LoRa®

LoRa 2.4 GHz: a methodology to study coexistence in the 2.4 GHz ISM band

Journées LPWAN, Toulouse, July 2022

Gwendoline HOCHET DEREVIANCKINE | gwendoline.hochet-derevianckine@inria.fr

With Alexandre Guitton (UCA) | Oana Iova (INSA Lyon) | Baozhu Ning (Semtech) | Fabrice Valois (INSA Lyon)

Inria | INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON | SEMTECH | UCA UNIVERSITÉ Clermont Auvergne

Agenda

1. LoRa Background
2. Introduction to LoRa 2.4 GHz
3. A methodology for coexistence study
4. Conclusions and perspectives

1

LoRa Background

LoRa®

- LPWAN technology — Aims to collect data on large coverage
- Based on PHY configurations — CSS modulation + combination of Spreading Factor (SF) / Bandwidth (BW) / Coding Rate (CR)
- LoRaWAN is on top of LoRa — Defines network architecture + MAC layer
- Works on ISM frequency bands — Europe: 433 and 868 MHz; North America: 915 MHz; China: 470 MHz

LoRa 2.4 GHz: Advantages

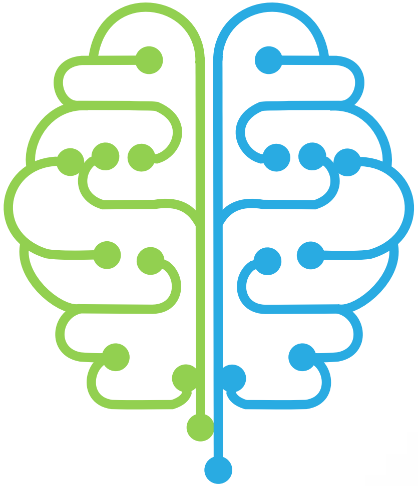
LoRa sub-GHz limitations

Frequency bands are country-dependent → one hardware per region

Duty cycle between 0.1 and 10% depending on the frequency used

Low data rate (up to 12.5 kb/s) due to frequency band

Localization accuracy from 20 to 200 meters



LoRa 2.4 GHz solutions

All regional regulations for the 2.4 GHz ISM band have large intersections → design of a single hardware

No duty cycle constraints

Allow higher data rate (up to 253.91 kb/s) → reduces latency and offers real-time applications opportunities

Improved localization accuracy (about 2 meters)

PHY Layer configurations in 2.4 GHz: what is new?

	LoRa sub-GHz	LoRa 2.4 GHz
Spreading Factor	**7-12**	**5**, 6-12
Bandwidth (in kHz)	**125**, 250, **500**	**203**, 406, 812, **1625**
Coding Rate	**4/5**, 4/6, 4/7, **4/8**	
Data Rate (in kb/s)	**0.98** to **12.5** (uplink) **0.98** to **21.9** (downlink)	**0.595** to **253.91**
MAC Standardization	**LoRaWAN**	**/**

