RSSF: Towards Real-Time Decoding of LoRa Packets without Prior Knowledge of their Spreading Factor

Pei Tian*†‡, Carlo Alberto Boano‡, Markus Schuß‡, and Jianming Wei*†
*Shanghai Advanced Research Institute, Chinese Academy of Sciences, China
†University of Chinese Academy of Sciences, China
‡Institute of Technical Informatics, Graz University of Technology, Austria
Email: {tianpei2018, wjm}@sari.ac.cn, {cboano, markus.schuss}@tugraz.at

***Abstract*—The selection of the spreading factor (SF) has important implications on the radio on-time, energy consumption, achievable data rate, and communication range of LoRa devices. In practical applications, LoRa packets can only be exchanged when the SF between transmitter and receiver matches. To ensure that this is the case, current approaches either statically hard-code the SF used to communicate between two devices, or negotiate which SF to use through handshaking mechanisms. Unfortunately, statically assigning the SF may lead to sub-optimal performance, and changing the assignment at runtime through a negotiation process incurs a significant overhead in terms of both latency and energy consumption. In this paper, we propose RSSF, a scheme that allows an off-the-shelf LoRa device to receive and decode a packet without prior knowledge of the SF used by the transmitter. RSSF leverages the observation that the SF with which a packet was sent can be inferred by analyzing the received signal strength (RSS) samples captured while receiving the first preamble symbols, and by characterizing their periodicity. In real-world systems, however, the waveform obtained by sampling the RSS during the reception of the first preamble symbols contains several spikes due to the receiver's DC offset cancellation, which makes it difficult to accurately identify periods. We show that this problem can be solved by letting an off-the-shelf LoRa receiver sample the RSS on a frequency that is shifted by half of the bandwidth from the original frequency at which the packet was transmitted. We then implement a lightweight algorithm that determines the SF by partitioning the RSS samples into sliding windows of different size (corresponding to each possible SF) and by measuring the zero-crossing intervals for each window size. We evaluate RSSF's performance experimentally using both software-defined radios and off-the-shelf LoRa nodes, showing that RSSF can accurately determine the SF within the first 5 preamble symbols.**

***Index Terms*—LoRa, Spreading factor, Received signal strength, SX1276, USRP, Zero-crossing algorithm.**

I. Introduction

Low-power wide area networks (LPWANs) have emerged as an attractive technology for connecting low-cost energy-constrained devices over large areas. Among LPWAN technologies, LoRa has gained significant popularity due to its unique properties, including the openness of its ecosystem, the absence of subscription costs (i.e., users can set up their own devices operating in unlicensed ISM bands), low power consumption, and – most importantly – its high receiver sensitivity [1]. The latter is attributed to the use of chirp spread spectrum (CSS) modulation, which is robust against channel interference. The signal is modulated as linear chirp frequencies that sweep within a specific bandwidth (BW), either increasing (up-chirps) or decreasing (down-chirps) over time. LoRa transceivers enable end-users to fine-tune physical layer parameters to meet diverse application requirements. A key parameter is the spreading factor (SF), which affects the radio on-time, energy consumption, data rate, and communication range of LoRa devices. Many studies have investigated the performance of different SFs and other physical layer parameters as well as their impact on receiver sensitivity and communication range under various link conditions [2]–[4].

The need to coordinate the SF. As different SFs are orthogonal to each other, LoRa packets can only be received when the SF with which they are sent is known in advanced by the receiver. This is not the case when transmitting packets to a LoRaWAN gateway, as the latter can support eight channels and receive packets sent with all SFs simultaneously. Commercial LoRa end-devices, however, typically only support reception on one channel. Therefore, to ensure that packets are correctly received, current approaches use either a fixed SF at the receiver and a variable SF at the transmitter (i.e., they transmit the same packet multiple times using different SFs [5], [6]), or let devices negotiate the SF through handshaking mechanisms [7], [8]. However, these methods result in additional overhead and require multiple transmission attempts, which negatively affects the responsiveness of a system and its energy efficiency. An alternative is to statically assign the SF used by the different links based on the characteristics of the network, and several works investigated this in the context of single-hop LoRa networks [8]–[12]. However, achieving an optimal assignment of SFs through mathematical and statistical models becomes more difficult in multi-hop networks [13], [14]: as a result, existing LoRa mesh solutions [15]–[17] still rely on manually-configured SF, and there is a lack of methods for SF coordination among nodes.

Contributions. In this paper, we enable the reception of packets on off-the-shelf LoRa end-devices without prior knowledge of the SF with which they were sent. Specifically, we present RSSF, the first[1] scheme that enables an automated detection of the SF by sampling the signal strength of the first preamble symbols being received. RSSF takes advantage of the characteristics of LoRa's CSS modulation and packet structure, as well as of the effect that the low-pass filter (LPF) incorporated in the transceiver has on the received signal strength (RSS).

[1] In a prior demo abstract [18], we have shown a preliminary implementation of this concept. In this paper, we provide the theoretical explanation of its working principle, enhance its effectiveness, and evaluate its performance.

