Dynamic LoRa PHY layer for MAC experimentation using FIT/CorteXlab testbed

Amaury Paris, Leonardo S. Cardoso, Jean-Marie Gorce
Univ Lyon, INSA Lyon, Inria, CITI
amaury.paris@insa-lyon.fr

Abstract

In this paper we present a complete GNU Radio, dynamic and customizable physical (PHY) layer for long range (LoRa) transceiver, usable with the FIT/CorteXlab radio testbed and derived from the original EPFL LoRa implementation. The created adaptation, through a standardized interface, allows end-users an easy connection to an external medium access control (MAC)/upper layer to experiment scenarios in a fully reproducible and isolated environment. It also provides several PHY layer key performance indicators and metrics such as signal to noise ratio (SNR), received signal energy, binary error rate (BER) and other, that can be used to gauge the performance of the ongoing communications as well as construct MAC layers able to use this information.

Finally, the interface allows our plug&play PHY solution to be used with any existing or newly adapted MAC layer, without having to implement it in GNU Radio. This will be demonstrated during our presentation.

Index Terms *- IoT, FIT/CorteXlab, LoRa, PHY, MAC, experimentation, Software Defined Radio*

1. Introduction

Wireless communication and other research fields rely on experimentation platform to validate or build new scientific models. Creating such platforms is time consuming because of constraints, specific from one topic to another. In PHY or MAC experimentation within the Internet of things (IoT), and in general with wireless communication, we can face three issues: synchronisation (frequency, time and frame), reproducibility and real-time control.

The FIT/CorteXlab platform, whom offers key features to aid in the development and test of IoT as well as any other wireless network, already provides solutions for the first two issues. The last-one, real-time control, would allow the user to play with the PHY layer variables as the experimentation evolves, while receiving feed-back that allows decision making and adaptation. The work herein addresses this limitation.

2. FIT/CorteXlab radio testbed

The FIT/CorteXlab [1] is a radio testbed built with a substantial set of software defined radio (SDR) nodes. It allows users to set up large wireless network for realistic scenarios testing. All SDR nodes offers a wide flexibility with the possibility to transmit at any frequency, bandwidth, power and implement any modulation as desired. It is located in a shielded room, isolated from any outside interference, making the transmission channel, and therefore any experiment, reproducible. All SDR nodes have the possibility to share a common clock to help with frequency and time synchronisation and/or to use a time-slot transmission synchronisation framework [2] that helps with frame synchronisation issue in slotted systems.

The platform is accessible remotely through a service interface and open to anyone [3] over the Internet.

3. Design architecture

3.1. Overview

The EPFL implementation of the LoRa PHY layer [4] is adapted for the FIT/CorteXlab testbed. Also, it has been verified and confirmed that the implementation is able to communicate with certified LoRa devices.

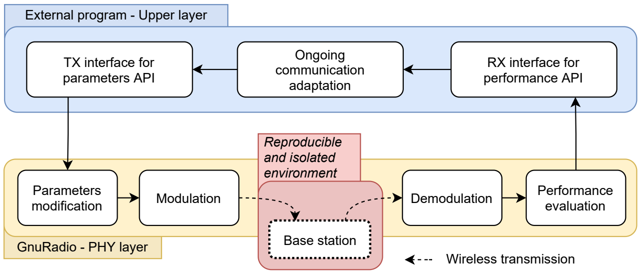
External program - Upper layer
TX interface for parameters API
Ongoing communication adaptation
RX interface for performance API
Reproducible and isolated environment
Parameters modification
Modulation
Base station
Demodulation
Performance evaluation
GnuRadio - PHY layer
Wireless transmission

Figure 1. System architecture

Our goal being the development of MAC/upper layers for LoRa wide area network (LORAWAN), we need to interface this PHY layer with the product of our research in GNU Radio. This is not convenient as, not only it would require any user to learn GNU Radio before testing any MAC or upper layers, but also GNU Radio implementations of upper layers requires expert usage of GNU Radio's toolbox. Both can be prohibitively expensive in terms of time and expertise for users simply wanting to experiment with MAC and higher layers onto a flexible LoRa PHY. Therefore a standardized two-way API interface for GNU Radio has been developed to receive and return information from/to any external program (tested in Python and C++). The API is used for the TX chain to set transmission parameters and for the RX chain to return key performances indicator, as presented in Fig.1, allowing a decision-making external algorithm (or upper layers) to control the PHY layer with real-time evaluation and evolution.

