

Radio channel access challenges in LoRa low-power wide-area networks

Congduc Pham¹, Ahcène Bounceur², Laurent Clavier³, Umber Noreen²,
Muhammad Ehsan¹

¹UNIVERSITY OF PAU, LIUPPA, PAU, FRANCE ²UNIVERSITY OF BREST, LABSTICC, BREST,
FRANCE ³IMT LILLE, LILLE, FRANCE

4.1 Purpose of this chapter

Recently, low-power wide-area networks (LPWANs) play a key role in the Internet-of-things (IoT) maturation process. Under the LPWAN broad term are a variety of technologies enabling power-efficient wireless communication over very long distances. For instance, technologies based on ultranarrowband modulation (UNB)—for example Sigfox—or chirp spread spectrum (CSS) modulation—for example LoRa—have become de facto standards in the IoT ecosystem. Most of LPWAN technologies can achieve more than 20 km in line of sight (LOS) condition. In a typical long-range one-hop connectivity scenario, the gateway is the single interface to Internet servers through cellular/asymmetric digital subscriber line/Ethernet/Wi-Fi technologies depending on what is available locally. Devices typically communicate directly to one or more gateways, which removes the need for constructing and maintaining a complex multihop network. Recent deployment tests with LoRa gateways located on top of high building show more than 6 km range in urban scenarios for smart city applications [1]. A large city can easily be covered with less than 10 gateways. Indoor smart building applications are also enabled by the easy coverage of buildings several stories high. Communication to high-altitude balloons has also been realized successfully [2,3] and tests with low-orbit satellites are on the way [4]. These very versatile technologies definitely provide a better connectivity answer for battery-operated IoT devices by avoiding complex synchronization and costly relay nodes to be deployed and maintained.

Given the incredible worldwide uptake of LPWANs for a large variety of innovative IoT applications, including multimedia sensors, it is important to understand the challenges behind large-scale and dense LPWAN deployment, especially because both Sigfox and LoRa networks are currently deployed in unlicensed bands. This situation is most likely not going

to change, at least in the next few years, as working in the unlicensed band allows for a much quicker uptake of the technology.

This chapter has a particular focus on LoRa technology as it can be deployed in a private and ad-hoc manner, making experimental deployments much easier. Existing studies on LoRa scalability and radio channel access mechanisms for LoRa LPWAN will be reviewed and promising approaches will be presented in more detail. The chapter will also provide to the readers useful information on the LoRa physical layer, as well as on promising interference mitigation techniques that can be applied, such as capture effect (CE) and successive interference cancellation.

The chapter will also give a large part on experimental results. The authors have conducted a large number of LoRa LPWAN experimental studies, as well as real-world deployments of both IoT test-beds and IoT production networks in the context of three R&D projects (two EU H2020 projects—WAZIUP and WAZIHUB—and one national ANR project—PERSEPEUR) in addition to numerous academic and industrial collaborations. Models and large-scale simulations are also considered and the chapter will present the open-source CupCarbon simulation tool, which especially addresses the modeling and performance evaluation of radio physical layers. The original feature of CupCarbon over other existing open-source simulation environments is its particular focus on radio propagation modeling in urban environments by taking into account three-dimensional (3D) maps of the simulated areas.

We hope that the chapter will provide a proper balance between research and experimental results. It therefore definitely targets students, academicians, researchers, and industry professionals who need a broad understanding of the technologies, their challenges, the existing solutions along with their limitations, and feedbacks from real-world deployments, as well as insights on how larger-scale scenarios can be evaluated.

4.2 Review of LoRa physical layer

LoRa utilizes bandwidth of usually 125 kHz to broadcast a signal. Using bands that are not too narrow allows LoRa to exhibit some robustness against some characteristics of the channel such as frequency selectivity and Doppler effect, to name a few. The transmitter generates chirp signals by varying their frequency over time and keeping phase between adjacent symbols constant. Receiver can decode even a severely attenuated signal 19.5 dB below the noise level [5]. The main characteristics of LoRa modulation depend on spreading factor (*SF*), coding rate (*CR*), and bandwidth (*BW*). Spreading factor $SF = \log_2(R_c/R_s)$ is the ratio between symbol rate (R_s) and chip rate (R_c). LoRa employs six orthogonal spreading factors, 7–12. *SF* provides a tradeoff between data rate and range through higher receiver's sensibility. Along with the spreading factors, forward error correction (FEC) techniques are used in LoRa to increase the receiver's sensibility further. Code rate *CR* defines the amount of FEC in LoRa frame. LoRa offers $CR = 0, 1, 2, 3$, and 4 , where $CR = 0$ means no encoding. The choice of higher *SF* and *CR* values dramatically increase the time on air (ToA), T_{air} . Using a

higher value for BW will reduce T_{air} but lowers the receiver's sensibility. LoRa provides several bandwidth values from 7.8 to 500 kHz with 125, 250, and 500 kHz being the most used. Taking these parameters into account, the useful bit rate R_b equals:

$$R_b = \frac{4 \times SF \times BW}{(4 + CR) \times 2^{SF}} \text{ (bits/s)} \quad (4.1)$$

4.2.1 LoRa physical layer (PHY) structure

LoRa is a Semtech proprietary technology and is not fully open. This section gives the analysis on the working of PHY in LoRa, according to our understanding. Fig. 4–1 shows the block diagram of LoRa transceiver, which is briefly explained in the following paragraphs.

4.2.1.1 Encoding

First, the binary source input bits pass through an encoder. The output of encoder depends on the choice of CR value. Encoding reduces the packet error rate in the presence of short bursts of interference. LoRa uses Hamming codes for FEC. These are linear block codes and are easy to implement. LoRa uses coding rates CR of $4/5$, $2/3$, $4/7$, and $1/2$, which means if the code rate is denoted as k/n , where k represents the number of useful information bits, and encoder generates n output bits, then $(n - k)$ are the redundant bits. If we assume CR values between 1 , 2 , 3 , and 4 for coding rates $4/5$, $4/6$, $4/7$, and $4/8$, respectively, then the error detection and correction capabilities are as shown in Table 4–1.

4.2.1.2 Whitening

The output of the encoder passes through the whitening block. Whitening is an optional step in LoRa, which can be implemented by Manchester encoding to induce randomness. Here, the purpose of whitening is to make sure that there are no long chains of 0s or 1s in the payload.

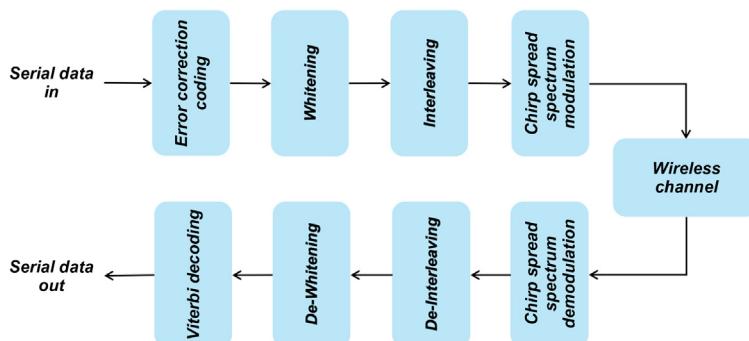
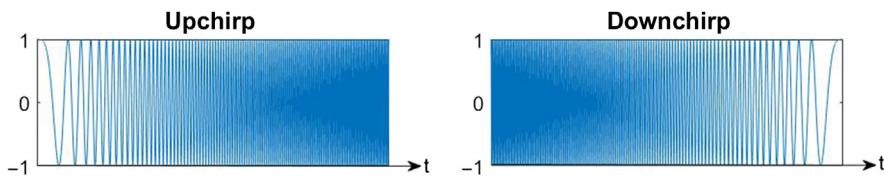


FIGURE 4–1 LoRa PHY layer architecture.

Table 4–1 Error detection and correction capabilities of LoRa.

Coding rates	Error detection (bits)	Error correction (bits)
4/5	0	0
4/6	1	0
4/7	2	1
4/8	3	1

**FIGURE 4–2** Upchirp and downchirp signal.

4.2.1.3 Interleaving

The output of whitening block is passed to the interleaving block. The interleaver uses diagonal placing method to scramble each $4 + CR$ codeword and sends it to the spreading block.

4.2.1.4 Chirp spread spectrum modulation

This block here spreads each symbol over an upchirp, according to the SF value used. For example, for $SF = 7$ and $SF = 12, 128$ and 4096 chips/symbol will be used, respectively. The relationship between the symbol rate $R_s = BW/2^{SF}$ and chip rate R_c is:

$$R_c = 2^{SF} \times R_s \quad \text{and} \quad R_c = \frac{2^{SF} \times BW}{2^{SF}}, \text{ so, } R_c = BW \text{ chips/s}$$

It takes much larger BW for transmission than required for the considered data rate. Chirp signal is a sinusoidal signal with either linearly increasing or decreasing frequency. A linear chirp waveform can be expressed as:

$$c(t) = \begin{cases} \exp(2\pi j(at + b)t), & -\frac{T_s}{2} \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

with $at + b = f_{\min} + \frac{f_{\max} - f_{\min}}{T_s} t$

where f_{\max} and f_{\min} are the maximum (125 kHz in our case) and minimum frequency, respectively. T_s is the symbol duration. An upchirp and a downchirp are shown in [Fig. 4–2](#).

Each symbol of SF bits can be represented by shifting the frequency ramp based on the symbol value. So each coded chirp is obtained by a cyclic shift in an upchirp, as illustrated in [Fig. 4–3](#). Circular shift in raw upchirp is expressed in [Eq. \(4.3\)](#).

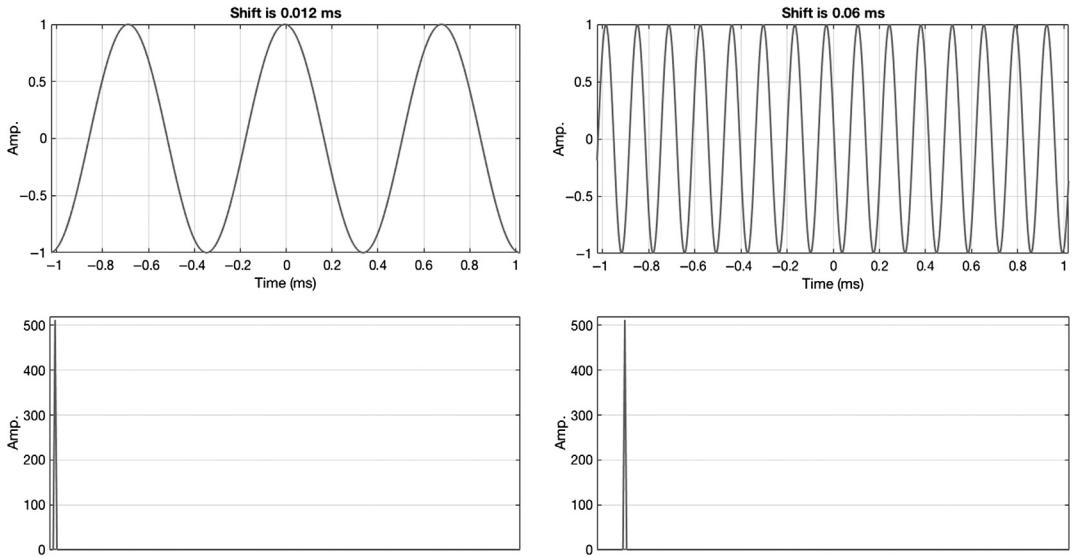


FIGURE 4-3 Matched filter output.

$$c(t) = \begin{cases} \exp(2\pi j(a(T_s + t - \Delta t) + b)(T_s + t - \Delta t)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t \\ \exp(2\pi j(a(t - \Delta t) + b)(t - \Delta t)), & \frac{-T_s}{2} + \Delta t \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \quad (4.3)$$

where Δt is the shift in time that depends on the symbol value.

4.2.1.5 Chirp spread spectrum demodulation

The matched receiver for a linear chirp is performed by multiplication with the downchirp, as shown in Fig. 4-3.

This process can be represented mathematically for $\Delta t = 0$. Note that, in case of symbol value “0,” the transmitter will send an upchirp:

$$y(t) = \exp(2\pi j(at + b)t) \times \exp(-2\pi j(a(T_s - t) + b)(T_s - t)) \quad (4.4)$$

$$y(t) = \exp(2\pi j(at^2 + bt - aT_s^2 + 2aT_s - at^2 - bT_s + bt)) \quad (4.5)$$

$$y(t) = \exp(2\pi j(2atT_s + 2bt - aT_s^2 - bT_s)) \quad (4.6)$$

$$y(t) = \exp(2\pi j(2t(at_s + b) - aT_s^2 - bT_s)) \quad (4.7)$$

details of exact ToA computation, which can be found in [5], one can say that *SF* and *BW* have direct influence on the ToA of the LoRa packet, as these parameters typically define the symbol rate: higher *SF* increases ToA, while higher *BW* decreases ToA at the cost of lower receiver's sensibility.

4.2.2 PHY performance

Fig. 4–5 shows the bit error rate (BER) performance of LoRa-based PHY, plotted over the signal-to-noise ratio (SNR). Vertical axis shows different bit error rate values and horizontal axis shows SNR values. **Fig. 4–5** shows the impact of *SF*: increasing *SF* decreases noticeably the BER at the cost of a reduced data rate and increased ToA. These results have been produced with $CR = 0$ and $BW = 125$ kHz.