Channel characterization: the 2.4 GHz ISM band

The 2.4 GHz ISM band is widely used by wireless technologies

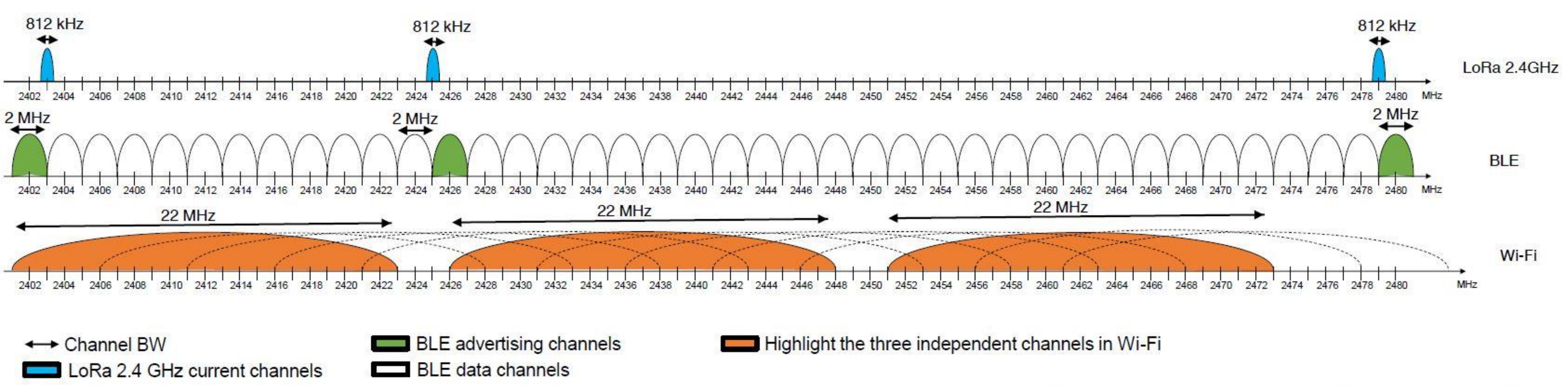


Figure 1 – Frequency channels and bandwidth of LoRa 2.4 GHz, BLE, and WiFi in the 2.4 GHz ISM band

3

Channel characterization: motivations

3 main categories in the state of the art:

- Spectrum occupancy [1]
- Coexistence:
  - **Without concurrent communications of others wireless technologies [2]**
  - **Interference mitigation/detection [3]**
- Cross-technology communication [4] [5]

Objective: being able to characterize the environment in order to perform experiments in a known environment and observe the impact of a given technology on the performance of another technology and vice versa

3

Channel characterization : steps

1) Spectrum occupancy at a given time ⇔ screenshot of the channel

2) Technology identification (**this talk**)

3) Percentage of channel occupancy per technology

Channel characterization: setup

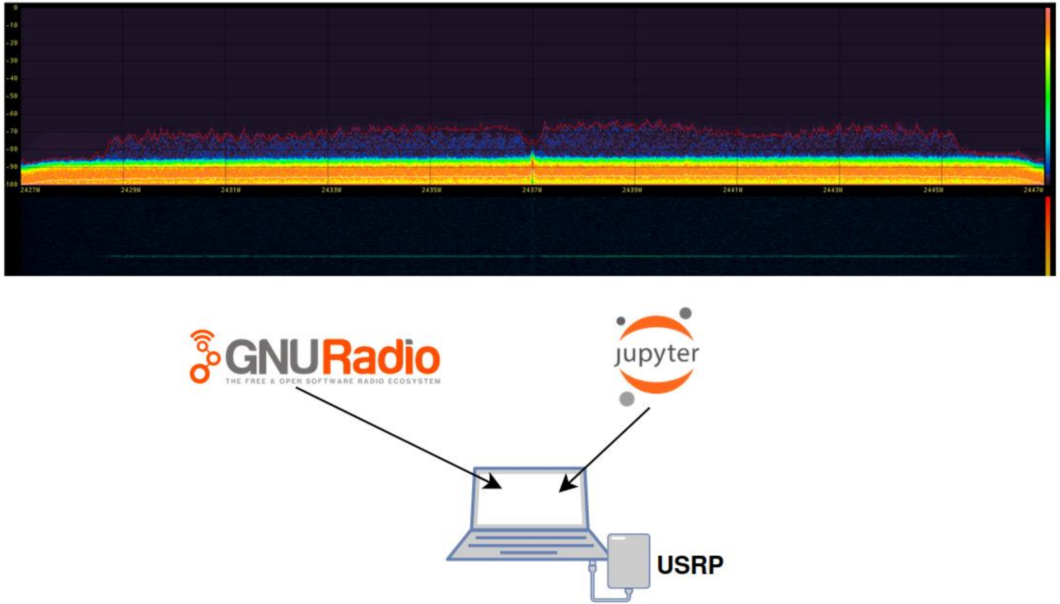


Figure 2 – Setup for capturing and processing signals in the 2.4 GHz ISM band

- **Capturing signals in the 2.4 GHz ISM band:**
  - 1 USRP connected to a laptop for sensing the environment
  - GnuRadio software for signal visualization and recorded
- **Post-processing data:**
  - Binary files containing complex float data
  - Python programming
  - JupyterLab software for signal processing
- **Parameters:**
  - Capture duration approximately 1 minute per file
  - BW = 4 MHz
  - Split the record BW into 20 sub channels ➔ 1 sub channel = 200 kHz

3

Channel characterization: is it noise or signal?

