# Demo: Real-Time Decoding of LoRa Packets Without Prior Knowledge of their Spreading Factor

Fengxu Yang[‡†], Pei Tian[‡‖¶], Xiaoyuan Ma[§], Jianming Wei[‡], and Carlo Alberto Boano[¶]

[‡]Shanghai Advanced Research Institute, Chinese Academy of Sciences, China;
[†]School of Information Science and Technology, ShanghaiTech University, China;
[‖]University of Chinese Academy of Sciences, China; [§]SKF Group, China;
[¶]Institute of Technical Informatics, Graz University of Technology, Austria

yangfx@shanghaitech.edu.cn {tianpei2018,wjm}@sari.ac.cn
ma.xiaoyuan.mail@gmail.com cboano@tugraz.at

## Abstract

In this demo, we propose `SFDec`, a scheme that allows off-the-shelf LoRa devices to receive a packet without prior knowledge of the spreading factor with which it was sent. `SFDec` stems from the observation that the spreading factor of a packet can be inferred by analyzing the received signal strength with which the first preamble symbols are received. After deriving the spreading factor, the radio can be configured to continue packet reception accordingly, without compromising a successful decoding of the packet. We showcase the feasibility of `SFDec` on real hardware, highlighting that the scheme does not require any hardware modifications and does not affect the operations of the original LoRa device.

## 1 Introduction

Low-power wide area networks (LPWANs) have been used to connect billions of energy-constrained devices over large geographical areas in the past decade [3]. LoRa is one of the most representative LPWAN technologies: its distinctive properties are the lack of subscription costs (users can deploy their own network infrastructure), the low current consumption of the radio allowing battery-powered operations for many years, as well as a high receiver sensitivity [5].

LoRa achieves a high receiver sensitivity thanks to the adoption of chirp spread spectrum (CSS) modulation. The carrier signal of LoRa consists in fact of *chirps*: sweep signals whose frequency increases (up-chirps) or decreases (down-chirps) over time across a specific frequency range. To address the needs of a wide range of applications, LoRa transceivers allow end-users to directly fine-tune a number of physical layer settings, such as the range of frequencies over which the chirps are spread (bandwidth), as well as the *spreading factor* (SF). The latter controls the chirp rate: LoRa spreads each symbol (i.e., bit of information) over $2^{SF}$ chips, with SF being typically a value between 7 and 12. Hence, the SF influences the data rate, the radio on-time, as well as the radio sensitivity and communication range [2].

The employed SF must be agreed in advance, as different spreading factors are orthogonal to each other [10]: for this reason, LoRa packets can only be received when the SF of the transmitter and of the receiver are identical. To ensure that this is the case, existing approaches either send the same information using different spreading factors (e.g., using a fixed SF at the receiver and a variable SF at the transmitter [4, 7]) or negotiate the SF explicitly (e.g., by means of handshaking mechanisms [6, 12]). These approaches, however, introduce extra overhead, and force a receiver to introduce various reception slots with a specific SF over time in case different transmitters make use of distinct SF values.

**Our contribution.** In this demo, we present `SFDec`, a scheme allowing an off-the-shelf LoRa device to receive a packet without prior knowledge of the SF with which it was sent, thereby removing the need of multiple sending attempts or explicit handshakes. Using `SFDec`, a LoRa receiver can detect an ongoing packet transmission and determine its SF by sampling the received signal strength (RSS) of the packet's first preamble symbols. Based on the characteristics of the measured RSS signal, `SFDec` infers the SF of the packet being received and configures the transceiver such that the remainder of the packet can be decoded successfully. We detail the principle behind `SFDec` in § 2 and show a preliminary evaluation of its feasibility on real hardware.

## 2 `SFDec`: Principle and Feasibility

**Working principle.** `SFDec` stems from the observation that packets sent with a different SF lead to a different RSS footprint at the receiver, which can be characterized while receiving the packet's preamble[1]. We illustrate this observation by experimentally measuring the RSS while receiving LoRa packets using two ChirpBox nodes[2] placed at a distance of 30 cm, using a transmission power of 14 dBm, and employing a bandwidth (BW) of 125 kHz. One of the nodes periodically transmits data using a different SF, whereas the other one records the RSS with a sampling frequency of 28.5 kHz.

[1]A LoRa packet consists of a preamble followed by an explicit header, a payload, and a 2-byte Cyclic Redundancy Check computed based on the payload content. The length of the preamble (made of multiple identical up-chirps) is programmable and is typically set to 8 symbols by default.

[2]ChirpBox nodes are equipped with an SX1276 LoRa transceiver operating at 433 MHz and an STM32L476RG micro-controller [11].

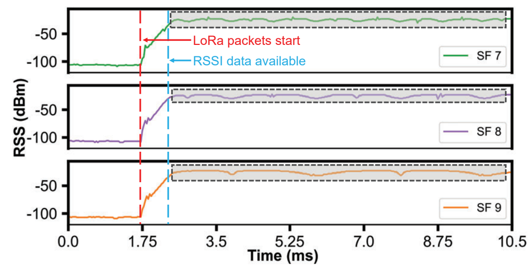


**Figure 1. RSS values recorded by a LoRa device when receiving the preamble of packets sent using different SFs.**

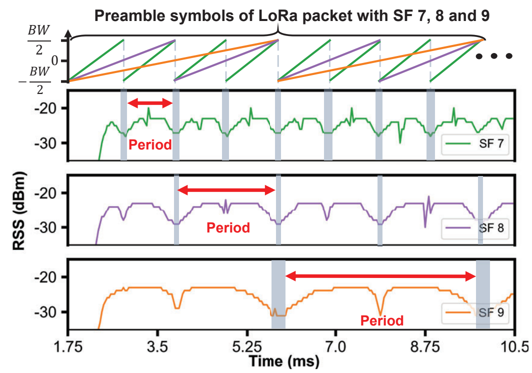


**Figure 2. Detailed RSS evolution after reaching a steady value. Depending on the employed SF, one can observe a different periodicity and amplitude of the RSS profile.**