Table 4–2 shows the data rates associated with the spreading factor *SF* and the coding rate *CR*. It can be easily noticed that *CR* also affects the data rate. But increasing *SF* and *CR* helps combat the harsh wireless conditions.

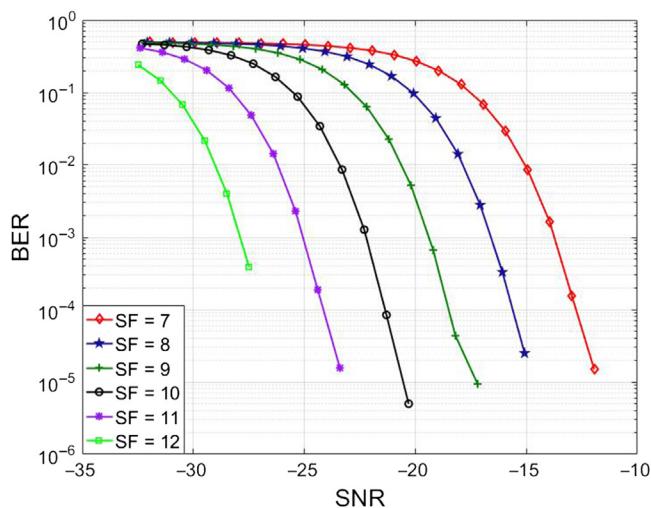


FIGURE 4–5 BER with different *SF* values and $CR = 0$.

Table 4–2 Data rates offered in LoRa.

SF	Data rate ($CR = 0$)	CR	Data rate ($SF = 7$)
7	6.8 kbps	0	6.8 kbps
8	3.9 kbps	1	5.4 kbps
9	2.1 kbps	2	4.5 kbps
10	1 kbps	3	3.9 bps
11	671 bps	4	3.4 kbps
12	366 bps	—	—

4.2.3 Interference in LoRa

A LoRa network does not define any particular channel access control, and it is therefore similar to a pure so-called ALOHA system. One of the limitations of ALOHA-based systems is its blind transmission strategy that allows the transmitters to transmit whenever there is a frame to send without carrier sensing. The vulnerable time in pure ALOHA-based network is twice the frame time, for example $2 \times T_{\text{air}}$ in LoRa. That means, a packet will be destroyed by any overlapping transmission starting in the time window that starts one packet time before the transmission of the packet and closes at the end of the transmission of the packet. Hence, the throughput of such network is: [6]

$$\eta = Ge^{-2G} \quad (4.14)$$

"2" in the superscript of exponential is because the vulnerable time is twice the frame time T_{air} . G represents average number of transmission attempts during frame time.

If we consider a large ALOHA LoRa network, transmission attempts occur according to the Poisson process with average rate G attempts per slot. If we assume equal-length packets and only nodes with same SF will collide, then G can be defined as the average number of attempts per time frame:

$$G = N \times p_i \times \lambda_i \times T_{\text{air}} \quad (4.15)$$

where T_{air_i} is the ToA of one LoRa frame (stated in Eq. 4.13), λ_i is the packet generation rate of all N end-devices using SF_i in a network, and p_i the probability that an end-device uses SF_i . The maximum throughput of pure ALOHA system can be defined by taking the derivative of Eq. (4.14) with respect to average traffic G and setting it equal to zero, $d/dG(Ge^{-2G}) = 0$, gives $G = 1/2$. Substituting this value into Eq. (4.14) gives:

$$\eta = \frac{1}{2e^{-1}} = 0.1839 \quad (4.16)$$

So, pure ALOHA-based network can give 18.39% of maximum efficiency. For LoRa modulation, this efficiency can be increased by improving the receiver technique. Fig. 4–6 shows a collision occurring between two consecutive LoRa symbols.

I_1 and I_2 are the symbols from the interference signal. R_s is the symbol of the ongoing transmission. The expression for $-T_s/2 \leq t \leq -T_s/2 + \Delta T$ can be written as:

$$S_{ii}(t) = S(t) + \begin{cases} \exp(2\pi j(a(\Delta T - t + \Delta t_1) + b)(\Delta T - t + \Delta t_1)), & (1) \\ \exp(2\pi j(a(\Delta T - t + \Delta t_1 - T_s) + b)(\Delta T - t + \Delta t_1 - T_s)), & (2) \\ 0, & \text{otherwise} \end{cases} \quad (4.17)$$

Where conditions (1) and (2) are, respectively, $-T_s/2 \leq t \leq -T_s/2 + \Delta t_1$ and $-T_s/2 + \Delta t_1 \leq t \leq T_s/2$. Then, the expression for $\frac{-T_s}{2} + \Delta T \leq t \leq T_s/2$ can be written as:

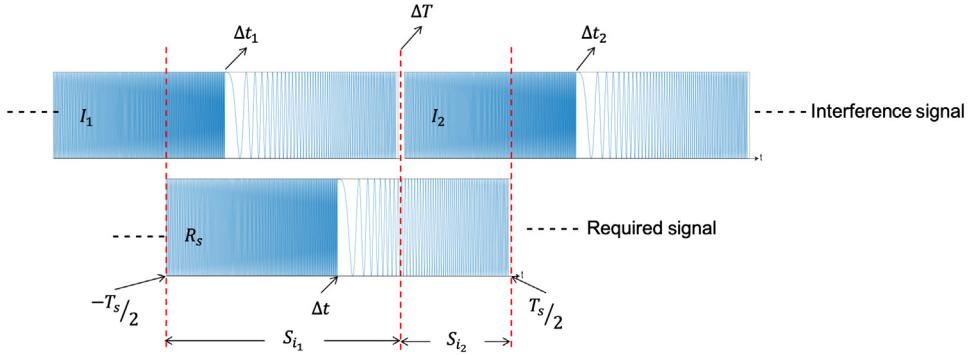


FIGURE 4–6 Interfering LoRa symbols at receiver.

$$S_{i_2}(t) = S(t) + \begin{cases} \exp(2\pi j(a(t + T_s - \Delta T + \Delta t_2) + b)(t + T_s - \Delta T + \Delta t_2)), & (3) \\ \exp(2\pi j(a(t - \Delta T + \Delta t_2) + b)(t - \Delta T + \Delta t_2)), & (4) \\ 0, & \text{otherwise} \end{cases} \quad (4.18)$$

Here, conditions (3) and (4) are, respectively, $-T_s/2 \leq t \leq -T_s/2 + \Delta t_2$ and $-T_s/2 + \Delta t_2 \leq t \leq T_s/2$. At this point, multiplication with downchirp $D(t)$ is performed, and the expression would be:

$$y_{i_1} = S_{i_1}(t) \times D(t)$$

$$y_{i_2} = S_{i_2}(t) \times D(t)$$

$$y_{i_1}(t) = y_i(t) + \begin{cases} \exp(2\pi j(2at(T_s - \Delta T - \Delta t_1) + a(\Delta T^2 + \Delta t_1^2 - T_s^2 + 2\Delta T\Delta t_1) \\ + b(\Delta T + \Delta t_1 + T_s))), & (1) \\ \exp(2\pi j(2at(2T_s - \Delta T - \Delta t_1) + a(2\Delta T\Delta t_1 - 2T_s\Delta T - 2s\Delta t_1 \\ + \Delta T^2 + \Delta t_1^2) + b(\Delta T + \Delta t_1 - 2T_s))), & (2) \\ 0, & \text{otherwise} \end{cases} \quad (4.19)$$

$$y_{i_2}(t) = y_i(t) + \begin{cases} \exp(2\pi j(2t(2aT_s - a\Delta T + a\Delta t_2 + b) + a(2T_s\Delta t_2 - 2T_s\Delta T \\ - 2\Delta T\Delta t_2 + \Delta T^2 + \Delta t_2^2) + b(\Delta t_2 - \Delta T))), & (3) \\ \exp(2\pi j(2t(aT_s - a\Delta T + a\Delta t_2 + b) + a(\Delta T^2 + \Delta t_2^2 - T_s^2 \\ - 2\Delta T\Delta t_2) + b(\Delta t_2 - \Delta T - T_s))), & (4) \\ 0, & \text{otherwise} \end{cases} \quad (4.20)$$

The frequency response of Eqs. (4.19) and (4.20) will give sharp narrow peaks at time Δt , Δt_1 and Δt_2 that correspond to the constant value of the coded chirps, as shown in Fig. 4–7. The receiver can make the decision by selecting the peak with the highest amplitude value.

The receiver can successfully differentiate between the interferers based on their received power values. The problem occurs when one or more interferers carry either the same value

$(\Delta t = \Delta t_1 = \Delta t_2)$ so that their power adds up or have an equivalent or more power than the useful one. In Fig. 4–8, first interferer has the same received power as the useful user and in Fig. 4–9, all interferers contain the same value ($\Delta t = \Delta t_1 = \Delta t_2$), which causes an increase in peak value in the frequency domain.

4.2.4 Orthogonality of LoRa transmissions

The spreading factor SF in LoRa is the ratio between symbol rate R_s and chip rate R_c : $SF = \log_2(R_c/R_s)$. LoRa employs six spreading factors from 7 to 12 that provide some sort of orthogonality in data transmission (collision-free) occurring on the same frequency. This orthogonality property can be explained as follows: different SF will give a different chirp rate, which is the change in the frequency with respect to time. As CSS uses frequency chirps with linear variation of frequency over time, when plotting frequency against time the chirp rate will be the slope of the line. Therefore, theoretically, different SF will give different slopes that provide the orthogonality property.

However, when using a LoRa combination of BW and SF , not all combinations are orthogonal because some of them define the same chirp rate (same slope) [7]. This can be further explained as follows: the slope (chirp rate) can be defined as slope $= (f_{\max} - f_{\min})/T_s$, where T_s is the symbol duration. As the symbol rate is $R_s = BW/2^{SF}$ then $T_s = 1/R_s = 2^{SF}/BW$. With $f_{\max} - f_{\min} = BW$ we have slope $= BW * (BW/2^{SF}) = BW^2/2^{SF}$.

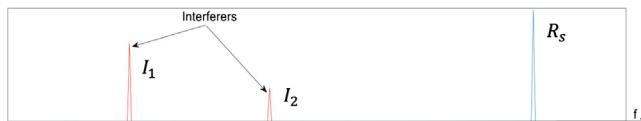


FIGURE 4–7 Coded chirps at 30, 100, and 128 (different TX power). $SF = 8$.

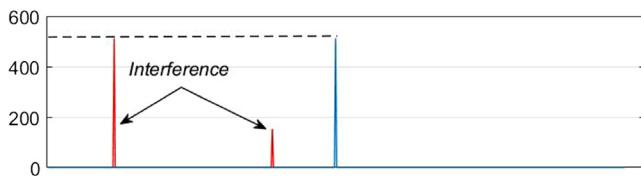


FIGURE 4–8 Coded chirps at 30, 100, and 128 (same TX power). $SF = 8$.

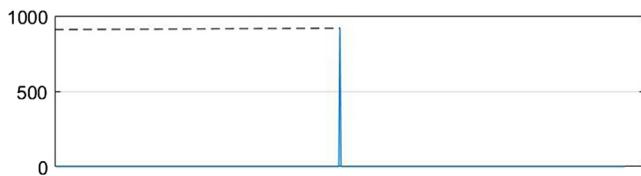


FIGURE 4–9 FFT of three coded chirps at 128. $SF = 8$.

When considering only the LoRa lower layers that mainly consist in the physical layer without any Listen Before Talk (LBT, or Carrier Sense) mechanism, a simple scalability study can use the simple ALOHA model shown in [Section 4.2.3](#). If we add SF orthogonality as explained previously, we can consider that each orthogonal BW and SF combination provides an independent ALOHA channel. Here, the issue is how to distribute or assign a given (BW, SF) combination to devices. We will discuss this issue and present the adaptive data rate (ADR) mechanism proposed by the higher-level LoRaWAN specification in [Section 4.3.1](#).

At the highest level, limitation of device's duty cycle imposed by regulation agencies, or by a user community with a so-called "fair share" approach such as the one proposed by TheThingsNetwork, can be a solution to reduce uplink traffic and thus the collision probability. This will be explained in more detail in [Section 4.3.2](#).

4.3.1 Impact of adaptive data rate

The LoRaWAN specifications [\[15\]](#) (v1.1 at the time of writing) propose an ADR mechanism for optimizing data rates, airtime, and energy consumption in the network. By optimizing, that is mainly reducing, airtime, contention on the radio medium is also reduced and thus the collision probability. Basically, LoRa gateways can use information, such as RSSI and/or SNR, from uplink messages from a given device to determine the "margin" that is still available to correctly demodulate messages. This margin is used to determine how much the device can increase its data rate, therefore reducing the airtime. Practically, airtime can be reduced by using higher value for BW and/or smaller value for SF . After determining the new parameters, a downlink command message will be sent by a gateway to a device to set the new data rate.

The agility of the ADR mechanism has been studied in Ref. [\[16\]](#). The authors reported that the convergence time can be very high because many packets need to be received by gateways. As network size increases the convergence time increases, as well due to high contention on the radio medium: "from around 200 minutes for a 100-node network to more than 3000 minutes for a 4000-node network." The convergence time also increases significantly when the link quality degrades. The explanation by the authors is as follows: devices need lots of time to move from lower to higher value of SF to regain connectivity because the process unfortunately requires devices to lose sufficient number of sent packets before moving to higher SF .