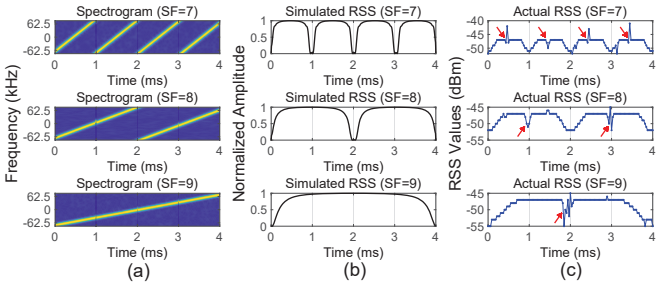


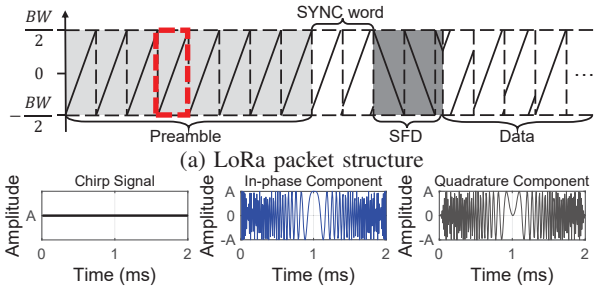
Fig. 1: Spectrogram of the up-chirps in a LoRa preamble sent with SF = 7, 8, 9, and BW = 125 kHz (a); the corresponding theoretical RSS after being filtered by a LPF (b); the RSS actually obtained with an off-the-shelf LoRa receiver (c).

The key principle behind RSSF is illustrated in Fig. 1, which shows the spectrogram of LoRa up-chirps transmitted using different SFs (a), and the corresponding theoretical RSS at the receiver (b). One can clearly see how the transitions between consecutive up-chirps result in a drop in RSS. Such drops are caused by the non-ideal behaviour of the receiver's LPF (see Sec. II), and occur whenever the chirp frequency changes from $\frac{BW}{2}$ to $-\frac{BW}{2}$. How often this occurs depends on the SF used, as this affects how quickly up-chirps are transmitted: in the Fig. 1a example, every 1, 2, 4 ms for SF = 7, 8, 9, respectively.

Therefore, by analyzing the RSS waveform during the reception of the first preamble symbols, and by identifying the periodicity of the drops in signal's amplitude using a zero-crossing algorithm, RSSF can determine the SF of the packet and configure the radio to switch to the identified SF and successfully complete the packet's reception.

However, RSSF needs to address two key challenges. The first one is illustrated in Fig. 1c, which shows the RSS waveform that was measured by an off-the-shelf LoRa receiver. The red arrows mark sudden changes in the signal's amplitude that occur in the middle of the transmission of each chirp, regardless of the SF at which it was sent. These sudden changes makes it difficult to reliably characterize the periodicity of the intended drops in signal amplitude shown in Fig. 1b. We show that these sudden changes are due to direct current (DC) offset cancellation (see Sec. III), and that they can be eliminated by recording the RSS waveform on a channel whose frequency is $\frac{BW}{2}$ higher than the nominal one (see Sec. IV). The second challenge is how to accurately but quickly determine the SF, even in the presence of noise. In fact, one needs to configure the radio with the identified SF within a short time, as at least two preamble symbols need to be received to correctly lock on the packet [19]. To this end, we implement a lightweight algorithm that determines the SF by partitioning the RSS samples into sliding windows of different size (corresponding to each possible SF), and by computing the interval of zero-crossing points for each window size (see Sec. IV).

We finally evaluate the effectiveness of RSSF experimentally: our results show that we can determine the SF within the first 5 preamble symbols and with an accuracy above 95%, while retaining a good detection performance even when receiving packets with a low signal-to-noise ratio (see Sec. V).



(a) LoRa packet structure

(b) Amplitude and IQ components of the chirp signal

Fig. 2: LoRa chirp signal at SF=8 and BW=125 kHz.

II. Working Principle

After providing background info about LoRa's CSS modulation and packet structure (Sec. II-A), we describe *why* RSS measurements can be leveraged to detect the SF (Sec. II-B) and *how* RSSF does so using a zero-crossing algorithm (Sec. II-C).

A. LoRa Preliminaries

LoRa employs CSS modulation, in which the signal is modulated as linear chirp frequencies that sweep across a given BW (125, 250, or 500 kHz), either increasing from $-BW/2$ to $+BW/2$ (up-chirp), or decreasing from $+BW/2$ to $-BW/2$ (down-chirp) over a chirp duration T. The SF specifies the frequency sweeping rate, i.e., in a chirp duration T, there are 2^{SF} individual frequencies (i.e., 2^{SF} chips per chirp), and the chip period is $1/BW$. The total chirp duration T can hence be represented as $T = 2^{SF}/BW$. For a given bandwidth, T increases exponentially with the SF.