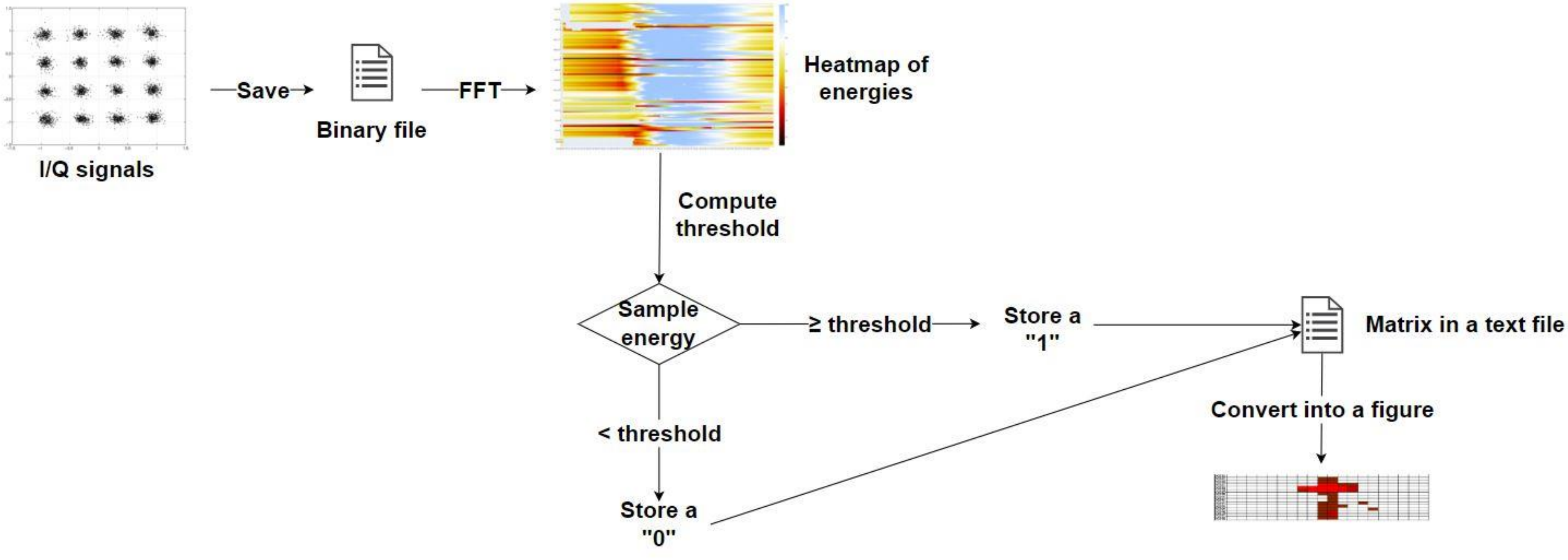
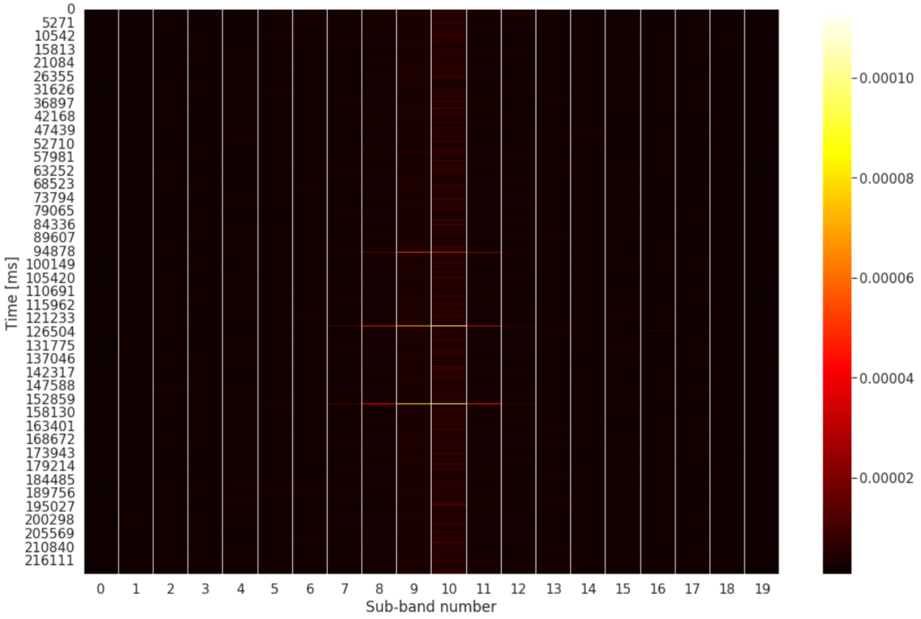
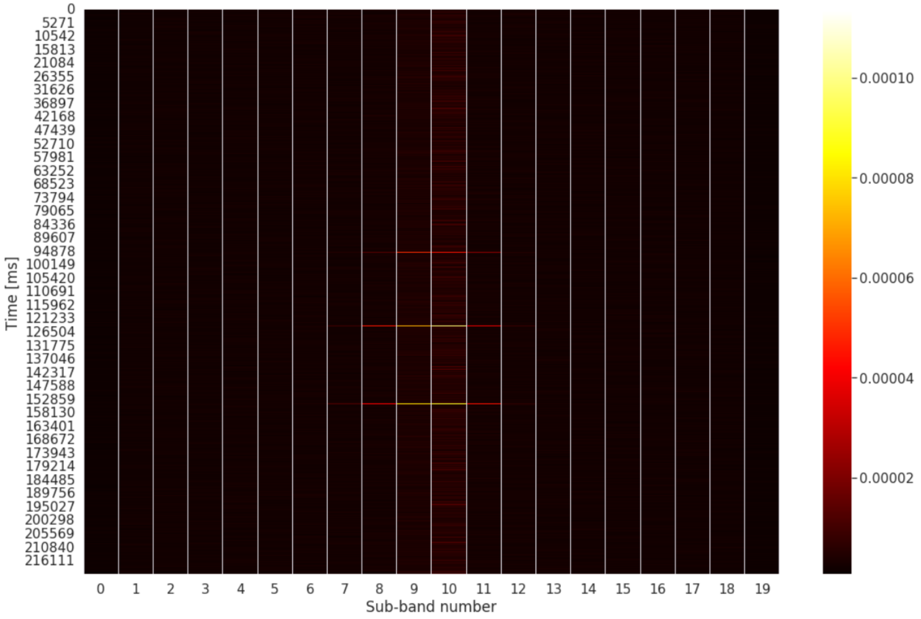