For static devices, as it is most likely the case in a large variety of IoT deployment, the ADR mechanism can indeed help to reduce the ToA of uplink transmissions. However, as many devices can converge to use the same LoRa parameters, collision probability may increase, leading to lower link quality, thus impacting convergence time as observed in [\[16\]](#). We believe there is a risk that the ADR mechanism will make gateways and devices oscillating from one parameter combination to another without taking into account that more variety in (BW, SF) combinations provides more orthogonality, making the system closer to N independent ALOHA channels. These issues certainly need more studies in the future to see how smarter parameter assignment strategies can be proposed as slightly addressed in [\[12\]](#).

There are also some interesting works on scheduling LoRa transmissions when the network size is not too large [17–19].

4.3.2 Impact of duty-cycle limitation

The flexibility of long-range transmission in the unlicensed bands comes at the cost of stricter legal regulations such as limited duty-cycling, for example maximum transmission time per hour. For instance, in Europe, electromagnetic transmissions in the unlicensed EU 863–870 MHz industrial-scientific-medical (ISM) band used by Semtech's LoRa technology falls into the ETSI's short-range devices (SRD) category. The ETSI EN300-220-1 document [20] specifies for Europe various requirements for SRD, especially those on radio activity. Basically, a transmitter is constrained to 1% duty cycle (i.e., 36 s/h) in the general case. This duty-cycle limit applies to the total transmission time, even if the transmitter can change to another channel. Note that this duty-cycle limitation approach is also adopted in China in the 779–787 MHz ISM band. US regulations in the 902–928 MHz ISM band do not specify duty cycle but rather a maximum transmission time per packet with frequency hopping constraints. LBT along with adaptive frequency agility can be used to go beyond the 1% duty-cycle limit but then additional restrictions are introduced: the ToA for a single transmission cannot exceed 1 s. If this 1 s limit is respected, then the transmitter is allowed to use a given channel for a maximum T_x on time of 100 s over a period of 1 hour for any 200 kHz bandwidth.

Therefore, due to regional restrictions on operation in licensed free ISM bands, frame generation rate λ_i usually depends on the duty cycle D and ToA. If we assume that deployed network locates in Europe, then according to ETSI 1% duty cycle applied on the usage of each subband [21].

$$\lambda_i = \frac{D}{T_{\text{air}_i}} \quad \text{or} \quad \lambda_i = \frac{1}{T_{\text{of } f_i} + T_{\text{air}_i}} \quad (4.21)$$

Higher specifications such as LoRaWAN can define $(T_{\text{of } f_i} = (T_{\text{air}_i}/D) - T_{\text{air}_i})$ to be the minimum period during which a device cannot access the medium due to the duty-cycle restriction. In this case Eq. (4.14) becomes:

$$\eta_i = p_i \lambda_i N T_{\text{air}_i} e^{(-2p_i \lambda_i N T_{\text{air}_i})}, \quad i \in SF \quad (4.22)$$

The term e^{-2G} is called probability of successful transmission P of a frame and $0 \leq P \leq 1$. For a LoRa network, it can be defined as:

$$P_i = e^{(-2p_i \lambda_i N T_{\text{air}_i})}, \quad i \in SF \quad (4.23)$$

The outage probability will be:

$$P_{\text{out}_i} = 1 - P_i, \quad i \in SF \quad (4.24)$$

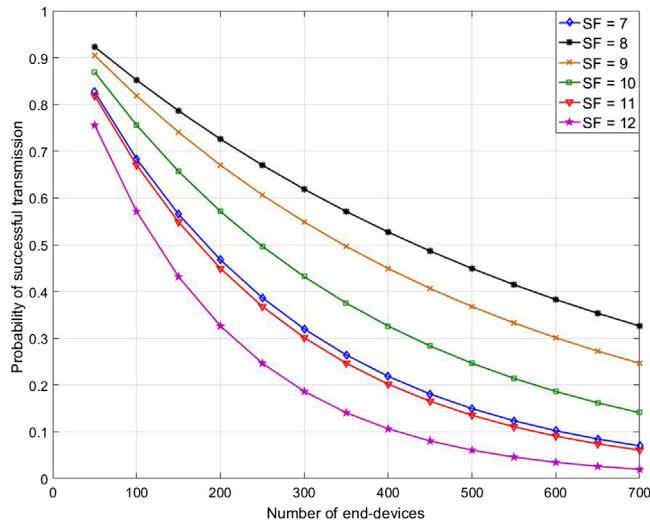


FIGURE 4–11 Probability of successful transmission.

For simplicity, we are going to consider that different spreading factors do not interfere, and we are interested in the limit in successful transmissions in a LoRaWAN system. If the probabilities p_i for an end-device to use SF_i are the same, then the definition of $T_{of,f}$ will make the load on each SF similar. In Fig. 4–11, we illustrate the probability of successful transmission depending on the number of users and SF when $p_{7-12} = [0.19, 0.08, 0.10, 0.14, 0.20, 0.28]$. With the increase in the network size, the probability of successful reception drops due to an increase in collisions in the network but is higher than a traditional ALOHA system thanks to the duty-cycle limitations. Our analytic results shown in Fig. 4–11 are consistent with simulation results presented in Ref. [13].

4.3.3 Interference mitigation: capture effect

LoRa PHY is a kind of frequency modulation that manifests capture effect (CE). In the past, many theoretical studies on CE have been performed to increase the packet reception rate (PRR) of a network, in the presence of a collision. But in the context of LoRa, not much research has been done. Practical studies in Refs. [9,13,14,22] have shown CE for LoRa-based system. In Ref. [23], authors presented capture study on equal power collisions in the pure ALOHA-based 802.15.4 system. In Ref. [24], authors state that their collision detection approach can differentiate between a packet collision and packet loss for 802.15.4-based system.

In general, the receiver keeps monitoring for the new potential preambles and if its signal-to-interference noise ratio (SINR) is above a given ratio, receiver stops ongoing reception and resynchronizes with the new packet and demodulates the signal. We are going to characterize CE in the next section (the probability for a packet to be decoded despite the presence of one interferer). Note that for the sake of simplicity, we only consider two-packet

collision scenario. This analysis can be extended to three or more packet collisions. The capture characteristics of any radio transceiver depend on the modulation, decoding schemes, and its hardware design and implementation. In a radio-frequency interference environment, a particular signal X can be successfully decoded if:

$$\text{SINR}_X = \frac{P_X}{\sum P_I + \sigma^2} > Th \quad (4.25)$$

where P_X is the source signal strength, $\sum P_I$ is the aggregate interference strength from the other active users in the network, σ is the channel noise coefficient, and Th is the minimum SINR threshold required to successfully decode signal X . When two or more packets collide, with CE, it is still possible to receive one of them. CE enables the receiver to decode a packet that satisfies Eq. (4.25), even if it arrives during the reception of an ongoing packet.

For a LoRa modulation, as shown in Figs. 4–8 and 4–9, only the strength of the strongest interferer will matter as long as the simultaneous number of interferers is not too high and the probability to have interferers with a shift that falls at the same time remains low. So Eq. (4.25) will become:

$$\text{SIR}_X = \frac{P_X}{P_I} > Th \quad (4.26)$$

In the literature [14,22–25], usually, two capture scenarios are taken into account. Both capture scenarios are shown in Fig. 4–12.

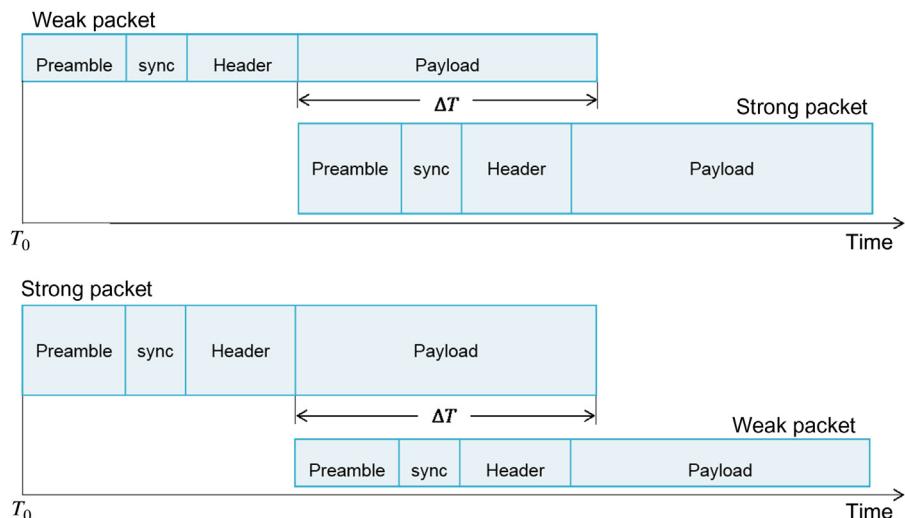


FIGURE 4–12 Capture Scenarios: (top): Stronger First and (bottom): Stronger Last.

- Decoding the First: During the reception of a packet, a second packet arrives and creates collision. In this case, receiver synchronizes with the first packet and tries to perform successful reception.
- Decoding the Last: Another scenario would be to decode the packet that arrives later. This necessitates to be able to detect the preamble of the second packet and then to correctly decode the packet.

It is worth mentioning that a recent work proposes a mechanism able to decode both transmissions either when there is a small time difference between the two signals or when they are well synchronized [26].

4.3.3.1 Capture effect simulations

We first conducted simulations and generated random collisions at the receiver by generating two LoRa packets with time difference ($T_0 \leq \Delta T \leq T_{\text{air}}$). The first signal arrives at T_0 and the second signal arrives after a random duration, within the T_{air} of the first packet. The transmission of an interfering packet can start at any time, and overlapping length ΔT of both packets varies randomly. The goal here is to identify under which power settings the collision detection and successful reception will work. In both cases, PRR is measured at the receiver, in simple steps as follows:

- Preamble detection: if preamble detection is valid, then it passes for sync word detection.
- Sync word detection: after the receiver detects the preamble it searches for the sync word and finds the starting of the header.
- Validation of header and payload: if header and payload data are not corrupted, then it is considered as successful frame reception.

The packet structure used in these simulations is shown in Fig. 4–4. The preamble consists of four upchirps, two downchirps, and a quarter of an upchirp. However, increase in preamble duration can improve the detection probability. An explicit header is used with a 2-byte CRC. The header is encoded with $CR = 4$. The payload is 20 bytes long, with no channel encoding $CR = 0$ and $SF = 8$. Channel coding is used to improve the reliability of the communication system by adding redundancy in the transmit data.

Fig. 4–13 presents the capture results. The probability of successful reception is calculated with 1000 packets transmissions for each power setting on random overlapping lengths ΔT . The x -axis shows signal-to-interference (SIR) power and y -axis shows the probability of successful reception with capture. Note that we do not expect that the received power ordering is known a priori. From Fig. 4–13, we can assume that if the received power difference between two interferers is around 1 dB, the receiver can successfully decode the strong packet. Thus, Eq. (4.26) can be expressed as:

$$\text{SIR}_X = \frac{P_X}{P_I} \geq 1 \text{ dB} \quad (4.27)$$

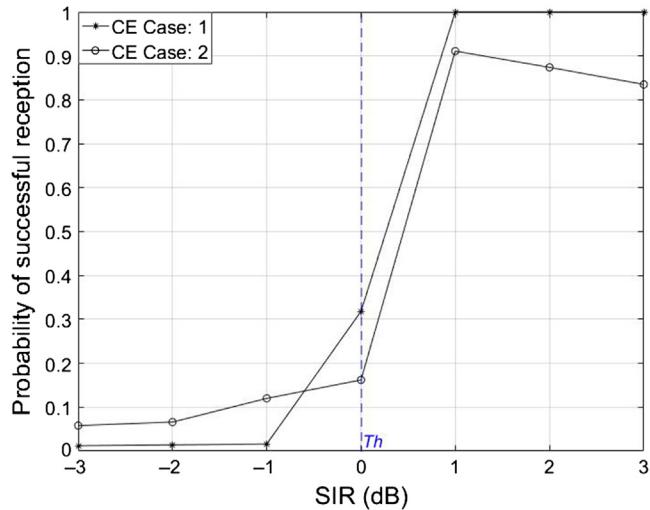


FIGURE 4–13 Capture results with $SF = 8$, $CR = 0$, and $BW = 125$ kHz.

Successfully decoding one of the colliding packet can significantly increase the system throughput of any network. In the presence of CE, the total throughput of the system of N nodes can be expressed as:

$$\eta_{CE} = G \times [P(\text{no collision}) + P(\text{collision}) \sum_{i=1}^G P(\text{SIR}_i > Th)] \quad (4.28)$$

$P(\text{no collision})$ and $P(\text{collision})$ are probability of no collision and probability of collision at the receiver, respectively. Th is a threshold value set on signal to interferer i power ratio SIR_i of received signal.