Fig. 1 illustrates the RSS traces measured by the receiver for different SFs. The packet transmission was initiated at t=1.75 ms, after which the RSS value slowly increases from -110 dBm to roughly -20 dBm in about 1 ms, regardless of the employed SF. This behaviour resembles the one of other low-power wireless radios and is due to the internal averaging of the RSS over time [1]. After reaching a steady value, the RSS fluctuates in the range (-30, -20) dBm in a periodic fashion. Fig. 2 shows a closer view of how the RSS evolves over time (once it reached a steady value) while receiving preamble symbols sent with different SF. One can observe that the RSS fluctuates with a greater period when the packet is sent with a higher SF (and the same trend shown in Fig. 2 for SF 7 to 9 applies to higher SFs). This is because the RSS values drop as soon as the transmission of an up-chirp begins due to the sudden and large change in frequency (from BW/2 to -BW/2). Therefore, one can identify the periodicity of the RSS drops to derive the SF with which the packet's preamble was sent. Exploiting this information, one can then configure the SF accordingly and continue the packet reception: packets can still be decoded successfully, because only 1.9 symbols are typically sufficient to detect a LoRa packet [10].

**Implementation.** We design `SFDec` to derive the SF after analyzing the RSS of 4 preamble symbols, and create a prototypic implementation running on ChirpBox nodes as follows.

*Step 1: data filtering & peak extraction.* `SFDec` waits for the RSS values to reach a steady value above a pre-defined threshold, and then smooths the values using a Savitzky-Golay filter [9]. A difference algorithm [8] is then applied to the smoothed RSS values to identify the RSS drops, and the indexes of these drops within the RSS sequence are extracted.

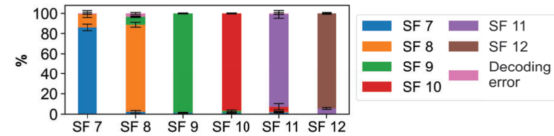


**Figure 3. Accuracy of `SFDec` for different SFs.**

*Step 2: SF computation and packet reception.* Based on the drops in RSS and their indexes, the period is calculated and the SF inferred accordingly. `SFDec` then switches the radio to the calculated SF in order to receive and decode the packet.

**Preliminary evaluation.** We evaluate the ability of `SFDec` to correctly detect and decode arbitrary LoRa packets sent with different SFs as follows. We let the ChirpBox node acting as a transmitter send 100 packets for each SF (7 to 12), and let the receiver log for each packet which SF was detected and the decoded packet. We set the BW to 125 kHz, the coding rate to 4/5, and the payload size of the packets to 10 bytes. We repeat each experiment 3 times. Fig. 3 shows our results: in average, `SFDec` correctly detects the SF using the first four preamble symbols and decodes the packet in 92.27% of the cases. The best performance is achieved with SF 9 (98.7% of the packets correctly decoded). The worst performance is obtained with SF 7 (86.0% of the packets correctly decoded and 13.3% of the packets erroneously detected as SF 8). Such a mis-recognition is caused by the Savitzky-Golay filter, which cannot remove interference effectively and leads to errors in the peak extraction.

## 3 Conclusions and Future Work

Our preliminary results show the feasibility of the proposed approach. In the future, we plan to refine the data filtering and peak extraction algorithms to further improve `SFDec`'s performance, especially in the presence of low signal-to-noise ratios and external interference. We will also study the impact of node mobility on `SFDec`'s performance.

**Acknowledgments.** This work was funded by the STCSM (19DZ1202200) and was also supported by the TU Graz LEAD project "Dependable IoT in Adverse Environments".

## 4 References


[1] C. Boano et al. JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation. In *Proc. of IPSN'11*.

[2] M. Cattani et al. An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Comm. *JSAN*, 6(2), 2017.

[3] Cisco. Cisco Annual Internet Report (2018–2023) White Paper, 2020. [Online] https://bit.ly/3aH0wkZ – Last access: 2022-07-21.

[4] A. Farhad et al. Resource Allocation to Massive Internet of Things in LoRaWANs. *Sensors*, 20(9), 2020.

[5] B. Foubert et al. Long-Range Wireless Radio Technologies: A Survey. *Future Internet*, 12(1), 2020.

[6] A. Loubany et al. Adaptive Algorithm for Spreading Factor Selection in LoRaWAN Networks with Multiple Gateways. *Comp. Net.*, 2020.

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

[8] G. Palshikar et al. Simple Algorithms for Peak Detection in Time-series. In *Proc. of the 1st ICADABAI Conf.*, 2009.

[9] A. Savitzky et al. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8), 1964.

[10] Semtech. SX1268 Datasheet, 2019. [Online] https://sforce.co/3pnhGJg – Last access: 2022-07-21.

[11] P. Tian et al. ChirpBox: An Infrastructure-Less LoRa Testbed. In *Proc. of the 18th EWSN Conf.*, 2021.

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