Fig. 2a illustrates the structure of a LoRa packet in time and frequency. A LoRa packet contains the preamble, two up-chirps as the sync word, 2.25 down-chirps as the start-of-frame-delimiter (SFD), and the data field consisting of an optional header, payload, and optional cyclic redundancy check (CRC). The preamble includes a sequence of identical up-chirps which are utilized for synchronization between transmitter (TX) and receiver (RX), and whose number is configurable (typically set to 8). The up-chirps in the preamble have the same starting frequency at $-BW/2$ and are called *basic up-chirps* [20]; the up-chirps in the payload have different starting frequencies depending on the actual data. The instantaneous frequency of a basic up-chirp can be expressed as:

$$f(t) = -\frac{BW}{2} + kt, 0 \leq t < T, \quad (1)$$

where $k = BW/T$ is frequency sweep rate. By integrating this linearly changing frequency w.r.t. time, the basic up-chirp $x(t)$ can be represented as:

$$x(t) = Ae^{j2\pi \int_0^t f(t)dt} = Ae^{j2\pi(-\frac{BW}{2}t + \frac{1}{2}kt^2)}, 0 \leq t < T, \quad (2)$$

where A is the amplitude of the chirp. Fig. 2b shows the modulated signal amplitude, in-phase component and quadrature-phase component of the basic up-chirp during T. The signal amplitude remains constant at $A = \sqrt{I^2 + Q^2}$ throughout the chirp's transmission. One would hence expect the RSS to remain also constant when receiving the chirp.

B. Leveraging the Received Signal Strength

However, as shown in Fig. 1b, when receiving two consecutive basic up-chirps, the RSS drops during the transition from $+BW/2$ to $-BW/2$, i.e., there is a significant energy loss (and consequent lower amplitude in the RSS waveform) in-between the reception of two basic up-chirps. The drop is caused by the LPF of the receiver, which filters the receiving channel with cut-off frequencies at $-BW/2$ and $+BW/2$ for noise reduction. Ideally, the RSS should remain at a constant amplitude A after passing through the LPF. However, in practice, the response curve of a LPF is a gradually-attenuating process, which causes some loss of energy around the cut-off frequencies. Because the basic up-chirp has the lowest and highest frequencies at the beginning and end of the chirp, its signal amplitude drops at the beginning/end of T, causing the measured RSS waveform to exhibit a periodic drop in amplitude every T. Since the chirp duration T varies exponentially for different SFs, one can leverage the periodicity of the RSS samples to infer T, and hence, the employed SF.

C. Zero-Crossing on RSS samples

Several algorithms are available for calculating the period of signal waveforms, such as the Fourier transform [21], Hilbert transform, eigen value decomposition (EVD) [22], and zero-crossing. Among them, zero-crossing is particularly appealing for the design of `RSSF`, as its low computational complexity fits the requirements of LoRa end-devices, which typically have limited energy, memory, and processing power. When using zero-crossing in `RSSF`, given a number of RSS samples $\boldsymbol{R}_s \triangleq [s_1, \ldots, s_L]$, where L is the number of samples, the receiver first calculates its average value $E(\boldsymbol{R}_s)$ as the reference level. The zero-crossing (ZC) points are identified by observing the samples s_n and s_{n+1} in $\boldsymbol{R}_s$ that cross the reference level $E(\boldsymbol{R}_s)$, which satisfy either $s_n < E(\boldsymbol{R}_s)$ and $s_{n+1} > E(\boldsymbol{R}_s)$, or $s_n > E(\boldsymbol{R}_s)$ and $s_{n+1} < E(\boldsymbol{R}_s)$, where n is the sample index. A threshold value ρ between two points is used to avoid generating ZC points at very small fluctuations. The ZC points marked in set $\boldsymbol{Z}(\boldsymbol{R}_s)$ are determined as:

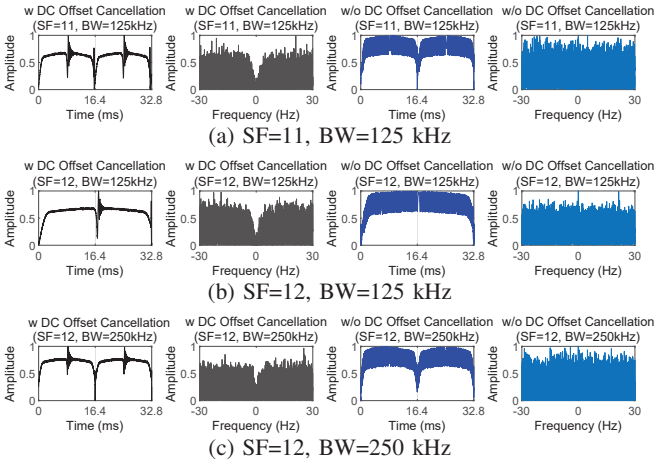
$$\boldsymbol{Z}(\boldsymbol{R}_s) = \{n \mid a_n a_{n+1} < 0 \text{ and } |s_{n+1} - s_n| > \rho\}, \quad (3)$$

where $a_n = s_n - E(\boldsymbol{R}_s)$. We use $\hat{T}$ as the average interval of ZC points in a given window size, which is expressed as:

$$\hat{T} = \frac{L}{N_z + 1}, \quad (4)$$

where N_z is the total number of ZC points $\boldsymbol{Z}(\boldsymbol{R}_s)$.