In a pure ALOHA network, a node can successfully transmit a frame if no other node has a frame to transmit during two consecutive frame times (vulnerable time $2T_{\text{air}}$). The probability of a node having no frame to send is $(1 - p)$. The probability that none among the rest of $N - 1$ nodes have a frame to send will be $(1 - p)^{N-1}$. The probability that none of the $N - 1$ nodes have a frame to send during the vulnerable time is $(1 - p)^{2(N-1)}$. Then the probability of being alone of a particular node will be $P = p(1 - p)^{2(N-1)}$.

4.3.3.2 Capture effect experimentations

In Refs. [13,14], the authors experimentally proved the possibility of successful reception of concurrent transmissions using LoRa's modulation. They concluded that there are two important things to keep an eye on. First, the start time of the collision and second is the interfering signal strength. The authors concluded that when the received signal strength indication (RSSI) from the interfering signal is the same or lower than the signal being interfered, and that if the interfering transmission starts after the preamble of the transmission being interfered, then the interfered transmission will be received correctly. They found that

to synchronize with a transmitting node, the receiver only needs six symbols of the preamble to be received without collision.

More detailed experimentations with very accurate timing have been performed in Ref. [13]. The authors also tested the case when the RSSI of the interfering transmission is higher at the receiver than the interfered transmission. They found that if the interfering transmission or interfering signal starts after the end of the preamble and header time, the transmission being interfered will be received with wrong payload CRC. However, in case the last six symbols of the transmitter's preamble can be received correctly, the receiver can synchronize with the transmitter and the reception can be successful. It is also important to note that the authors only used the 125-kHz channel bandwidth, and they think that additional experiments are required for other bandwidth channels.

In order to get accurate timing, authors in Ref. [13] experimented with devices placed close together and all connected to a timing unit. We performed additional experiments to get results in a real setting with more LoRa transmission parameters.

4.3.3.2.1 Capture effect setting

Our experimentation setting consists of two transmitters (one master and one slave node) and two receivers (gateways) as depicted in Fig. 4–14: the master node is at around 25 m from the gateway and the slave node was placed at a distance of around 150 m from the same gateway. Tests are performed outdoor in LOS conditions.

To synchronize the transmitter nodes, the master node continuously sends a message to the slave node. On receiving a message from the master, the slave acknowledges the reception of the message by sending an ACK. The slave node synchronizes with the master's clock by taking the message reception time and subtracting the ToA of the message from it. The master node on receiving the ACK performs the same action: it takes the ToA of the ACK and removes it from the time of reception of the ACK, hence synchronizing with the slave clock. Once the nodes are synchronized, they start broadcasting a message every 25,000 ms. We switch on an LED at the beginning of a transmission to visually check that the nodes have successfully synchronized.

Once the nodes are synchronized, to analyze the CE, the nodes start transmitting at the same time. Then, every 10 messages, we add a predefined delay at the slave node (the delay is approximately 1/8th of the ToA of the transmitted message): the first 10 messages (round 0) are sent at the same time by both nodes, the next 10 messages (round 1) are sent by the slave with a delay and so on. If t_{master} is the transmission time at the master, the slave will send its message at $t_{\text{slave}} = t_{\text{master}} + r * \text{delay}$, where r is the round number. The delay is introduced after every 10 messages until there is no transmission overlap.

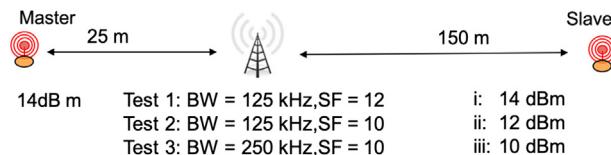


FIGURE 4–14 Experimentation setting.

We tested three different LoRa settings by varying bandwidth BW and the spreading factor SF . We also performed the tests with different maximum power settings for each transmitting node. We also performed the tests with different maximum power settings for each transmitting node: first with both the transmitters having the same maximum output power of 14 dBm, then we reduce the maximum output power of the slave node to 12 dBm for the second test and 10 dBm for the third test. Finally, we did a final test with both the transmitters having maximum output power of 10 dBm. Code rate CR is kept same for all the experiments. Both transmitters have the same payload, which is 240 bytes, which will remain constant throughout the experiment. The master and slave nodes are Arduino Nano boards each equipped with a LoRa inAir9 radio module. All the communication took place at 868 MHz frequency band. Two gateways are used: one is an Arduino Nano with the same inAir9 radio module and one is a LoRaWAN RAK831 gateway with an SX1301 radio concentrator running a simple util_pkt_logger program.

4.3.3.2.2 Results Test 1

For the first group of tests, we have BW set to 125 kHz and SF to 12. For a payload of 240 bytes, the ToA is 8870 ms. The slave uses a delay increment of 1000 ms every 10 messages. Fig. 4–15 shows the results when we put the slave node at 14 dBm (same as master), then at

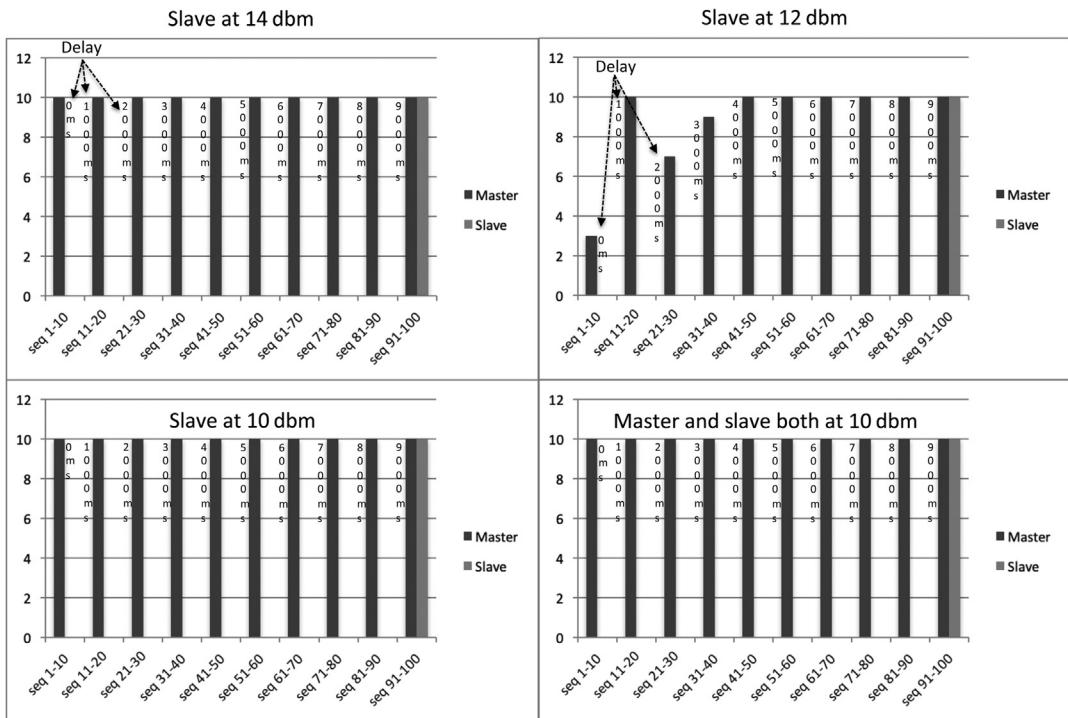


FIGURE 4–15 LoRa test 1.

12 and 10 dBm. In these results, we clearly see that we were able to receive most of the messages from the master, hence proving the CE. When the slave reaches a cumulative delay of 9000 ms, there is no overlap anymore and gateways receive from both transmitters.

4.3.3.2.3 Results Test 2

For the second group of tests, SF is now 10, while BW remains at 125 kHz. ToA is now 2206 ms and the slave uses a delay increment of 300 ms. Results are summarized in Fig. 4–16 and again confirm the CE as 89% of the messages were received.

4.3.3.2.4 Results Test 3

For the third group of tests, BW is now set to 250 kHz and SF remains at 10. Toa is 1100 ms and the slave uses a delay increment of 300 ms. The results were found very similar to Test 1, with 94.5% of the messages received with no error.

4.3.3.2.5 Test in indoor conditions

We also performed the three previously described scenarios in indoor conditions. While the main results remain the same confirming the CE for the strongest and first transmitted messages, there is more instability in the results and no packets can be received in many cases.

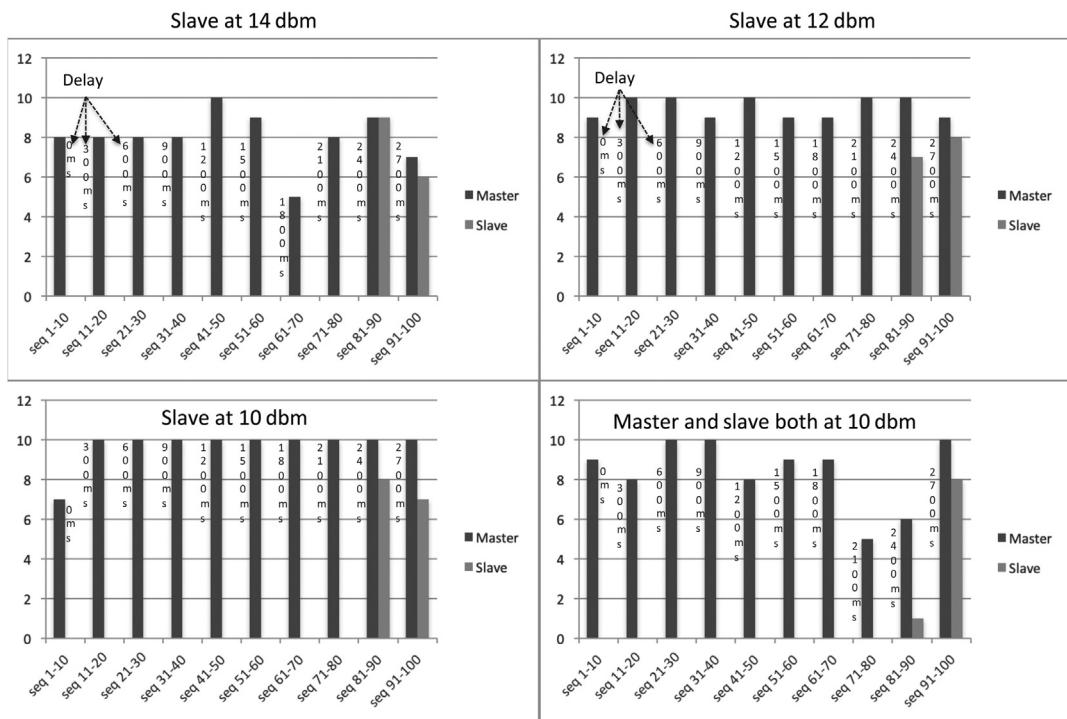


FIGURE 4–16 LoRa test 2.

4.3.4 Interference mitigation: interference cancellation

As we have described earlier, in LoRa, collision occurs due to the simultaneous arrival of two or more packets with the same SF at the receiver. According to CE, the strongest packet can be received successfully and the other packet will be considered as interference. However, successive interference cancellation facilitates the recovery of weaker packet too, as well. First, stronger signal is decoded normally; decoded signal is then subtracted from the combined signal. Then, weaker signal is extracted from the residue [27]. In case of multiple interferences, this can lead to an iterative process. The strongest signal is detected first from the received signal and then the next strongest and so on. After each signal's decoding, the received signal for that user can be reconstructed by recreating the transmit signal and applying an estimate of the channel to it. This can be subtracted from the composite received signal, which then allows subsequent users to experience a cleaner signal. A block diagram of CE and successive interference cancellation (SIC) is shown in Fig. 4–17.

There is very little work on SIC for LoRa to the best of our knowledge. In Ref. [28], the authors use simulation to optimize gateway placement when interference cancellation can be realized. In Ref. [29], the authors investigated SIC for UNB networks due to its specific interference behavior. Here, we try to provide some simple analytic results to show SIC potential benefits when considering SIC as pure receiver technique, which means it does not require any type of modification on the transmitter.