3.2. Transmitter

Key parameters to set on the GNU Radio TX PHY chain, as described in Table 1, are transmitted through a user datagram protocol (UDP) connection with JSON format {*"parameter":"new value"*}. They can be applied onto the chain in two different ways:

- **Tags:** propagate new parameter value through all TX blocks and update it;
- **Top block set/get functions:** update new parameter value for one block only.

Parameter		Application	Value
Spreading Factor	SF	Tags	[7-12]
Coding Rate	CR	Tags	[0-4]
TX Gain	GTX	Set function	*See SDR*[1]
RX Gain	GRX	Set function	*See SDR*[1]
TX Frequency	FTX	Set function	*See SDR*[1]
RX Frequency	FRX	Set function	*See SDR*[1]
Message	MSG	TX Chain	ASCII

Table 1. Settable TX parameters

All parameters are applied upon reception in GNU Radio, so do messages that are immediately modulated and transmitted.

3.3. Receiver

On the receiver side, key performance indicator are computed within the reception chain and gathered to be returned, in the same way as for the TX interface, to the external program/upper layer implementation.

Here is the list of all returned indicators, by category, for each received and demodulated packet:

- **Header:** Coding rate, payload length;
- **Demodulation:** Number of corrected bit, number of erroneous bit, spreading factor, content (message), cyclic redundancy check (CRC) verification status;
- **Energy:** Energy of the received packet, average energy of noise prior to the received packet, SNR of the received packet, average of packet energy and SNR over the last received packet.

Noise samples are buffered to compute SNR when the packet demodulation is over and to avoid any delay, as described in Fig. 2.

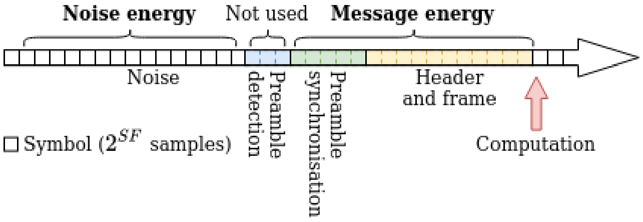

Figure 2. Energy computation for one message energy and SNR

All energy indicators are computed in decibel (dB). Also, improperly demodulated messages received in the RX chain, identifiable with the "false" CRC verification status, are also returned through the RX interface for performance evaluation. All these indicators can be used by a MAC/upper layer to adapt its transmission and set different TX parameters.

4. Demonstration

The FIT/CorteXlab platform is opened to all, after requesting an account and following several tutorials available on the CorteXlab Wiki page [3]. All is required is an Internet connection and a Unix shell.

A demonstration of the presented implementation is available on the tutorial page as "LoRa dynamic PHY layer". The proposed scenario, with a node to transmit messages upon request and a base station to acknowledge the transmission, act as a basic example and is made to visualize the project layout.

References

[1] Abdelbassat Massouri et al. *CorteXlab: An open FPGA-based facility for testing SDR & cognitive radio networks in a reproducible environment*. IEEE, 2014.

[2] O. Oubejja et al. "Framework for PHY-MAC layers Prototyping in Dense IoT Networks using FIT/CorteXlab Testbed". In: *Univ Lyon and INRIA CITI* (2019).

[3] INRIA CITI. *FIT/CorteXlab general information, access and usage*. URL: `https://wiki.cortexlab.fr/doku.php`.

[4] Joachim Tapparel et al. *An Open-Source LoRa Physical Layer Prototype on GNU Radio*. 2020. arXiv: `2002.08208 [eess.SP]`.

[1] Value depend on SDR, for USRP B210 frequency range is 70 MHz - 6 GHz and gain range is 0 - 89.6 dB (high gain values are not recommended because of distortion)