Given a RSS sampling time of T_r and setting the zero-crossing window size $L = \lceil T/T_r \rceil$, so that L is equal to the number of RSS samples in one chirp duration, two ZC points should appear in the window. The computed $\hat{T}$ can be obtained as $L/3$ from Eq. (4). For unknown SF packets, we make an initial assumption about the SF, and calculate $\hat{T}$ using a window size L at $\lceil T/T_r \rceil$. We then compare the computed $\hat{T}$ with the expected value of $L/3$ to determine whether the assumed SF is correct. A match between the two values suggests that the assumed SF is very likely the SF used for transmission.



(a) SF=11, BW=125 kHz

(b) SF=12, BW=125 kHz

(c) SF=12, BW=250 kHz

Fig. 3: RSS of basic up-chirps with (w) and without (w/o) DC offset cancellation for different SFs and bandwidths.

III. From Theory to Practice

Fig. 1c shows the RSS waveform obtained using an off-the-shelf `SX1276` LoRa end-device [19] sampling the first preamble symbols of packets sent with different SFs at the highest possible rate, i.e., every 31.8 µs. Regardless of the SF, sudden changes are observed in the RSS waveform (marked with red arrows): these occur in the middle of each chirp, and appear as either spikes or drops in signal amplitude. As `RSSF` aims to detect the periodic drop in amplitude every T, these sudden changes could be mistakenly identified as the beginning of the next chirp, affecting the correct SF detection. Although setting an appropriate threshold would allow to filter out these sudden changes, it is difficult to determine its value in advance, as conservative thresholds may prevent the SF detection to succeed at low signal-to-noise ratios (SNRs).

The impact of DC offset cancellation. These sudden changes are caused by DC offset cancellation, which is introduced in off-the-shelf LoRa receivers to mitigate DC offset voltage due to self-mixing in the mixer [23]. Although essential, DC offset cancellation introduces a huge energy loss at the RX channel frequency, resulting in sudden changes in the RSS waveform. We confirm this using an `USRP B205mini-i` software-defined radio (SDR), which allows for faster sampling and more fine-grained control over the RX, and show our results in Fig. 3. Specifically, Fig. 3a shows the spikes in signal amplitude over time (leftmost plot), and the sudden drop in the frequency domain (derived with an FFT) around the RX channel frequency (middle-left plot) when DC offset cancellation is enabled[2]. The drop in RSS at the RX channel frequency can be considered equivalent to the effect of a high-pass filter with cut-off frequency close to 0 Hz. When DC offset cancellation is disabled, the sudden changes are no longer visible, neither in the signal amplitude over time (middle-right plot), nor over frequency (right plot), i.e., the energy distribution across the entire bandwidth is uniform. Similar observations apply for different SFs (Fig. 3b) and bandwidths (Fig. 3c).

[2]To better observe the effect, the plot only shows a 60 Hz area (±30 Hz) around the RX channel frequency instead of the entire bandwidth of the chirp.

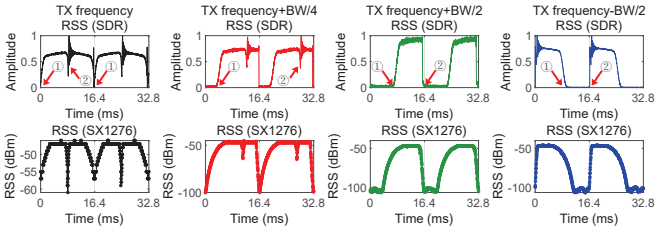


Fig. 4: Comparison of the RSS and in-phase real component obtained from an SDR (USRP B205mini-i) and the RSS obtained from an off-the-shelf LoRa device (SX1276) on the original and shifted RX channel frequency.

Unfortunately, commercial LoRa devices do not support disabling DC offset cancellation, and there are no related registers in their datasheet [19]. Therefore, we need to resort to alternative methods to avoid the sudden fluctuations in RSS.

IV. Making RSSF Applicable in Real-World Systems

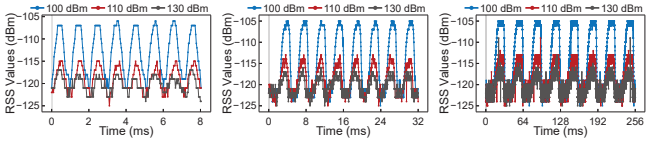
To avoid sudden changes in the RSS waveform, we propose to sample the RSS at a frequency that is shifted by $\pm BW/2$ compared to that at which the packet was sent (Sec. IV-A). We show that this method is effective on LoRa receivers for different SFs and SNRs (Sec. IV-B), and detail how RSSF derives the SF based on the ZC intervals (Sec. IV-C). We finally summarize RSSF's architecture and algorithm (Sec. IV-D).

