Demo: A Cell-level Traffic Generator for LoRa Networks

Michele Gucciardo
University of Palermo
Palermo, Italy
michele.gucciardo@unipa.it

Ilenia Tinnirello
University of Palermo
Palermo, Italy
ilenia.tinnirello@unipa.it

Domenico Garlisi
University of Palermo
Palermo, Italy
domenico.garlisi@unipa.it

ABSTRACT

In this demo we present and validate a LoRa cell traffic generator, able to emulate the behavior of thousands of low-rate sensor nodes deployed in the same cell, by using a single Software Defined Radio (SDR) platform. Differently from traditional generators, whose goal is creating packet flows which emulate specific applications and protocols, our focus is generating a combined radio signal, as seen by a gateway, given by the super-position of the signals transmitted by multiple sensors simultaneously active on the same channel. We argue that such a generator can be of interest for testing different network planning solutions for LoRa networks.

1 INTRODUCTION

In recent years, we have assisted to an impressive proliferation of wireless technologies and mobile-generated traffic, which is now the highest portion of the total internet traffic and will continue to grow with the emergence of IoT applications [1]. Understanding how wireless networks would respond to the increasing capacity demand is therefore of crucial importance. High-density deployments of wireless networks are difficult to study in real testbeds or installations, especially when we can expect that hundreds or even thousands of mobile devices (not limited to traditional user terminals, but including smart objects and sensors) are simultaneously active in the same environment.

In this demo we consider the case of LoRa technology, which represents a critical example of wireless technology working with high-density devices. Indeed, LoRa technology has been conceived for Low Power Wide Area Networks (LPWAN), characterized by low data rate requirements per single devices, large cells and heterogeneous application domains, which may lead to extremely high numbers of devices coexisting in the same cell. Despite of the robustness of the LoRa PHY [2], patented by Semtech, in WAN scenarios where multiple gateways can be installed, the scalability of this technology is still under investigation [3]. Current studies are mostly based on simulation results [4] and assume that the utilization of multiple transmission channels and spreading factors lead to a system that can be considered as the simple super-position of independent (single channel, single spreading factor) sub-systems. This is actually a strong simplification, especially because the spreading factors adopted by LoRa are pseudo-orthogonal [5] and therefore, in near-far conditions, collisions can prevent the correct reception of both the overlapping transmissions using different spreading factors.

For characterizing these phenomena on real gateways, we designed and implemented a cell traffic generator for LoRa-based networks on top of the well known USRP software-defined-radio (SDR) platform. While traditional traffic generators are limited to the generation of packet flows emulating specific applications and protocols [6, 7], our focus is on the generation of a *combined* radio signal given by the super-position of the signals, generated by multiple devices operating in the same cell, as seen by a gateway.

Similar approaches have been recently proposed for studying high-density WiFi networks [8], resulting from installations in a stadium, a conference hall, an airport, etc. Commercial solutions try to emulate multiple devices by means of a special contention scheduler working on a single radio interface [9, 10] or by embedding multiple radio interfaces in the same traffic generator [11]. SDR-based solutions, summing overlapping signals in software before generating the resulting aggregated signal on the air, have also been explored in [12] for studying high-density cognitive networks.

Also our solution is based on SDR and software-based combination and scheduling of signals to be transmitted in real-time, according to a desired traffic and cell scenario. To this purpose, we modeled LoRa modulation (based on chirp spread spectrum) and LoRa access mechanism (based on aloha and duty cycle enforcement). Simple models for traffic sources and physical channels are also included. We also provide a software receiver, able to process the trace produced by the traffic generator and evaluate low-level statistics to be compared with the statistics of a real gateway.

2 LORA TRANSCEIVER IN SDR

An important component of our Traffic Generator is obviously the software-defined implementation of a LoRa transceiver. Indeed, this block is essential for producing elementary LoRa-modulated signals, as a function of the traffic source schedules, which are then translated on time and attenuated according to the physical channel model of each node.