Let us assume a LoRa network consisting of a gateway node (receiver) and N transmitters scattered around the gateway node in Poisson field $\Phi = \{(L_i, H_i)\} \subset \mathbb{R}^d \times \mathbb{R}^+$. Where L_i represents the location of each transmitting node and H_i is the channel attenuation coefficient. The SIR-

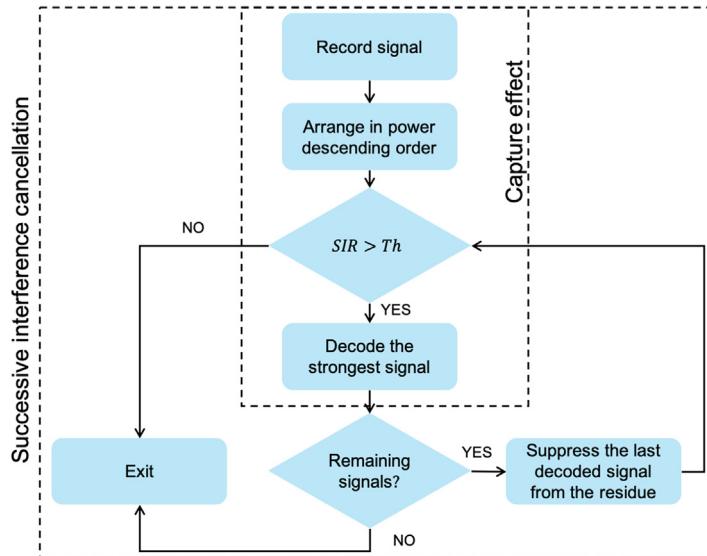


FIGURE 4–17 Block diagram for CE and SIC.

based successful decoding of single user is defined in Eq. (4.26). Which says that a particular signal X at $L \in \Phi$ can be decoded successfully if its SIR is greater than some threshold. Any signal x_r can be decoded successfully from the residue of received signal $Y(t) = \sum_{\text{active users}} X_i(t)H_i(t)$, if its signal to residual interference plus noise power ratio $S I_r R_x$ is:

$$S I_r R_x = \prod_{i=1}^n \frac{P_i}{P_{i+1}} \geq Th \quad (4.29)$$

Figs. 4–18 and 4–19 show the probability of successful decoding of the packets and throughput of LoRa-based system with collisions, consecutively. We have varied the network size N . The assumption is that all the nodes are using the same SF, and their packet generation rate is given in Eq. (4.21). It can be observed that CE and SIC can significantly improve

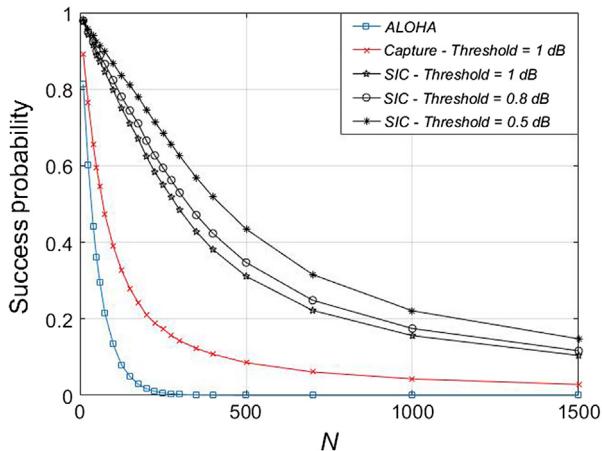


FIGURE 4–18 Probability of successful transmission.

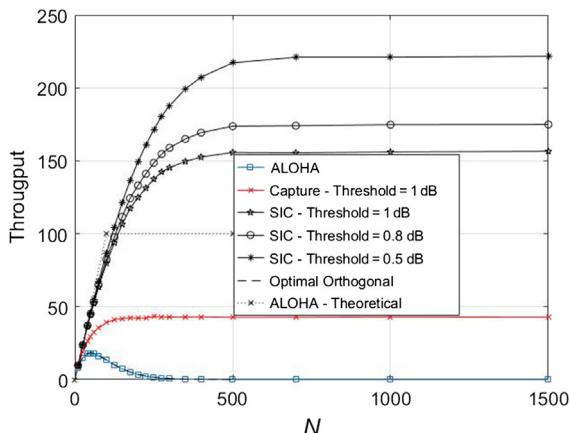


FIGURE 4–19 Throughput.

network performance. This case can be considered a worst-case scenario because no channel coding is used which would significantly improve the performance.

4.4 Channel access: sharing the bandwidth

Sharing the wireless channel is a joined task of the PHY and media access control (MAC) layer. While sharing at PHY layer is usually done with advanced signal coding/modulation schemes, sharing at MAC layer usually assumes that simultaneous transmissions at PHY layer will result in interferences leading to so-called packet collisions. Therefore, MAC layer sharing mechanisms usually implement either a time-division multiple access (TDMA) method or a competition-based approach where nodes compete to get the channel. We focus here in the latter category as TDMA methods need a high synchronization level (note that Zorbas et al. [17] did propose a TDMA-like scheduling mechanism for LoRa). In the competition-based category, random access protocols such as the early ALOHA and various variants of carrier-sense multiple access (CSMA) are widely used in wireless networks because of their relative simplicity and their distributed operation mode that does not require coordination nor overwhelming signaling overheads. There has been a notable amount of research done on the performance of ALOHA and CSMA in wireless networks. It is beyond the scope of this paper to go through all these contributions but interested readers can refer to Refs. [30–32] as a starting point.

4.4.1 Review of media access control mechanisms

4.4.1.1 IEEE 802.11

Among many CSMA variants, the one implemented in the IEEE 802.11 (Wi-Fi) is certainly one of the most used in wireless networks thanks to the worldwide success of Wi-Fi technology and is therefore quite representative of the approach taken by most of random access protocols with so-called backoff procedure. Fig. 4–20 illustrates the IEEE 802.11 CSMA

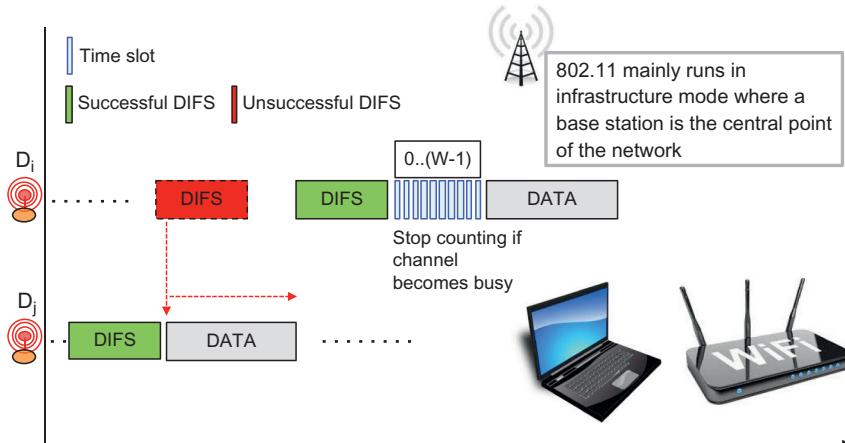


FIGURE 4–20 IEEE 802.11 DCF CSMA/CA.

mechanism used in the basic distributed coordinated function (DCF) mode, which is the common operation mode of Wi-Fi networks with a base station. In this basic mode, the optional received signal strength indication mode is not used. The basic DCF IEEE 802.11 CSMA/CA (collision avoidance) works as follows:

- Collision detection is not used since a node is unable to detect the channel and transmit data simultaneously, thus CA variant.
- A node senses the channel to determine whether another node is transmitting before initiating a transmission.
- If the medium is sensed to be free for a DCF interframe space (DIFS) time interval, the transmission will proceed (green DIFS).
- If the medium is busy (red DIFS), the node defers its transmission until the end of the current transmission and then it will wait for an additional DIFS interval before generating a random number of backoff slot time chosen in the range $[0, W - 1]$. W is called the backoff window or contention window.
- The backoff timer is decreased as long as the medium is sensed to be idle, and frozen when a transmission is detected on the medium, and resumed when the channel is detected as idle again for more than DIFS.
- When the backoff reaches 0, the node transmits its packet.
- The initial W is set to 1. W is doubled for each retry (exponential backoff) until it reaches a maximum value.
- If the maximum number of retries is reached, report error to higher layers.

The random backoff timer is applied after a busy channel because it is exactly in that case that the probability of a collision is at its highest value. This is because several users could have been waiting for the medium to be available again.

4.4.1.2 IEEE 802.15.4

Closer to the domain of IoT, IEEE 802.15.4 was for many years the standard for low-power devices such as wireless sensor networks (WSNs). Being a short-range technology (about 100–200 m in real deployment), 802.15.4 is complemented at higher layers with multihop routing mechanisms. At the MAC layer, IEEE 802.15.4 proposes both nonbeacon-enabled mode with unslotted CSMA/CA channel access mechanism and beacon-enabled networks with slotted CSMA/CA. Here, again, we are describing the nonbeacon-enabled mode as the beacon-enabled needs a coordinator and higher level of synchronization that is definitely not suited for LoRa IoT networks. The IEEE 802.15.4 nonbeacon-enabled with unslotted CSMA/CA mode works as follows:

- Collision detection is not used since a node is unable to detect the channel and transmit data simultaneously, thus CA variant.
- Before a transmission, a node waits for a random number of backoff periods chosen in the range $[0, 2^{BE} - 1]$. backoff exponent (BE) is set to 3 initially.

- If at the end of the waiting time the medium is sensed to be free (clear channel assessment, CCA) the transmission will proceed.
- If the medium is busy, the node defers its transmission, increases BE until it reaches a maximum value, and waits for additional $[0.2^{BE} - 1]$ backoff periods.
- If the maximum number of retries is reached, report error to higher layers.

Compared to IEEE 802.11, IEEE 802.15.4 always implements a backoff timer prior to any transmission and simply increases the backoff timer interval each time the channel is found busy for the same packet, without constantly checking the channel to know when it is going back to idle. There are several reasons for these differences. One reason is that simply increasing the backoff timer interval is less energy-consuming than determining the end of the current transmission, especially if the transmission of a packet can take a long time (802.15.4 usually runs at 250 kbps while 802.11 runs at 11 Mbps and above). Another reason is that the node and traffic density of IEEE 802.15.4 networks is expected to be much smaller than those of Wi-Fi networks. There is an additional reason 802.15.4's CSMA is different from 802.11's CSMA: 802.15.4 for WSN mainly runs under mesh topology (i.e., P2P and without central coordinator) with a shorter radio range (i.e., low transmit power); therefore the spatial reuse is higher, contributing again to decrease the traffic density at any given point in the network.

Again, there has been a huge amount of research in improving the basic 802.15.4 MAC protocol to better support multihop and duty-cycled low-power WSN. For instance, and to name a few, sensor mac [33] which introduces synchronization features to have common active periods and berkeley mac [34] and X-MAC [35], both with low-power listening capabilities. Readers can refer to Ref. [36] for a survey of MAC protocols for WSN (Fig. 4–21).

4.4.2 Clear channel assessment in LoRa

Before investigating what CSMA approach can be adapted for LoRa, it is necessary to know how a LoRa channel can be defined busy or idle to implement a carrier-sense mechanism.

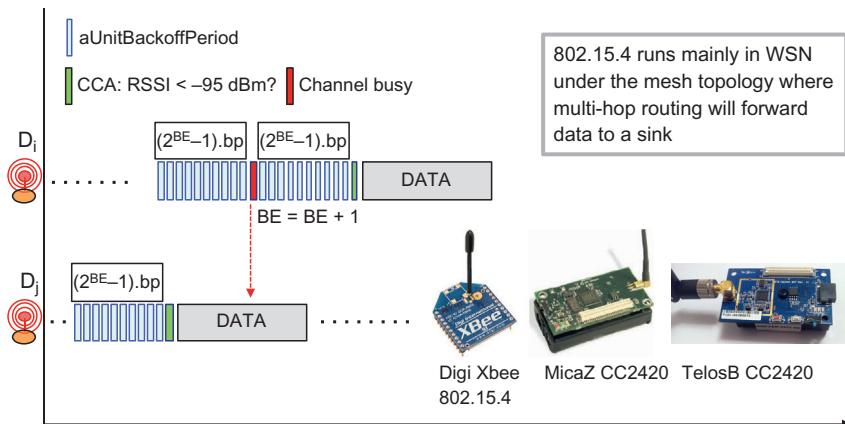


FIGURE 4–21 IEEE 802.15.4 nonbeacon unslotted CSMA.

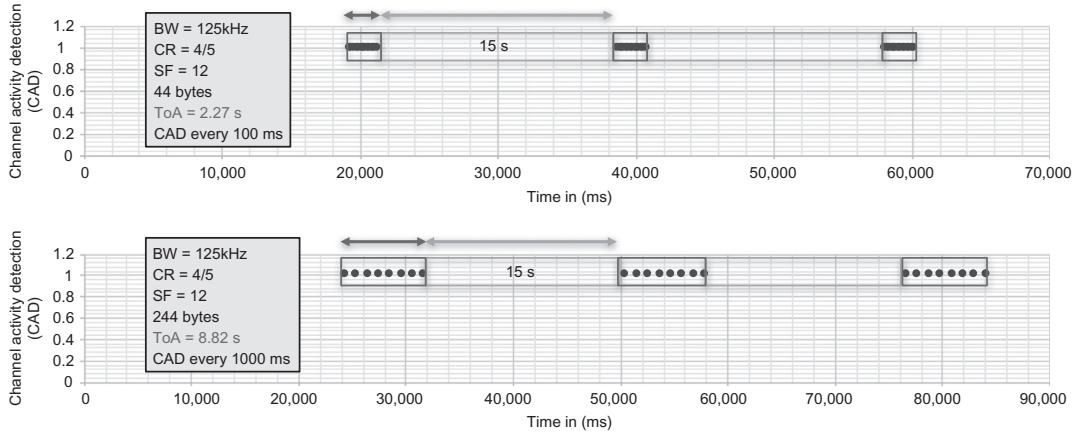


FIGURE 4–22 Test of the LoRa CAD mechanism.

As LoRa reception can be done below the noise floor, the use of the RSSI is not reliable enough. For CCA, there is a special channel activity detection (CAD) procedure that can be realized by a LoRa chip. To assess the CAD reliability, we experimentally tested how the CAD procedure can detect real transmissions on the wireless medium. We use a dedicated device to constantly perform CAD procedure and another device to send periodic messages. Fig. 4–22 shows two cases: (1) 44-byte message (40 bytes payload + 4-byte header) every 15 s with a CAD procedure every 100 ms and (2) 244-byte message (240 + 4) every 15 s with a CAD procedure every 1000 ms. As can be seen in Fig. 4–22 the LoRa CAD procedure can correctly detect all the LoRa transmission, and not only the preamble.