A. RX Channel Shifting

We measure the RSS waveform obtained when shifting the RX channel frequency by different offsets (0, $+BW/4$, $+BW/2$, and $-BW/2$) compared to the original frequency used by the transmitter (which sends packets at a center frequency of 434 MHz with SF=11 and BW=125 kHz). Fig. 4 shows the amplitude of such RSS waveforms, obtained with the USRP B205mini-i SDR (top) and the SX1276 LoRa device (bottom). For both the SDR and the off-the-shelf LoRa device, the signal experiences energy loss on the cut-off frequency of the LPF ①, and a sudden change in energy in the middle of the chirp due to the DC offset cancellation ②. When the offset is set to $+BW/4$, the first quarter of the chirp is filtered by the receiver's LPF, resulting in a clearly-distinguishable RSS amplitude between $T/4$ and T, whereas the sudden energy change occurs at $3T/4$. When the RX channel frequency is shifted by $+BW/2$, the first half of the chirp is filtered by the LPF, and the sudden energy change occurs at the end of the chirp. Similarly, when the RX channel is shifted by $-BW/2$, the sudden energy change occurs at the beginning of the chirp. From the above observations, RSSF sets the RX channel to $\pm BW/2$ to obtain a clear RSS waveform, where the period is recognizable through a zero-crossing algorithm. The clear RSS waveform can then be used to determine the SF accurately, after which the RX channel frequency can be restored to the original frequency at which the packet was transmitted to decode the following part of the LoRa packet.

B. RSS Sampling on Commercial LoRa Devices

We verify next whether the approach proposed in Sec. IV-A is effective on off-the-shelf LoRa receivers for different SFs and SNRs. To this end, we let the SX1276 obtain the RSS waveform when sampling the preamble of packets sent with SF = 7, 9, 12 and a transmission power of 0 dBm[3]. Transmitter and receiver are connected with an SMA cable, a fixed 40 dBm attenuator, as well as a 90 dBm programmable attenuator (Minicircuits RCDAT-8000-90), which we use to artificially lower the RSS of received packets. Specifically, we run experiments with an attenuation of 100, 110, and 130 dBm. Fig. 5 shows how, thanks to the frequency shift introduced in Sec. IV-A, the periodicity of the RSS samples remains clear even when the signal is highly attenuated, which is key in accurately determining the SF. When using the highest SF and attenuation level (SF=12 and 130 dBm), as expected, the peaks and valleys of the RSS waveform have similar values, which makes it harder to accurately recognize the periodicity.



(a) SF=7 (b) SF=9 (c) SF=12

Fig. 5: RSS samples obtained by a SX1276 device when receiving packets sent with SF = 7, 9, 12 and an attenuation of 100 dBm, 110 dBm, and 130 dBm.

C. Determination of SF

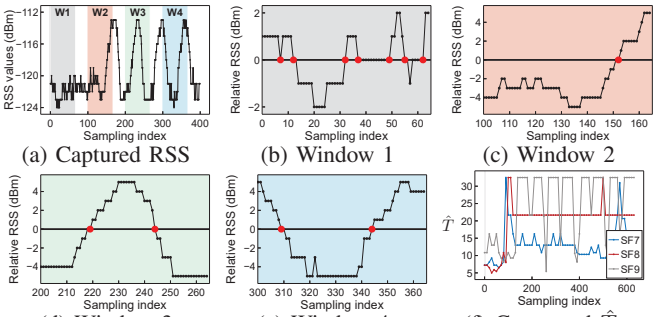
As discussed in Sec. II-C, to determine the SF, RSSF employs an algorithm that computes the average time between ZC points $\hat{T}$ in a given window size $L = \lceil T/T_r \rceil$. As for a chirp duration T two ZC points are expected, $\hat{T}$ should be $\approx L/3$. For example, when SF = 8, T = 2.04 ms: when using the highest RSS sampling frequency T_r = 31.8 µs, the window size to check whether the packet was sent with SF = 8 is L = 65.

Fig. 6a shows an exemplary RSS capture of the first four preamble symbols of a packet sent with SF = 8. Figs. 6b to 6e show four exemplary windows (W1–W4) of 65 sampling points within this capture[4]. In W1, the preamble has not been received yet, and there are multiple ZC points due to the noise on the channel, which results in $\hat{T} = L/7$ (Fig. 6b). In W2, the first preamble symbol is received, resulting in a RSS increase: this leads to one ZC point and hence $\hat{T} = L/2$ (Fig. 6c). In W3, a preamble symbol is contained, resulting in the expected two ZC points and $\hat{T} = L/3$ (Fig. 6d). The same applies to W4, which captures the second and first half of two consecutive symbols (Fig. 6e). Therefore, the presence of a preamble sent with the expected SF produces a consistent value $\hat{T} = L/3$.