LoRa modulation. It is derived from *Chirp Spread Spectrum (CSS)*, which makes use of *chirp signals*, i.e. frequency-modulated signals obtained when the modulating signal varies linearly in the range $[f_0, f_1]$ (upchirp) or $[f_1, f_0]$ (downchirp) in a symbol time T. Binary modulations, mapping 0/1 information bits in upchirps/downchirps, have been demonstrated to be very robust against in-band or out-band interference [2]. LoRa employes a M-ary modulation scheme based on chirps, in which symbols are obtained by considering different circular shifts of the basic upchirp signal. The temporal shifts, characterizing each symbol, are slotted into multiples of time

---

MobiCom '17, October 16–20, 2017, Snowbird, UT, USA

ACM ISBN 978-1-4503-4916-1/17/10.
https://doi.org/10.1145/3117811.3119862

$T_{chip} = 1/BW$, called chip, being $BW = f_1 - f_0$ the bandwidth of the signal. It results that the modulating signal for a generic n-th LoRa symbol can be expressed as:

$$f(t) = \begin{cases} f_1 + k \cdot (t - n \cdot T_{chip}) & \text{for } 0 \leqslant t \leqslant n \cdot T_{chip} \\ f_0 + k \cdot (t - n \cdot T_{chip}) & \text{for } n \cdot T_{chip} < t \leqslant T \end{cases}$$

where $k = (f_1 - f_0)/T$ is the slope of the frequency variations. The total number of symbols (coding SF information bits) is chosen equal to 2^{SF}, where SF is called spreading factor. The symbol duration T required for representing any possible shift is $2^{SF} \cdot T_{chip} = 2^{SF}/BW$. It follows that, for a fixed bandwidth, the symbol period increases with SF, enlarging the temporal occupancy of the signal. Figure 1 shows the modulating signal used for a basic upchirp and three examples of circular shifts obtained for $SF = 9$: the symbol time is $512\ T_{chip}$, while the three examplary shifts code the symbols 128, 256 and 384. The preamble of any LoRa frame is obtained by sending a sequence of at least eight upchirps, followed by two coded upchirps, used for network identification (sync word), and two and a quarter base downchirps. Payload data are then sent by using the M-ary modulation symbols.

LoRa provides three BW settings (125, 250 or 500 KHz) and seven different SF values (from 6 to 12). In general, a larger bandwidth translates in a data rate increase and a receiver sensitivity deterioration. Conversely, higher spreading factors can be used for improving the link robustness at the cost of a lower data rate. The computation of the signal samples has been implemented in MATLAB and sent to the DAC of the USRP board, by also considering the preamble format of LoRa PHY. No channel coding has been currently considered.

LoRa demodulation. An interesting feature of LoRa modulation is the pseudo-orthogonality of signals modulated under different spreading factors. This feature allows the receiver to easily separate multiple concurrent transmissions because of the cross-energy between symbols modulated at different spreading factors is almost zero. In general, a single-user receiver multiplies the received signal with the base downchirp. This leads to a signal with only two frequencies: $f_n = -k \cdot n \cdot T_{chip}$ and $f_n - BW = -(f_1 - f_0) - k \cdot n \cdot Tchip$. They will be both aliased to the same frequency f_n by downsampling at rate BW. The position of the peak at the output of an iFFT will correspond to the estimated symbol $\hat{n}$, as described in [13] and [14]. An example output of this workflow is shown in figure 4, when a reference signal at $SF = 9$ is overlapped to an interferer at $SF = 8$ with equal power. A clear peak reveals the reference symbol, whereas the interfering signal results in a wide-band spectrum with low spectral density. We exploited these considerations for implementing a LoRa demodulator in MATLAB, working on the signal samples generated by the modulator.

3 DEMONSTRATION DESCRIPTION

Goal of our demo is analyzing the cell-level performance of a LoRa network under different network plannings, i.e. under different allocations of per-node spreading factors and transmission powers. Indeed, rather than working with random spreading factors and uniform transmission powers, more robust configurations can be reserved for users experiencing bad channel qualities. We assume that all the nodes emulated by our traffic generator are configured for working on the same channel, while the number of nodes and their distribution within the cell can be configured in each experiment.

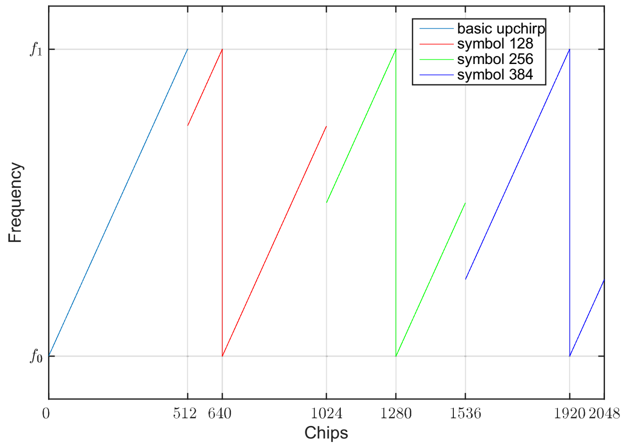

Figure 1: Modulating signal with $SF = 9$ for one basic upchirp and three symbols: 128, 256 and 384.