4.4.3 Adaptation from 802.11

As a first attempt toward a CSMA protocol for LoRa, we start by adapting the previously shown 802.11 CSMA protocol and not the 802.15.4 one, although 802.15.4 is widely used in WSN and early IoT implementation, for two reasons. The first reason is that LoRa network architecture is mainly a single-hop star topology from devices to gateway, which is very similar to the Wi-Fi topology with a base station. Therefore the concept and the management of the 802.11's random backoff timer after a busy channel looks efficient for such an environment. The second reason for not starting from 802.15.4 comes from its initial random waiting without channel sensing method that is more suitable for low-density networks than for high-density networks that will definitely be the case for LoRa networks.

To adapt the 802.11 CSMA protocol, we first need to define how the DIFS operation can be implemented. Usually, IFS should be related somehow to the symbol period T_{sym} . For LoRa, T_{sym} depends on BW and SF as follows: $T_{\text{sym}} = 2^S F / BW$. For instance, LoRa mode 1 uses $BW = 125$ kHz and $SF = 12$; therefore $T_{\text{sym}}^{\text{mode 1}} = 2^{12} / 125,000 = 0.032768$. In Ref. [37], it is reported that the CAD duration is between $1.75T_{\text{sym}}$ and $2.25T_{\text{sym}}$ depending on the spreading factor, see Fig. 4–23. We performed some experimental tests to verify the real

LoRa mode	BW/SF	Tsym (ms)	CAD duration (Tsym)45%	CAD duration (ms)	Experimental measures	
					min value	max value
1	BW125SF12	32.768	1.86	60.948	60	62
2	BW250SF12	16.384	1.86	30.474	29	31
3	BW125SF10	8.192	1.77	14.500	14	16
4	BW500SF12	8.192	1.86	15.237	15	16
5	BW250SF10	4.096	1.77	7.250	7	8
6	BW500SF11	4.096	1.81	7.414	7	9
7	BW250SF9	2.048	1.75	3.584	3	5
8	BW500SF9	1.024	1.75	1.792	1	3
9	BW500SF8	0.512	1.79	0.916	1	1
10	BW500SF7	0.256	1.92	0.492	0	1

FIGURE 4–23 Theoretical CAD duration and experimental measures.

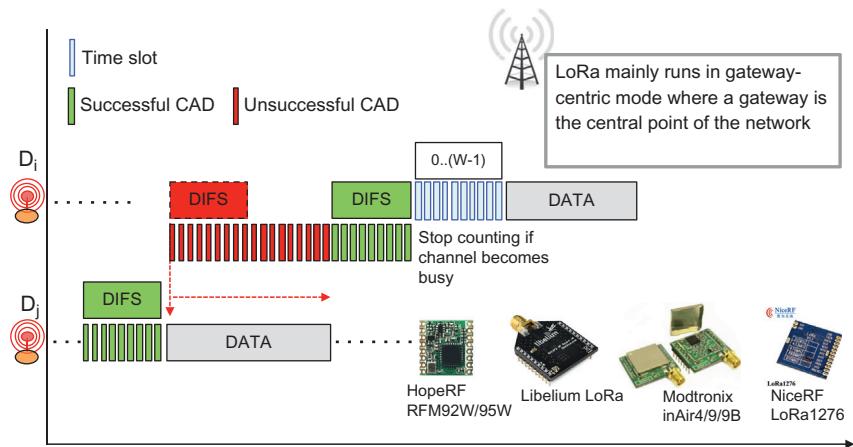


FIGURE 4–24 CSMA mechanism. Adapted from IEEE 802.11.

CAD duration against what is given in Ref. [37]: Fig. 4–23 also shows the minimum and the maximum values measured with a 1-ms accuracy clock (the Arduino millis() function). We can see that the measured CAD durations are quite consistent.

In our current implementation DIFS does not depend directly on T_{sym} but on the duration of the CAD mechanism; therefore we assign an integer number of CAD to DIFS. Our communication library provides a low-level doCAD(counter) function that takes an integer number of CAD, that is counter, performs sequentially the requested number of CAD, and returns to 0 if all CAD have been successful (no channel activity). If one CAD detects activity the function exits with a value greater than 0. The DIFS procedure shown in Fig. 4–24 works that way and once a failed CAD has been observed the node exits the DIFS procedure and continuously checks for a free channel.

In Fig. 4–24, DIFS is assigned nine CAD, which gives a duration of about $9 \times 61 \text{ ms} = 549 \text{ ms}$ for LoRa mode 1. At this point of the study, the duration of DIFS is not

really important as we only need to be able to assert a free channel for a given duration. The value of nine CAD provides enough time to detect channel activity and also provides the possibility to define a much shorter timer (using three CAD for instance), such as the 802.11's DIFS, to implement priority schemes is needed, and still be able to detect channel activity. Then, the random backoff timer is also defined as a number of CAD because the channel should be checked in order to freeze or continue the decrease of the backoff timer. The upper bound, W , of the random backoff timer can be set in relation to the number of CAD defined for DIFS. For instance, if $DIFS = 9$ CAD then W can be defined as $n \times DIFS$. For instance, if $n = 2$ then $W = 2 \times 9 = 18$ CAD.

It is also possible to double W for each retry (exponential backoff) until it reaches a maximum value. However, while 802.11 initiates a retry when no ACK is received after a given time, the usage of acknowledgment is not common in LoRa as it is very costly for the gateway (the gateway is considered as a normal node and therefore its radio duty cycle can be limited by regulations). Therefore there is no such retry concept with unacknowledged transmissions. Nevertheless, when 802.11 doubles W for each retry the underlying assumption for the transmission errors is a denser channel. Here, we can follow the same guideline and double W each time the channel cannot be found free for an entire DIFS, starting from the second DIFS attempt. In the current implementation, we set $W = 18$ CAD initially, and we can double it three times so the maximum value is $W = 144$ CAD which will give a maximum wait timer of 8784 ms for LoRa mode 1. If we add the value of the successful DIFS which is 9 CAD, that is 549 ms, then the maximum total wait time after a busy channel is about 9333 ms. which correspond roughly to the ToA of the maximum LoRa packet size. This property remains roughly true for all the defined LoRa modes and therefore can avoid waiting longer than necessary.

[Fig. 4-25](#) shows an experiment with an image sensor sending four image packets (about 240 bytes per packet), while another device (interactive device) is sending medium-size

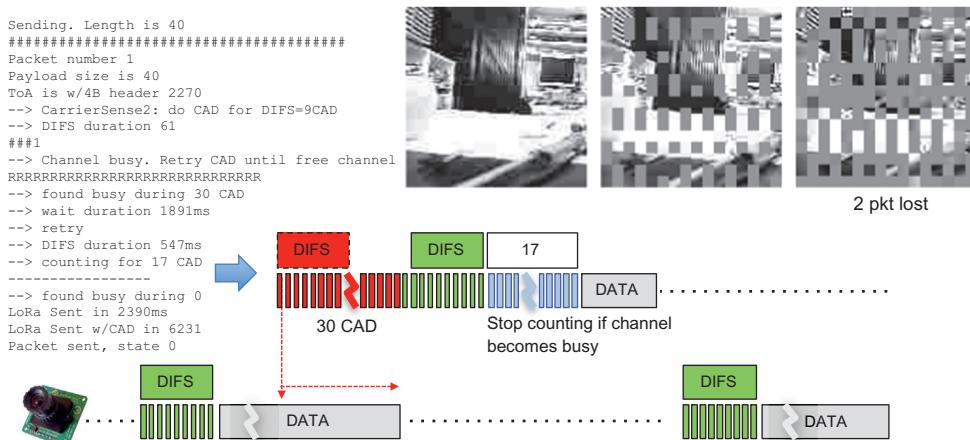


FIGURE 4-25 Experimental test of the proposed CSMA adaptation.

messages of 40 bytes. The output is from the interactive device and it can be seen that the adapted CSMA protocol can nicely avoid the collision by deferring the transmission of the interactive message. In the illustrated experiment, transmission is deferred only once before transmission succeeds as the time between two image packets is greater than a DIFS plus the random backoff timer of 17 CAD.

[Fig. 4–25](#) also shows the received image without any packet loss and two examples of received images when there is no channel access mechanism (pure ALOHA). In all our tests, the proposed CSMA protocol adapted from 802.11 and further referred to as $\text{CSMA}_{802.11}^{\text{LoRa}}$, totally avoids packet losses for both the image sensor and the interactive device.

4.4.4 Channel activity detection reliability issues

By testing further the CSMA mechanism in various long-range deployment, we observed a fast decrease of the CAD's reliability when distance increases: although a transmission can be successful at several kilometers, CAD starts to not reliably detect the whole transmission when the distance to the sender is about 1 km (with dense vegetation, CAD reliability can start to decrease even at 400 m). [Fig. 4–26](#) shows CAD reliability with the same traffic pattern previously shown in [Fig. 4–22](#) but with the sender and the device performing CAD separated by 400 m with some trees between them. As can be seen, the CAD procedure fails to detect channel activity many times during an ongoing transmission.

This CAD unreliability issue in real-world deployment scenario has a huge negative impact on the CS mechanism. For instance, in the previously proposed CSMA adaptation from 802.11, it is not possible anymore to rely on CAD to detect when the channel will become really free after a busy state nor to rely on a successful DIFS as a free channel indication to start transmission. However, what can be observed in [Fig. 4–22](#) and verified by the tests that we performed is that during a long transmission the probability that all CAD attempts fail is quite low. In all our tests, and up to 1 km in non line of sight conditions, there have always been some successful CAD during any transmission.

4.4.5 A solution to protect long messages

The CAD reliability issue raised previously calls for a different approach to prevent collisions. First, the previous DIFS is extended to the ToA of the longest LoRa packet in a given LoRa setting, for example 9150 ms for 255 bytes when $BW = 125 \text{ kHz}$ and $SF = 12$. During this

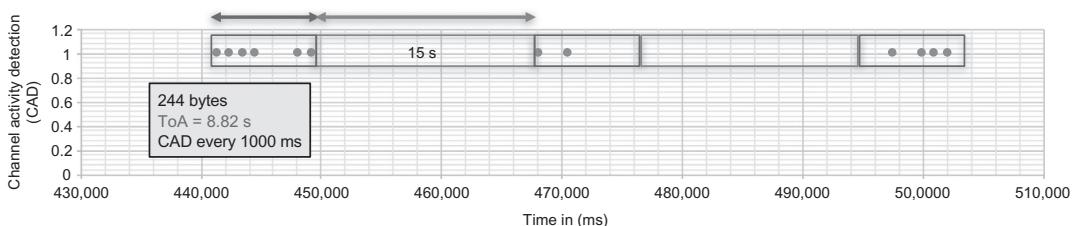


FIGURE 4–26 CAD fails to detect the activity of ongoing transmissions.

extended DIFS(ToA_{max}), CAD procedure is performed periodically (for instance every 1000 ms as in Fig. 4–22—bottom). The purpose of DIFS(ToA_{max}) is to maximize the probability to detect an ongoing transmission, which can possibly be a long message with many unsuccessful CADs, thus appearing by mistake as a short message.

Then, when a CAD fails during a DIFS(ToA_{max}), instead of continuously waiting for a free channel followed by a DIFS plus random backoff timer where CAD is checked constantly; here, there is a simple constant waiting period (pure delay) of ToA_{max} . Again, the purpose of the constant delay of ToA_{max} is to avoid performing CAD and transmission retries during the transmission of a possible long message, as a successful CAD does not guarantee a free channel. After the delay, the transmitter will try again to see a free channel for at least a DIFS(ToA_{max}) and the process continues until a maximum number of retries have been performed.

In all our experiments with the new proposed CSMA protocol, we can totally avoid packet losses for both the image sensor and the interactive device even when the nodes are 100s of meters away from each other. However, clearly, this is achieved at the cost of a much higher latency [38].

4.5 Studying large-scale LoRa deployments

The rapid growth in the field of WSNs and IoT entails the need of creating new simulators that have more specific capabilities to tackle interference and multipath propagation effects that are present in the wireless environment. Finding a suitable simulation environment that allows researchers to verify new ideas and compare proposed solutions in a virtual environment is a difficult task. Most of the existing open-source simulators are mainly used to study routing protocols and they offer limited real-time interference and radio propagation modeling features for smart cities and IoT applications.

The CupCarbon open-source simulator is the main simulation kernel of the French ANR project PERSEPTEUR that aims to develop models and algorithms for accurate simulations of signal propagation and interference in a 3D urban environment [39]. This simulator runs under the Java environment and can be downloaded from the Internet (<http://www.cupcarbon.com>). The main idea behind proposing CupCarbon is to keep simulation time short while taking into account a realistic evaluation of the wireless interference in a 3D environment with an accurately simulated radio channel. It supports wireless communication interference models such as Gaussian and α -stable models [40–42] and includes a 3D ray-tracing channel model. Focusing on physical layer, CupCarbon provides visualization of the impact of wireless interference and signal propagation. In addition to LoRa physical layer, CupCarbon also simulates the PHY layers of ZigBee (IEEE 802.15.4) and Wi-Fi (IEEE 802.11), making it suitable to study smart cities scenarios using all these wireless technologies (Fig. 4–27).