Fig. 6f shows the computed $\hat{T}$ over time when receiving a packet sent with SF = 7, 8, 9 with a fixed window size L = 65. For SF = 7, a relatively constant $\hat{T}$ can be observed, but its value is less than $L/3$. For SF = 9, a constant $\hat{T}$ throughout the

[3]The SX1276 provides two registers for reading RSS values, namely RegRssiValue and RegPktRssiValue. The former can be used to read the current RX channel's RSS anytime, whereas the latter provides the average RSS measured during the reception of the last packet.

[4]We display on the y-axis the relative RSS, which is computed as $s_n - E(\boldsymbol{R}_s)$, see Sec. II-C. We mark the $\boldsymbol{Z}(\boldsymbol{R}_s)$ points in red using $\rho = 0.2$.



(a) Captured RSS (b) Window 1 (c) Window 2

(d) Window 3 (e) Window 4 (f) Computed $\hat{T}$

Fig. 6: SF determination using an exemplary RSS capture of the first preamble symbols of a packet sent with SF = 8.

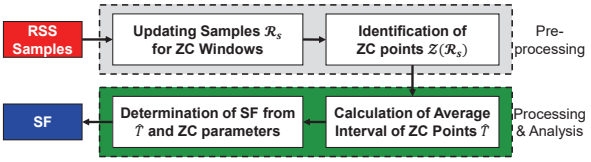


Fig. 7: Architecture of RSSF, which involves pre-processing of RSS samples and their analysis using a ZC algorithm.

preamble reception cannot be obtained, since L is too short. We hence propose to use six window size values L, one for each SF, as specified in Table I. We compute $\hat{T}$ using a sliding window with a step size $\Delta = 10$ RSS samples. RSSF will then look for M consecutive $\hat{T}$ with a constant value of $L/3$: when the condition is satisfied, the corresponding SF is returned, and the radio is configured accordingly to receive the packet.

D. Architecture

Fig. 7 illustrates the architecture of RSSF, which is employed in conjunction with Algorithm 1 for determining the SF. We assume that the transmitted SF is in the set $\mathcal{SF} = [7, 8, ..., 12]$, and test candidate SF values using the procedure explained in Sec. IV-C and the values listed in Table I. We denote the parameters of the candidate SF value being tested as L_{sf}, $\hat{T}_{sf}$, and M_{sf}. The set of RSS samples $\boldsymbol{S}$ is divided into a sliding window $\boldsymbol{R}_s$ of size L_{sf} (one for each candidate SF), which is updated whenever $\Delta = 10$ new RSS samples $\boldsymbol{s}$ are available. In order to ensure accurate SF determination, we also implement pre-processing of $\boldsymbol{R}_s$ with median filtering and removal of outliers. During the processing, the number of ZC points N_z can be determined by Eq. (3) and then $\hat{T}$ can be calculated by Eq. (4) for further analysis. If the measured $\hat{T}$ is constantly larger than $\hat{T}_{sf}$ (e.g., see in Fig. 6f how the $\hat{T}$ computed when SF = 9 is transmitted is higher than that when transmitting using SF = 8), we assume that the sought SF is larger than $\hat{T}_{sf}$, and mark the candidate SF with $Flag_l$ to indicate that smaller SFs should be excluded from subsequent attempts. In the presence of noisy RSS measurements, the computed $\hat{T}$ may be smaller than $\hat{T}_{sf}$, in which case RSSF may erroneously detect an SF that is too small. To avoid this, once RSSF finds that $M = M_{sf}$ (which indicates that the SF is possibly detected), it keeps observing the RSS for n_g additional windows. That is, it records the index of the current window n_d, mark $Flag_d$, and keeps observing the RSS for n_g additional windows. After n_g windows (i.e., at window n_w, where $n_w - n_d > n_g$), if no larger SF was detected (i.e., $Flag_l$ remained 0), RSSF deems that the SF marked by $Flag_d$ was the one used for packet transmission.

TABLE I: Zero-crossing parameters for BW = 125 kHz.

	SF					
	7	***8***	***9***	***10***	***11***	***12***
$\boldsymbol{L}$	33	65	129	258	516	1025
$\hat{\boldsymbol{T}}$	11	22	43	86	172	341
$\boldsymbol{M}$	5	13	15	8	10	10

Algorithm 1 SF Determination in RSSF

INPUT: $\boldsymbol{L} = [L_7, L_8, \ldots, L_{12}]$, $\hat{\boldsymbol{T}} = [\hat{T}_7, \hat{T}_8, \ldots, \hat{T}_{12}]$, $\boldsymbol{M} = [M_7, M_8, \ldots, M_{12}]$ in Table I, Δ and n_g, RSS samples $\boldsymbol{S} = [s_1, s_2, \ldots, s_N]$.
OUTPUT: sf.