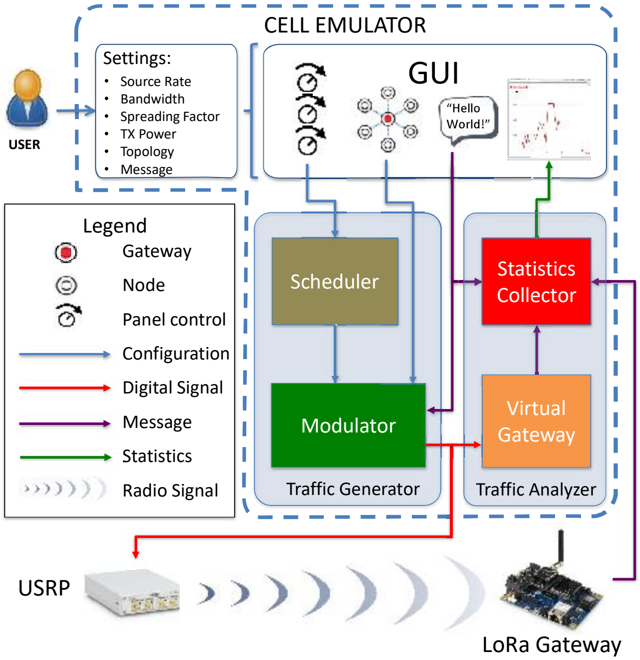

Figure 2: LoRa cell emulator architecture.

Figure 2 summarizes the software architecture of the LoRa cell emulator and the hardware used for the demo set-up: a real *LoRa Gateway*, implemented by using a Raspberry Pi 3 piloting the iC880A gateway transceiver, receives over-the-air the signal produced through a *USRP B210*, which results from the super-position of multiple LoRa-modulated signals generated by the cell nodes simultaneously active. Two additional monitors (HDMI), keyboards and mouses, not shown in the figure, will be utilized for visualizing the configuration interface and results.

The LoRa cell emulator is composed of three main software modules: (a) a *Traffic Generator*, (b) a *Traffic Analyzer* and (c) a *Graphical*

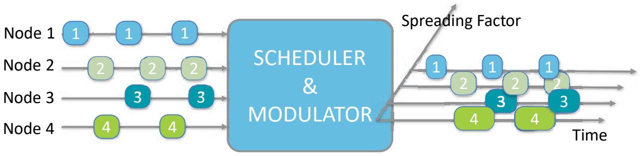

Figure 3: An example of per-node LoRa modulated signals, which are scheduled on time before being combined into the cell-level aggregated signal.

User Interface (GUI). The first two modules are written in MATLAB, whereas the GUI is Python based. The traffic generator, in turn, is composed of: a *Scheduler*, which is responsible of generating the activity intervals of each node according to the source rate and duty-cycle constraints; a *Modulator*, which receives the schedule of the nodes involved in transmission and generates the cell-level aggregated signal by summing up the LoRa modulated signals of the nodes simultaneously active, as shown in figure 3. Before combing the signal generated by each node into the cell-level signal, a path-loss attenuation model is applied by scaling and delaying the relevant chirps.

The digital signal is sent to both the USRP, for being transmitted over the air, and the Traffic Analyzer. The Traffic Analyzer is composed of a *Virtual Gateway* and a *Statistics Collector*. The former implements a LoRa demodulator working in parallel on multiple spreading factors, for identifying message preambles and then correlating each message with the corresponding basic upchirp. The latter compares the symbols demodulated in each message with the ones sent by the nodes, in order to produce low-level error statistics. The signal transmitted over the air is demodulated also by a real gateway, whose packet error statistics are sent to the statistic collector module for comparison with the results obtained by the virtual gateway.

The cell configuration, specified by means of the GUI, includes the tuning of the following parameters: number of nodes, nodes distribution (uniform, uniform in a given area), source rates, bandwidth, spreading factor allocation (random, distance-based), transmission power (uniform, distance-based). The parameters can be differentiated for groups of nodes, thus facilitating the planning for large cells.