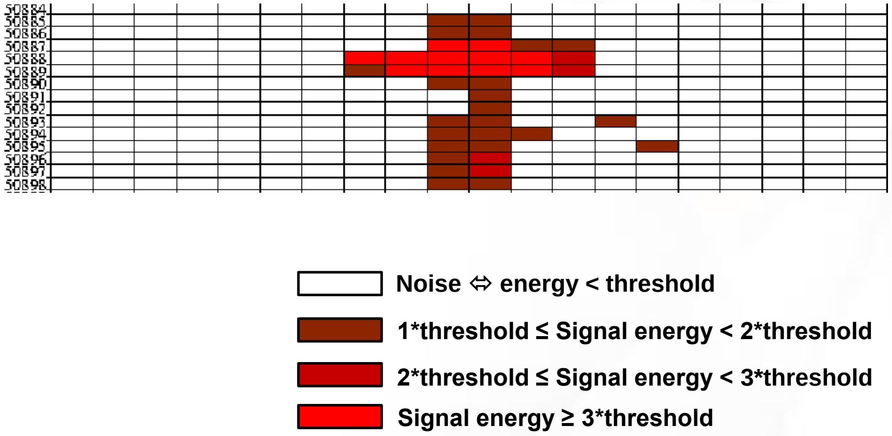
Figure 3 – Diagram of the steps for signal processing

