverage CE and transmission power to increase the DER [13], [31]–[33]. However, if CE can present some benefit in small-scale deployment scenarios, when the traffic density starts to increase its benefit can be close to null as shown in [14].

Besides CE, simultaneous receptions on a given logical channel can be successfully separated and received by a gateway [28], [34], [35], using various signal processing mechanisms on the particular chirp-based LoRa modulation. As far as we know, the feasibility remains to be studied on large-scale deployments, especially in presence of multiple SF that may interfere [31].

VI. Conclusions and perspective

We presented CANL LoRa, an open-loop collision avoidance mechanism implementing a Listen-Before-Talk strategy by setting EDs in LoRa packet reception mode. We showed that CANL outperforms classical CS approaches in dense LoRa networks and the extensive simulations showed that CANL is adapted to a wide diversity of parameters. Taken into account current LoRa chip hardware limitation, CANL_RTS can still provide high performance level in a real implementation. CANL and CANL_RTS will be further evaluated in real large scale implementation using our LoRa IoT platform [36].

References

[1] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, pp. 1–7, 3 2019.

[2] S. Ahsan, S. A. Hassan, A. Adeel, and H. K. Qureshi, "Improving channel utilization of LoRaWAN by using novel channel access mechanism," *IWCMC*, pp. 1656–1661, 6 2019.

[3] 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, pp. 34–40, 2017.

[4] M. Jouhari, E. M. Amhoud, N. Saeed, and M.-S. Alouini, "A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges," *IEEE Communications Surveys & Tutorials*, 2023.

[5] "TS001-1.0.4 LoRaWAN® L2 1.0.4 Specification." [Online]. Available: https://resources.lora-alliance.org/technical-specifications/ts001-1-0-4-lorawan-l2-1-0-4-specification

[6] S. Li, U. Raza, and A. Khan, "How Agile is the Adaptive Data Rate Mechanism of LoRaWAN?" *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 2018.

[7] O. Georgiou and U. Raza, "Low Power Wide Area Network Analysis: Can LoRa Scale?" *IEEE Wireless Communications Letters*, vol. 6, pp. 162–165, 4 2017.

Acknowledgments. This work is funded by the EU PRIMA INTEL-IRRIS project, which has received funding from the ANR and PRIMA S2 2020 research program under project ID 1560.

[8] M. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa low-power wide-area networks scale?" *MSWiM 2016*, pp. 59–67.

[9] M. El-Aasser, R. Badawi, M. Ashour, and T. Elshabrawy, "Examining carrier sense multiple access to enhance LoRa IoT network performance for smart city applications," *ICCE*, 9 2019.

[10] L. H. Shen, C. H. Wu, W. C. Su, and K. T. Feng, "Analysis and Implementation for Traffic-Aware Channel Assignment and Contention Scheme in LoRa-Based IoT Networks," *IEEE IoT Journal*, 2021.

[11] J. Ortín, M. Cesana, and A. Redondi, "Augmenting LoRaWAN performance with listen before talk," *IEEE TWC*, vol. 18, 2019.

[12] Semtech, "Data Sheet: SX1261/2, Long Range, Low Power, sub-GHz RF Transceiver, Rev. 2.1," 12 2021.

[13] N. Kouvelas, R. V. Prasad, N. Yazdani, and D. E. Lucani, "Np-CECADA: Enhancing Ubiquitous Connectivity of LoRa Networks," *MASS 2021*, pp. 374–382, 2021.

[14] C. Pham and M. Ehsan, "Dense Deployment of LoRa Networks: Expectations and Limits of CAD and Capture Effect for Radio Channel Access," *Sensors*, vol. 21, p. 825, 2021.

[15] T. Voigt and M. Bor, "LoRaSim," 2016. [Online]. Available: https://mcbor.github.io/lorasim/

[16] A. M. Al-Samman and et al., "A Survey on LoRaWAN Technology: Recent Trends, Opportunities, Simulation Tools and Future Directions," *Electronics*, vol. 11, p. 164, 1 2022.

[17] G. S. Sidhu, R. F. Andrews, and A. B. Oppenheimer, *Inside AppleTalk ® Second Edition*. Addison-Wesley, 1990.

[18] NumPy Developers, accessed June 29, 2023. [Online]. Available: numpy.org/doc/stable/reference/random/generator.html\#distributions

[19] Semtech, "Application Note: SX126x CAD Performance Evaluation Rev.2.1-11/2019," 2019.

[20] F. Loh, N. Mehling, and T. Hoßfeld, "Towards LoRaWAN without Data Loss: Studying the Performance of Different Channel Access Approaches," *Sensors*, vol. 22, p. 691, 1 2022.

[21] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: Efficient Carrier-Sense Multiple Access for LoRa," *MobiCom*, 2020.

[22] L. Beltramelli, A. Mahmood, P. Osterberg, and M. Gidlund, "LoRa beyond ALOHA: An Investigation of Alternative Random Access Protocols," *IEEE TII*, vol. 17, pp. 3544–3554, 5 2021.

[23] J. Álamos, P. Kietzmann, T. C. Schmidt, H. Hamburg, and G. M. Wählisch, "DSME-LoRa: Seamless Long-Range Communication between Arbitrary Nodes in the Constrained IoT," *TOSN*, 3 2022.

[24] M. O'Kennedy, T. Niesler, R. Wolhuter, and N. Mitton, "Practical evaluation of carrier sensing for a LoRa wildlife monitoring network," in *IEEE/IFIP Networking*, 2020, pp. 614–618.

[25] A. Loubany, S. Lahoud, and R. E. Chall, "Adaptive algorithm for spreading factor selection in LoRaWAN networks with multiple gateways," *Computer Networks*, vol. 182, p. 107491, 12 2020.

[26] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, "Improving reliability and scalability of LoRaWANs through lightweight scheduling," *IEEE Internet of Things Journal*, vol. 5, 6 2018.

[27] M. Slabicki, G. Premsankar, and M. D. Francesco, "Adaptive configuration of lora networks for dense IoT deployments," *IEEE/IFIP NOMS*, pp. 1–9, 7 2018.

[28] N. E. Rachkidy, A. Guitton, and M. Kaneko, "Collision Resolution Protocol for Delay and Energy Efficient LoRa Networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, 6 2019.

[29] D. Zorbas, K. Q. Abdelfadeel, V. Cionca, D. Pesch, and B. O'Flynn, "Offline Scheduling Algorithms for Time-Slotted LoRa-based Bulk Data Transmission," *WF-IoT*, 4 2019.

[30] J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Low overhead scheduling of LoRa transmissions for improved scalability," *IEEE IoT-J*, 2019.

[31] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance," *IEEE Communications Letters*, vol. 22, 4 2018.

[32] A. Rahmadhani and F. Kuipers, "When LoRaWAN frames collide," *MobiCom*, pp. 89–97, 10 2018.

[33] J. Haxhibeqiri, F. V. den Abeele, I. Moerman, and J. Hoebeke, "LoRa Scalability: A Simulation Model Based on Interference Measurements," *Sensors*, vol. 17, p. 1193, 5 2017.

[34] Z. Wang, L. Kong, K. Xu, L. He, K. Wu, and G. Chen, "Online Concurrent Transmissions at LoRa Gateway," *IEEE INFOCOM*, 2020.

[35] X. Xia, Y. Zheng, and T. Gu, "FTrack: Parallel Decoding for LoRa Transmissions," *ACM SenSys*, p. 192–204, 2019.

[36] C. Pham, "A DIY Low-cost LoRa gateway," accessed June 29, 2023. [Online]. Available: https://github.com/CongducPham/LowCostLoRaGw