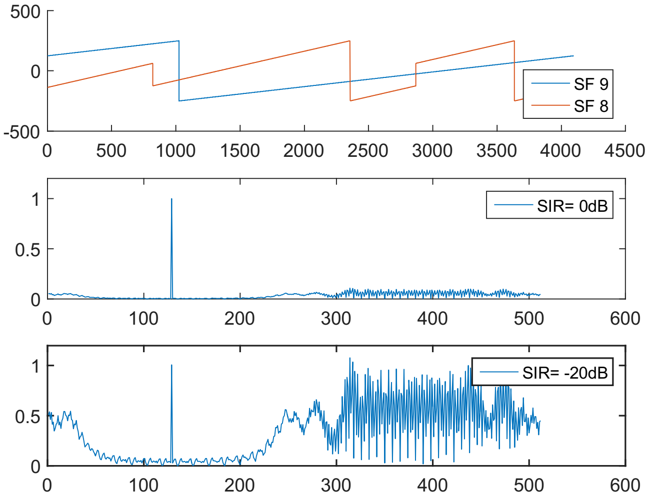

Figure 4: An example of collision between signals modulated with different SF. A LoRa symbol with SF equal to 9 (solid line) and two overlapping interfering symbols with SF equal to 8 (dashed line) received at different SIR levels, and iFFT output after multiplication with the base downchirp and downsampling.

Figure 4 shows an example of digitally overlapped signal generated by two nodes using different spreading factor: one signal modulated with SF equal to 9 (blue line) is overlapped to two symbols modulated with SF equal to 8 (red line). The figure also shows the output of the iFFT, after multiplication with the base downchirp at $SF = 9$ and downsampling, in two different scenarios: the two nodes are equally distant from the gateway and employ uniform transmission powers (middle figure); the distances of the two nodes are heterogeneous and results in one signal power 20dB greater than the other one (bottom figure). In this second case, although the two signals are in principle orthogonal, the interference between the two nodes can prevent the correct demodulation of the symbol modulated with $SF = 9$, that we assume is received at lower power (being higher spreading factors usually allocated to farer users). In fact, correlation peaks can occur in positions different from 128, as evident from the figure in the interval $[300 - 450]$.

4 ACKNOWLEDGEMENTS

This work has been partially supported by EU funded research project symbIoTe, H2020-ICT-2015 grant agreement number 688156.

REFERENCES

[1] Worldwide connected devices forecast www.statista.com
[2] Semtech. LoRa Modulation Basics. AN1200.22, Revision 2, May 2015. www.semtech.com
[3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?", In Proc. of MSWiM 2016. ACM, New York, pp. 59-67.
[4] B. Reynders and S. Pollin, "Chirp spread spectrum as a modulation technique for long range communication", In SCVT 2016, Mons, pp. 1-5.
[5] B. Reynders, W. Meert and S. Pollin, "Range and coexistence analysis of long range unlicensed communication", In ICT 2016, Thessaloniki, pp. 1-6.
[6] L. Kelch, T. P¡gel, L. Wolf, A. Sasse, "Traffic Generator for HSDPA Network Simulations", In ICCVE 2014, Vienna, 25-330.
[7] D. Kaulbars, F. Schweikowski, C. Wietfeld, "Spatially Distributed Traffic Generation for Stress Testing the Robustness of Mission-Critical Smart Grid Communication", In Proc. of GLOBECOM Workshops 2015, San Diego.
[8] G. A lvarez *et al.*, "Multi-client Emulation Platform for Performance Testing of High Density 802.11 Networks", In Wireless Days (WD) 2016, 1-6.
[9] CT520 LANforge WiFIRE 802.11a/b/g/n WiFi Traffic Generator. www.candelatech.com
[10] Multi-Client Traffic Generator/Performance Analyzer www.veriwave.com
[11] TM500 Network Tester www.cobhamwireless.com
[12] S. Papadakis, A. Makrogiannakis, M. Surligas, P. Papadakis, Single SDR Multiple Primary & Secondary Users Emulation Platform, in EuCNC 2015, Paris.
[13] O. Bernard, A. Seller, N. Sornin, "Low power long range transmitter", European Patent Application EP 2763321 A1 by Semtech Corporation, 2014
[14] C. Goursaud, J.M. Gorce, "Dedicated networks for IoT: PHY / MAC state of the art and challenges", in EAI endorsed transactions on Internet of Things, 2015.