3 Channel characterization: identification of a BLE advertisement



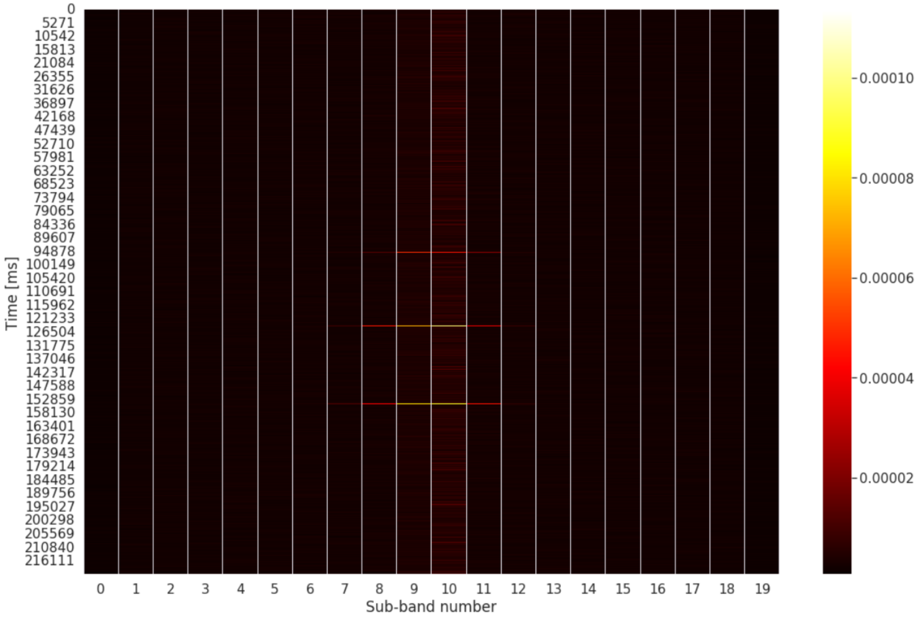
3 Channel characterization: identification of a BLE advertisement

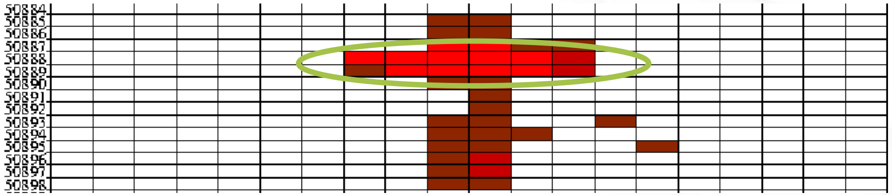




- Each line represents 250 µs
- Each rectangle represents 200 kHz
- 6 rectangles are detected to be a signal ➔ 6*200kHz = 1200 kHz
- BT is 1 MHz bandwidth ➔ we can associate this signal to BT

Channel characterization: identification of a BLE advertisement





Noise ⇔ energy < threshold

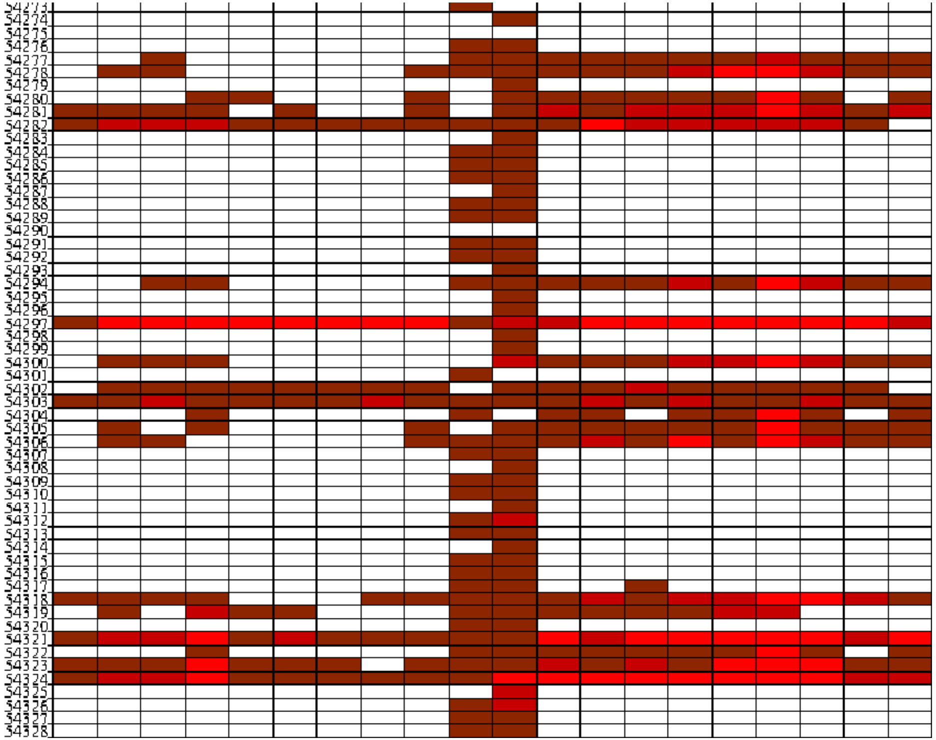
1*threshold ≤ Signal energy < 2*threshold