Initialize $Flag_d = 0, Flag_l = 0$, the index of window $n_w = 0$, $\mathcal{SF} = [7, 8, ..., 12]$.
while 1 **do**
  Obtain Δ RSS samples in $\boldsymbol{s} = [s_1, \ldots, s_\Delta]$, $\boldsymbol{S} = \boldsymbol{S} + \boldsymbol{s} \triangleq [s_1, \ldots, s_N, s_{N+1}, \ldots, s_{N+\Delta}]$, $n_w = n_w + 1$;
  for sf in $\mathcal{SF}$ **do**
    $\boldsymbol{R}_s = [s_{N+\Delta-L_{sf}}, \ldots, s_{N+\Delta}]$;
    Compute N_z according to (3) by bring $\boldsymbol{R}_s$;
    Update $\hat{T} = L_{sf}/(N_z + 1)$;
    Update $Flag_l$ and M according to Sec. IV-C by bring $\hat{T}, L_{sf}, \hat{T}_{sf}$;
    if $Flag_l == 1$ **then**
      $\mathcal{SF} = [x \mid x \in \mathcal{SF}$ and $x > sf]$; break;
    else if $M == M_{sf}$ and $Flag_d == 0$ **then**
      $Flag_d = 1, n_d = n_w$;
    else if $n_w - n_d > n_g$ and $Flag_d == 1$ **then**
      return sf.
    end if
  end for
end while

V. Evaluation

We evaluate the performance of RSSF experimentally and describe our setup in Sec. V-A. After showing the SF detection accuracy as a function of the used BW (Sec. V-B) and SNR of received packets (Sec. V-C), we show the number of preamble up-chirps required by RSSF to determine the SF (Sec. V-D).

A. Experimental Setup

We implement RSSF in Python, and determine the SF based on traces of RSS samples collected using commercial LoRa devices [19]. Specifically, as in Sec. IV-B, we connect two SX1276 devices (acting as TX and RX) with an SMA cable, a fixed 40 dBm attenuator, as well as a 90 dBm programmable attenuator. We let the TX send packets at 0 dBm, and use the default preamble length of 8 symbols (i.e., 8 up-chirps will be transmitted). We then conduct experiments with SF = [7,12] and BW = [125, 250, 500] kHz while enforcing an attenuation between 100 dBm and 130 dBm to artificially lower the RSS of received packets. The TX channel frequency was set to 434 MHz, and the corresponding RX channel frequency was set to 434 MHz + BW/2. We transmit 100 packets for each configuration, and repeat each experiment three times.

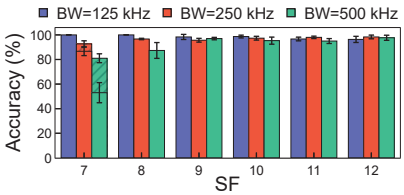


Fig. 8: Accuracy vs. bandwidth.

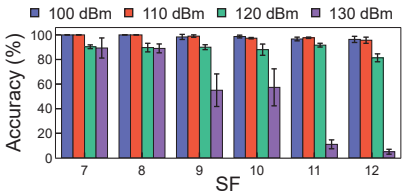


Fig. 9: Accuracy vs. SNR.

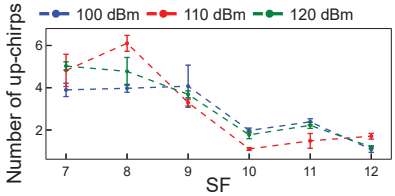


Fig. 10: No. of required up-chirps.

B. Accuracy for Different Bandwidths

Fig. 8 shows the accuracy of RSSF as a function of the employed BW. RSSF can accurately determine the SF with a minimum accuracy of 96.3% for all SFs when using BW = 125 kHz. However, as the bandwidth increases, the chirp duration becomes shorter, leading to a reduction in the number of RSS samples in each chirp. This makes it challenging for RSSF to obtain a constant $\hat{T}$, causing a slight drop in accuracy especially when using SF = 7. To tackle this limitation, one can increase the number of up-chirps sent within the preamble by a factor of two. This makes it easier to obtain a constant $\hat{T}$, and improves the accuracy of RSSF significantly, as demonstrated by the bars with a pattern fill for SF = 7.

C. Accuracy for Different Attenuations

Fig. 9 shows the accuracy of RSSF as a function of the attenuation level applied to the transmitted signal using the programmable attenuator (i.e., as a function of the SNR of received packets). For an attenuation of 100 and 110 dBm, RSSF can determine the SF with an average accuracy of 98.3 % and 98.2 %, respectively. As the attenuation increases, the performance of RSSF drops, especially when using an attenuation of 130 dBm (which is the noise floor of the receiver). In fact, as chirps sent with larger SFs have a longer airtime, it becomes increasingly difficult to recognize them due to additional noise in the RSS samples. Specifically, for SF=12, we obtain a detection accuracy of 96.3%, 95.7%, 81.3%, and 5% for an attenuation of 100, 110, 120, and 130 dBm, respectively.