A WSN/IoT network can be prototyped with the CupCarbon's intuitive graphical interface that embeds OpenStreetMap framework to allow sensor nodes to be directly placed on the

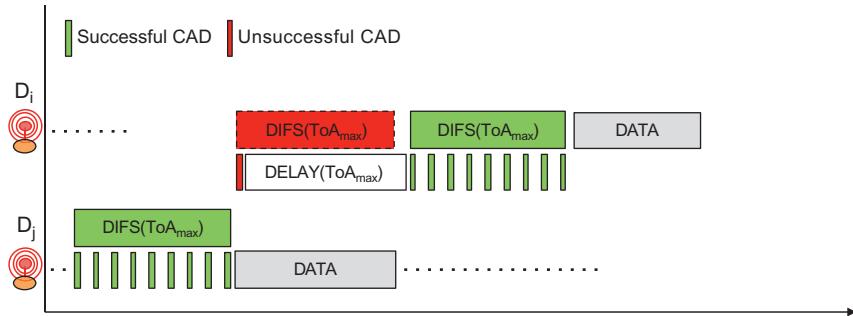


FIGURE 4–27 New CSMA proposition.

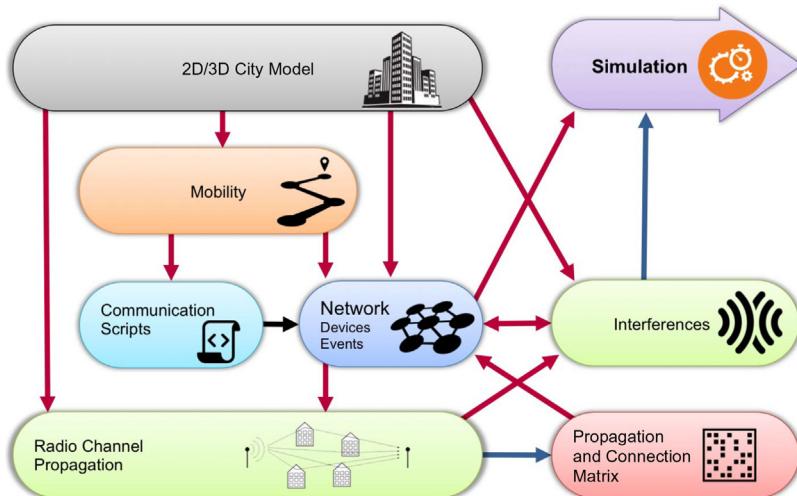


FIGURE 4–28 Main components of the CupCarbon architecture.

map. In addition, each sensor node can be individually configured by its command line with the script called SenScript. From SenScript, it is also possible to generate codes for hardware platforms such as Arduino.

4.5.1 The CupCarbon architecture

Fig. 4–28 shows the main modules of the CupCarbon simulator and we are going to present the main modules relevant for LoRa networks.

4.5.1.1 Two-dimensional/three-dimensional city model module

2D/3D visualization of the urban environment and the deployed network is an important part of a WSN/IoT simulation. The 3D environment helps to obtain an accurate deployment

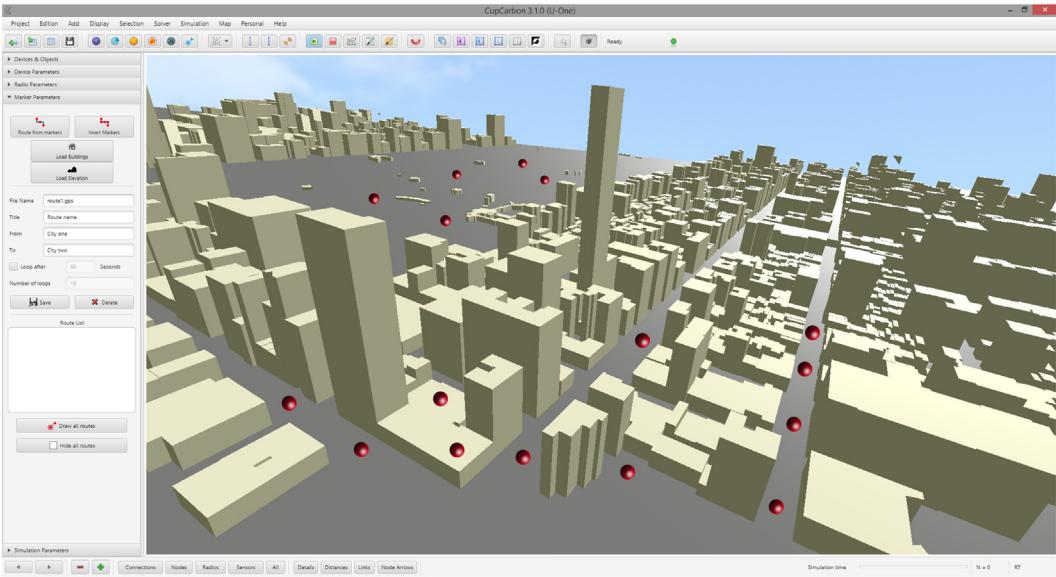


FIGURE 4–29 Example of a 3D display of a city in CupCarbon.

in which the elevation can be taken into account. Fig. 4–29 shows an example of a city displayed in the 3D environment of the CupCarbon simulator. The 3D environment of CupCarbon is composed of ground elevation, buildings, and various objects such as sensor nodes. More details can be found in Refs. [43,44].

4.5.1.2 Radio channel propagation module

Two radio propagation models are integrated into CupCarbon. The first one is a 2.5D based on a point to zone acceleration structure called visibility tree [45,46]. It provides the estimation of channel attenuation and the channel impulse response according to a large number of the receivers. The second model is a full 3D ray-tracing associated with a Monte Carlo algorithm. Resulting data help to determine the quality of the wireless channel between groups of the nodes and are used to decide whether a communication link can be established or not, see Fig. 4–30.

4.5.1.3 Interference module

This block is the core contribution of CupCarbon. It can be further divided into two categories:

1. **PHY layer:** a significant originality of CupCarbon is to propose very realistic models of many WSN/IoT physical layers. The evaluation of a link quality can now be based on accurate transmission conditions that consider the radio channel and the data encoding.

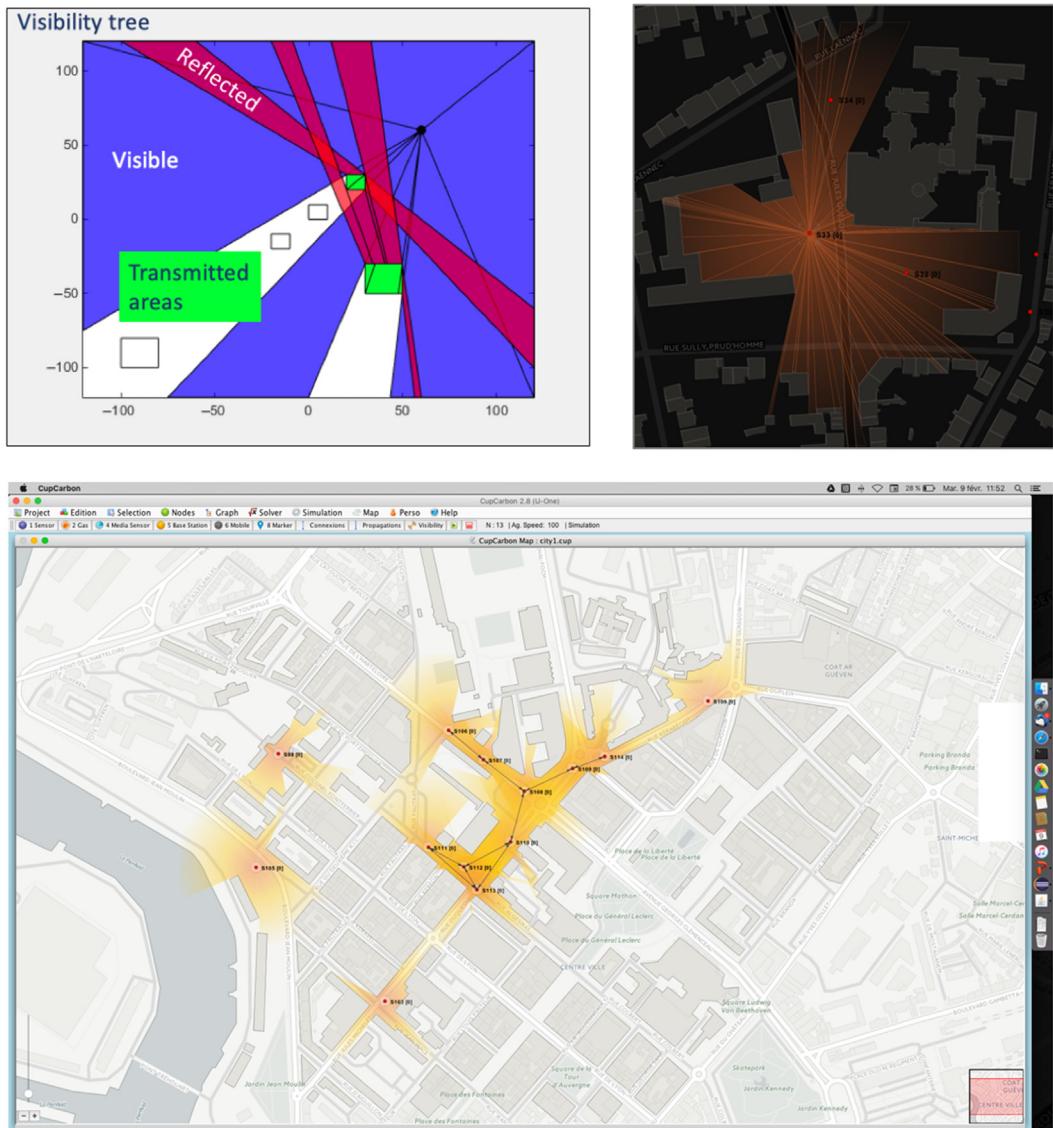


FIGURE 4–30 Channel propagation module.

2. Interference models: interference is a significant limiting factor in dense networks and CupCarbon proposes Gaussian or α -stable models. Radio in congested areas on the map can choose the impulsive interference model (α -stable model) while less congested areas can use the Gaussian model instead.

4.5.2 LoRa PHY/media control access integration in CupCarbon

Data from the MAC layer to the PHY layer are provided by the user from SenScript program. Then, a radio technology can be assigned: IEEE 802.11 or IEEE 802.15.4 or LoRa. Fig. 4–31 shows the block diagram of the whole procedure. With LoRa, the user must also assign the spreading factor SF between 7 and 12. The receiver technique should also be mentioned. By default, the receiver technique is none and the receiver will work normally without considering CE or successive interference cancellation. Default LoRa setting also uses the basic ALOHA approach and a node transmits a packet as soon as it has data to send. As described previously, ALOHA limits the system performance as it can only give 18.39% of maximum efficiency. However, CE and SIC can be used to enhance the system performance. More elaborated channel access methods such as the CSMA variants described in Section 4 can also be selected.

Fig. 4–32 shows the CupCarbon environment for the assignment of the parameters in case of selecting LoRa communication protocol. It shows interconnected sensor nodes placed arbitrarily. All the nodes are assigned the LoRa protocol. Nodes should be in the radio range of each other to communicate. The LoRa gateway can be defined as one of the nodes.

For the transmission, first the transmitter generates an upchirp signal using the input SF . By default the CR , SF , and BW values are considered to be 0, 8, and 125 kHz, respectively. Then, there are interleaving and modulation operations. After applying the transmission functions, the system calculates the parameters of the selected interference model, the Gaussian model, or the α -stable model, and applies the interference model to the transmitted signal. The receiver generates a downchirp signal using the appropriate SF and performs the demodulation.

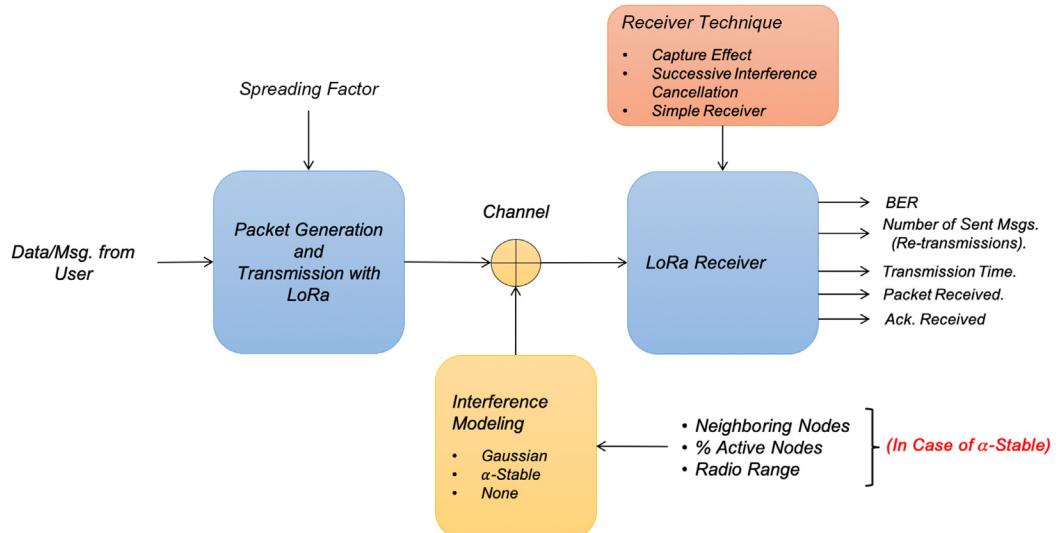


FIGURE 4–31 Schematic view of LoRa PHY, CE, and SIC in CupCarbon.