2*threshold ≤ Signal energy < 3*threshold

Signal energy ≥ 3*threshold

- Each line represents 250 µs
- Each rectangle represents 200 kHz
- 6 rectangles are detected to be a signal ➔ 6*200kHz = 1200 kHz
- BT is 1 MHz bandwidth ➔ we can associate this signal to BT

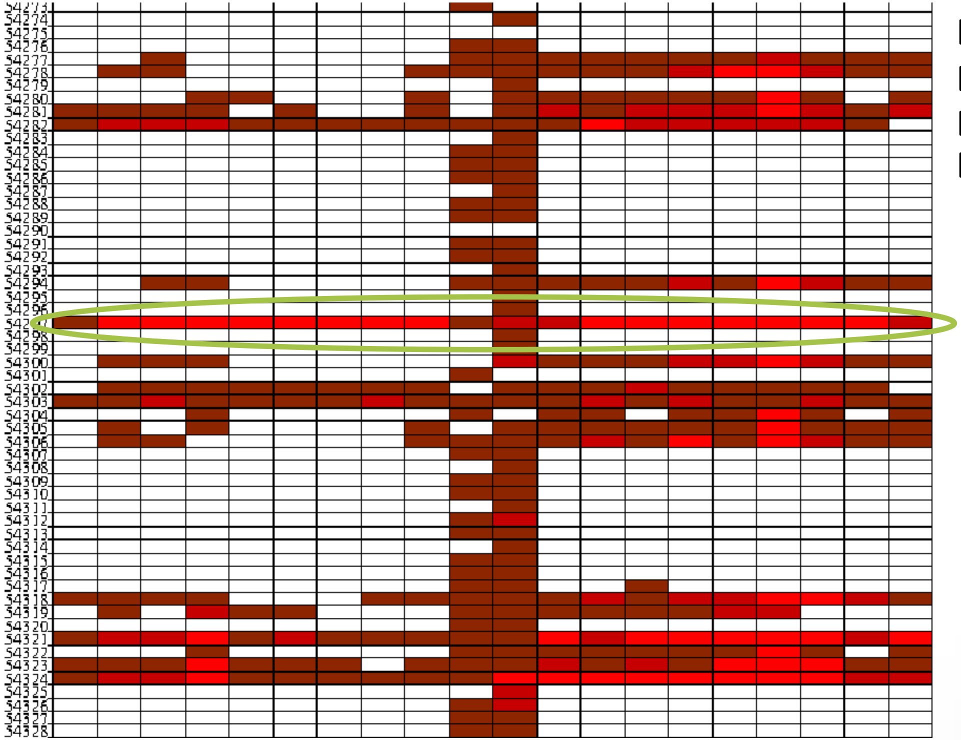
Channel characterization: identification of a Wi-Fi signal



- Noise ⇔ energy < threshold
- 1*threshold ≤ Signal energy < 2*threshold
- 2*threshold ≤ Signal energy < 3*threshold
- Signal energy ≥ 3*threshold

- Each line represents 250 µs
- Each rectangle represents 200 kHz
- Spectral occupancy of common wireless technologies:
  - LoRa 2.4 GHz = usually 812 kHz
  - BT = 1 MHz
  - BLE = 2 MHz
  - Wi-Fi = usually 22 MHz
- All the 4 MHz of bandwidth is detected as signal ➔ we can associate the signal to Wi-Fi

3 Channel characterization: identification of a Wi-Fi signal



- Noise ⇔ energy < threshold
- 1*threshold ≤ Signal energy < 2*threshold
- 2*threshold ≤ Signal energy < 3*threshold
- Signal energy ≥ 3*threshold

- Each line represents 250 µs
- Each rectangle represents 200 kHz
- Spectral occupancy of common wireless technologies:
  - LoRa 2.4 GHz = usually 812 kHz
  - BT = 1 MHz
  - BLE = 2 MHz
  - Wi-Fi = usually 22 MHz
- All the 4 MHz of bandwidth is detected as signal ➔ we can associate the signal to Wi-Fi