D. Number of Required Up-chirps

We finally show the average number of up-chirps required by RSSF to determine the SF, which was calculated based on the number of RSS samples required to correctly calculate the SF. Fig. 10 shows that RSSF can determine the SF with an average of 3.9 up-chirps at an attenuation of 100 dBm. Additionally, the number of chirps required decreases as the SF increases. For instance, for SF = 12, it takes an average of 1.1 chirps at an attenuation of 100 dBm to correctly determine the SF.

VI. Conclusions

In this paper, we have presented RSSF, a scheme that can determine the SF with which a LoRa packet was sent without prior knowledge by simply inspecting the RSS of its first preamble symbols. In fact, RSSF leverages the characteristics of CSS modulation and RSS sampling on LoRa devices, and employs a lightweight zero-crossing algorithm with sliding windows to quickly determine the SF. An experimental evaluation shows RSSF's effectiveness in determining the SF with high accuracy even at low SNRs. In future work, we will optimize the zero-crossing algorithm to further improve the SF detection accuracy, and conduct additional experiments in mobile settings and highly-congested environments.

References

[1] M. Bor *et al.*, "Do LoRa Low-Power Wide-Area Networks Scale?," in *Proc. of the 19th MSWiM Conf.*, 2016.

[2] J. Petäjäjärvi *et al.*, "On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology," in *Proc. of the 14th ITST Conf.*, 2015.

[3] J. Liando *et al.*, "Known and Unknown Facts of LoRa: Experiences from a Large-scale Measurement Study," *ACM Transactions on Sensor Networks*, vol. 15, no. 2, 2019.

[4] M. Cattani *et al.*, "An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication," *JSAN*, vol. 6, no. 2, 2017.

[5] A. Farhad *et al.*, "Resource Allocation to Massive Internet of Things in LoRaWANs," *Sensors*, vol. 20, no. 9, 2020.

[6] D. Mu *et al.*, "Runtime Control of LoRa Spreading Factor for Campus Shuttle Monitoring," in *Proc. of the 28th ICNP Conf.*, 2020.

[7] A. Loubany *et al.*, "Adaptive Algorithm for Spreading Factor Selection in LoRaWAN Networks with Multiple Gateways," *Computer Networks*, vol. 182, 2020.

[8] H. Xie *et al.*, "Spreading Factor Allocation for Large-Scale Deployment In LoRaWAN Network," in *Proc. of the 5th ICCCS Conf.*, 2020.

[9] S. Li *et al.*, "How Agile is the Adaptive Data Rate Mechanism of LoRaWAN?," in *Proc. of the 37th GLOBECOM Conf.*, 2018.

[10] M. Bor *et al.*, "LoRa Transmission Parameter Selection," in *Proc. of the 13th DCOSS Conf.*, 2017.

[11] F. Cuomo *et al.*, "EXPLoRa: Extending the Performance of LoRa by Suitable SF Allocations," in *Proc. of the 13th WiMob Conf.*, 2017.

[12] B. Reynders *et al.*, "Power and Spreading Factor Control in Low Power Wide Area Networks," in *Proc. of the 2017 ICC Conf.*, 2017.

[13] M. Aslam *et al.*, "Exploring Multi-Hop LoRa for Green Smart Cities," *IEEE Network*, vol. 34, no. 2, 2020.

[14] G. Zhu *et al.*, "Improving the Capacity of a Mesh LoRa Network by SF-Based Network Clustering," *IEEE Access*, vol. 7, 2019.

[15] M. Bor *et al.*, "LoRa for the Internet of Things," in *Proc. of the 1st MadCom Workshop*, 2016.

[16] C.-H. Liao *et al.*, "Multi-Hop LoRa Networks Enabled by Concurrent Transmission," *IEEE Access*, vol. 5, 2017.

[17] A. Abrardo *et al.*, "A Multi-Hop LoRa Linear Sensor Network for the Monitoring of Underground Environments: The Case of the Medieval Aqueducts in Siena, Italy," *Sensors*, vol. 19, no. 2, 2019.

[18] F. Yang *et al.*, "Real-Time Decoding of LoRa Packets Without Prior Knowledge of their Spreading Factor," in *Proc. of the 19th EWSN Conf., demo session*, 2022.

[19] Semtech, "SX1276-7-8-9 Datasheet," 2020. [Online] https://sforce.co/3lWhC11 – Last access: 2023-04-15.

[20] S. Tong *et al.*, "CoLoRa: Enabling Multi-Packet Reception in LoRa," in *Proc. of the 39th INFOCOM Conf.*, 2020.

[21] H. K. Kwok *et al.*, "Improved Instantaneous Frequency Estimation Using an Adaptive Short-Time Fourier Transform," *IEEE transactions on signal processing*, vol. 48, no. 10, 2000.

[22] G. Burel *et al.*, "Blind Estimation of the Pseudo-Random Sequence of a Direct Sequence Spread Spectrum Signal," in *Proc. of the 21st MILCOM Conf.*, 2000.

[23] Y. Zheng *et al.*, "A CMOS VGA With DC Offset Cancellation for Direct-Conversion Receivers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 1, 2008.