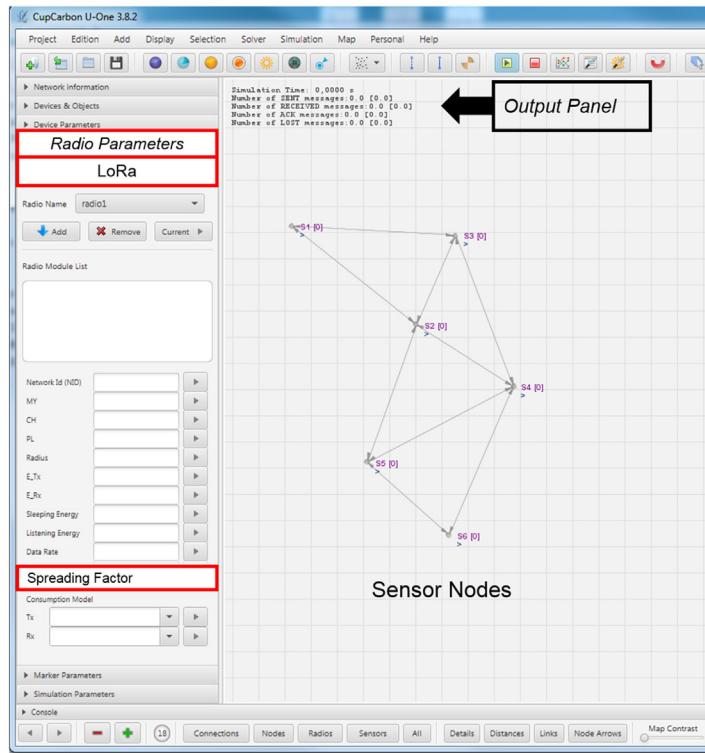


FIGURE 4–32 Radio parameter settings for LoRa in CupCarbon.

4.6 Conclusions

This chapter focused on LoRa LPWAN technology and presented the main issues and challenges of radio channel access in dense LoRa networks. Results from analytic analysis, simulations, and experimentations have been presented to illustrate the behavior of LoRa networks and try to assess its scalability in a dense environment. When considering only the LoRa physical layer without any LBT nor CSMA mechanism a deployed network is similar to a simple ALOHA system, which is known to have very small efficiency. However, this chapter also presented some positive effects such as CE and successive interference cancellation techniques. In addition, although not perfect, LoRa's SF quasiorthogonality can practically increase the scalability of deployed gateways, especially when they are based on radio concentrators capable of listening on at least eight frequencies, each of them accepting multiple SF values simultaneously. At the time of writing, 16-channel gateways are already available. At a higher level, legal regulations in many countries drastically limit the duty cycle or the ToA of a single transmission in current ISM license-free bands in order to prevent saturation of the radio channel. Some regulations also impose very tight frequency hopping constraints.

At the time of writing, given the relatively small number of deployed LoRa devices in the world, PRR is usually very good. In dense environments such as large cities, it is expected that the number of devices can be high. However, community-based initiatives also showed that the gateway density in such urban areas can also be very high, therefore increasing to some extent the number of available frequency channels. Last but not least, none of these measures can prevent packet collisions to happen and this chapter also reported how difficult it is to design an efficient CSMA-like mechanism because a reliable CCA is difficult to achieve. In this context, higher-level coordination mechanisms may be needed to further improve LoRa scalability.

4.7 Acknowledgments

This work was supported by the EU H2020 RIA WAZIUP project under grant agreement no. 687607 and by the ANR PERSEPTEUR project.

References

- [1] TTN mapper. <<https://ttnmapper.org/>> (accessed 06.04.19).
- [2] D. Akerman, High altitude ballooning. <<https://www.daveakerman.com/>> (accessed 06.04.19).
- [3] Ground breaking world record. <<https://bit.ly/2wOtad4>> (accessed 06.04.19).
- [4] T. Telkamp, LoRa transmission from low orbit satellite. <<https://www.thethingsnetwork.org/article/lora-transmission-from-low-orbit-satellite>> (accessed 06.04.19).
- [5] Semtech. <<http://www.semtech.com/images/datasheet/an1200.22.pdf>> (accessed 15.11.17).
- [6] S. Andrew, D.J.W. Tanenbaum, *Computer Networks*, Prentice-Hall, 2010, ISBN: 9780133485936.
- [7] B. Reynders, S. Pollin, Chirp spread spectrum as a modulation technique for long range communication, in: 2016 Symposium on Communications and Vehicular Technologies (SCVT), 2016, pp. 1–5. Available from: <https://doi.org/10.1109/SCVT.2016.7797659>.
- [8] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, I. Tinnirello, Impact of LoRa imperfect orthogonality: analysis of link-level performance, *IEEE Commun. Lett.* 22 (4) (2018) 796–799.
- [9] A. Rahmadhani, F. Kuipers, When LoRaWAN frames collide, in: ACM WiNTECH, 2018.
- [10] F. Adelantado, X. Vilajosana, P. Tuset-Peiró, B. Martínez, J. Melià-Seguí, T. Watteyne, Understanding the limits of LoRaWAN, *IEEE Commun. Mag.* 55 (9) (2017) 34–40. Available from: <https://doi.org/10.1109/MCOM.2017.1600613>.
- [11] O. Georgiou, U. Raza, Low power wide area network analysis: can LoRa scale? *IEEE Wirel. Commun. Lett.* 6 (2) (2017) 162–165. Available from: <https://doi.org/10.1109/LWC.2016.2647247>.
- [12] F.V.D. Abeele, J. Haxhibeqiri, I. Moerman, J. Hoebeke, Scalability analysis of large-scale LoRaWAN networks in ns-3, *IEEE Internet Things J.* 4 (6) (2017) 2186–2198. Available from: <https://doi.org/10.1109/IOT.2017.2768498>.
- [13] J. Haxhibeqiri, F.V.D. Abeele, I. Moerman, J. Hoebeke, LoRa scalability: a simulation model based on interference measurements, *Sensors* 17 (6) (2017) 1193. Available from: <https://doi.org/10.3390/s17061193>.
- [14] M.C. Bor, U. Roedig, T. Voigt, J.M. Alonso, Do LoRa low-power wide-area networks scale? *Proceedings of MSWiM 2016*, ACM, 2016, pp. 59–67.

- [15] LoRaWAN 1.1 specification. <<https://lora-alliance.org/resource-hub/lorawantm-specification-v11>> (accessed 06.04.19).
- [16] S. Li, U. Raza, A. Khan, How agile is the adaptive data rate mechanism of LoRaWAN? in: IEEE Global Communications Conference, GLOBECOM 2018, 9–13 December 2018, Abu Dhabi, United Arab Emirates, 2018, pp. 206–212. Available from: <https://doi.org/10.1109/GLOCOM.2018.8647469>.
- [17] D. Zorbas, K.Q. Abdefadeel, V. Cionca, D. Pesch, B. O'Flynn, Offline scheduling algorithms for time-slotted LoRa-based bulk data transmission, in: IEEE 5th World Forum on Internet of Things (WF-IoT), 2019.
- [18] B. Reynders, Q. Wang, P. Tuset-Peiró, X. Vilajosana, S. Pollin, Improving reliability and scalability of LoRaWANs through lightweight scheduling, *IEEE Internet Things J.* 5 (3) (2018) 1830–1842.
- [19] J. Haxhibeqiri, I. Moerman, J. Hoebeke, Low overhead scheduling of LoRa transmissions for improved scalability, *IEEE Internet Things J.* 6 (2018) 3097–3109.
- [20] ETSI, Electromagnetic compatibility and radio spectrum matters (ERM); short range devices (SRD); radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW; part 1: technical characteristics and test methods, 2012.
- [21] V241 EE. Electromagnetic compatibility and radio spectrum matters (ERM) short range devices (SRD); radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW, technical report, European Telecommunications Standards Institute, Sophia Antipolis Cedex, France, 2012.
- [22] M. Bor, J. Vidler, U. Roedig, LoRa for the Internet of things, in: Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks, 2016, pp. 361–366.
- [23] S. Kosunalp, P. Mitchell, D. Grace, T. Clarke, Experimental study of the capture effect for medium access control with ALOHA, *ETRI J.* 37 (2) (2015) 359–368. Available from: <https://doi.org/10.4218/etrij.15.0114.1369>.
- [24] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, D. Culler, Exploiting the capture effect for collision detection and recovery, The Second IEEE Workshop on Embedded Networked Sensors, 2005, EmNetS-II, 2005, pp. 45–52. Available from: <https://doi.org/10.1109/EMNETS.2005.1469098>.
- [25] T. Voigt, M. Bor, U. Roedig, J. Alonso, Mitigating inter-network interference in LoRa networks, in: Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks (EWSN'17), 2017.
- [26] N.E. Rachkidy, A. Guittot, M. Kaneko, Decoding superposed LoRa signals, in: IEEE LCN, 2018.
- [27] J.G. Andrews, Interference cancellation for cellular systems: a contemporary overview, *IEEE Wirel. Commun.* 12 (2) (2005) 19–29. Available from: <https://doi.org/10.1109/MWC.2005.1421925>.
- [28] H. Tian, M.A. Weitnauer, G. Nyengele, Optimized gateway placement for interference cancellation in transmit-only LPWA networks, *Sensors* 18 (11) (2018) 3884. Available from: <https://doi.org/10.3390/s18113884>.
- [29] Y. Mo, C. Goursaud, J. Gorce, On the benefits of successive interference cancellation for ultra narrow band networks: theory and application to IoT, in: IEEE International Conference on Communications, ICC 2017, 21–25 May 2017, Paris, France, 2017, pp. 1–6. Available from: <https://doi.org/10.1109/ICC.2017.7996900>.
- [30] M. Kaynia, N. Jindal, Performance of ALOHA and CSMA in spatially distributed wireless networks, in: Proceedings of IEEE International Conference on Communications (ICC), 2008.
- [31] Y. Yang, T.S.P. Yum, Delay distributions of slotted ALOHA and CSMA, *IEEE Trans. Commun.* 51 (2003).
- [32] F.A. Tobagi, Distribution of packet delay and interdeparture time in slotted aloha and carrier sense multiple access, *J. Assoc. Comput. Mach.* 29 (1982).

- [33] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, *IEEE/ACM Trans. Netw.* 12 (3) (2004) 493–506.
- [34] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys '04, 2004, pp. 95–107. ISBN 1-58113-879-2.
- [35] M. Buettner, G.V. Yee, E. Anderson, R. Han, X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys '06, 2006, pp. 307–320.
- [36] A. Bachir, M. Dohler, T. Watteyne, K.K. Leung, MAC essentials for wireless sensor networks, *IEEE Commun. Surv. Tutor.* 12 (2) (2010) 222–248.
- [37] Semtech, SX1272/73—860 MHz to 1020 MHz low power long range transceiver, rev.2-07/2014, 2014.
- [38] C. Pham, Investigating and experimenting CSMA channel access mechanisms for long-range LoRa IoT networks, in: Proceedings of the IEEE WCNC, 2018.
- [39] M. Saoudi, F. Lalem, A. Bounceur, R. Euler, M.T. Kechadi, A. Laouid, et al., D-LPCN: a distributed least polar-angle connected node algorithm for finding the boundary of a wireless sensor network, *Ad Hoc Netw. J.* 56 (1) (2017). Available from: <https://doi.org/10.1016/j.adhoc.2016.11.010>.
- [40] A. Bounceur, O. Marc, M. Lounis, J. Soler, L. Clavier, P. Combeau, et al., CupCarbon-Lab: An IoT emulator, in: 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC), 2018, pp. 1–2. Available from: <https://doi.org/10.1109/CCNC.2018.8319313>.
- [41] M. Kamal, M. Lounis, A. Bounceur, M.T. Kechadi, CupCarbon: a multi-agent and discrete event wireless sensor network design and simulation tool, in: SimuTools, 2014.
- [42] CupCarbon, ANR project PERSEPTEUR, CupCarbon simulator. <<http://www.cupcarbon.com>>. (accessed 15.11.15).
- [43] CupCarbon: ray-tracing technique with 3D visualization. <<https://www.youtube.com/watch?v=8HBhjVNIZgI>>. (accessed 30.04.19).
- [44] CupCarbon: integration of 3D maps. <<https://www.youtube.com/watch?v=5hyEXrVOPQU>>. (accessed 30.04.19).
- [45] T. Alwajeh, P. Combeau, A. Bounceur, R. Vauzelle, Efficient method for associating radio propagation models with spatial partitioning for smart city applications, 2016, pp. 1–7. Available from: <https://doi.org/10.1145/2896387.2901918>.
- [46] T. Alwajeh, P. Combeau, R. Vauzelle, A. Bounceur, A high-speed 2.5D ray-tracing propagation model for microcellular systems, application: smart cities, in: 2017 11th European Conference on Antennas and Propagation (EUCAP), 2017, pp. 3515–3519. Available from: <https://doi.org/10.23919/EuCAP.2017.7928760>.