Channel characterization for coexistence study

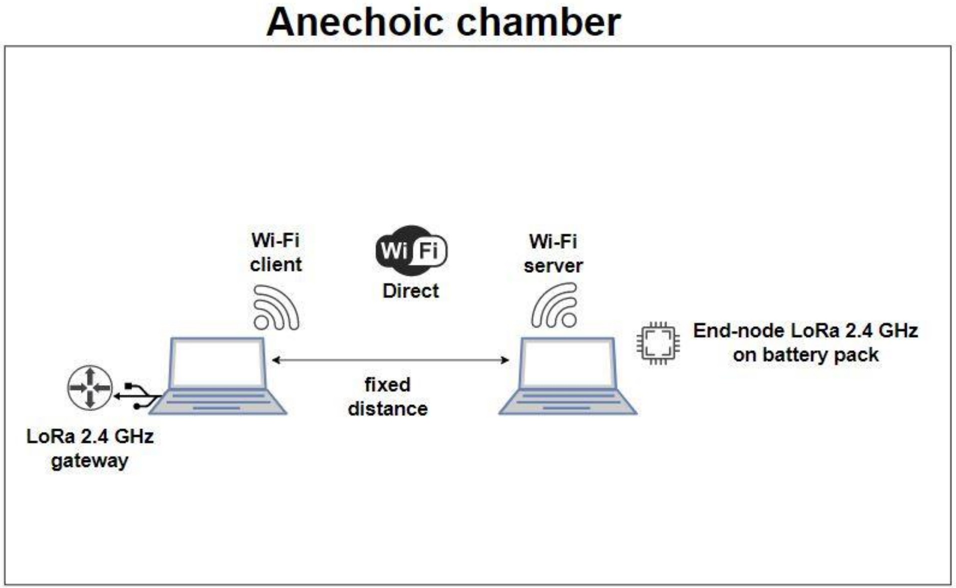


Figure 4 – Setup for coexistence study between LoRa 2.4 GHz and Wi-Fi in anechoic chamber

- **LoRa 2.4 GHz side:**
  - SX1280 provided by Semtech
  - 2 fixed configurations (highest data rate and highest reliability)
  - Varying P_{Tx} between 0 dBm and 13 dBm with a 3 dBm step
  - Evaluate the three current channels for LoRa 2.4 GHz (2403, 2425 and 2479 MHz)
- **Wi-Fi side:**
  - 2 Linux computer connected in Wi-Fi Direct
  - Generate UDP traffic with iperf or mgen
  - Varying P_{Tx} starting at 20 dBm and decreasing
  - Varying channels depending of LoRa channel evaluated
- **General parameters:**
  - Distance between devices is fixed
  - For each packet (LoRa and Wi-Fi), we collect RSSI, SNR and PDR
  - Experiments performed in anechoic chamber in order to avoid external interference

Concluding remarks

- LoRa 2.4 GHz will take place in a very crowded frequency band ➔ coexistence, and more specifically interference mitigation/detection is one big challenge to investigate

- We propose a methodology to characterize the 2.4 GHz ISM band environment and associate a signal to a known wireless technology

- We propose an experimental scenario to study the impact of LoRa on Wi-Fi performance, and vice versa, in a controlled environment

4

Perspectives: What’s next?

- Improve the technology identification method
  - Improve the definition of the different power levels ➔ make a gradient figure depending of the strength of the recorded signal
  - Automatize the technology identification

- Future works:
  - Run experiments on Wi-Fi and LoRa interference:
    - (1) In a controlled environment ⬄ anechoic chamber
    - (2) In an uncontrolled environment ➔ using the channel characterization in a work building
  - Improve coexistence base on previous results
  - Repeat experiments for other 2.4 GHz wireless technologies such as BT and BLE

Thank You

Questions?

Gwendoline HOCHET DEREVIANCKINE | gwendoline.hochet-derevianckine@inria.fr

References

[1] M. Höyhtyä, J. Lehtomäki, J. Kokkoniemi, M. Matinmikko and A. Mämmelä, "Measurements and analysis of spectrum occupancy with several bandwidths," in *2013 IEEE ICC*, Jun. 2013.

[2] G. Chen, W. Dong and J. Lv, "Lofi: Enabling 2.4 GHz LoRa and WiFi coexistence by detecting extremely weak signals," in *2021 IEEE INFOCOM*, May. 2021.

[3] C. Orfanidis, LM. Feeney, M. Jacobsson and P. Gunningberg, "Investigating interference between LoRa and IEEE 802.15. 4g networks," in *2017 13th IEEE WiMob*, Oct. 2017.

[4] Z. Li and Y. Chen, "BLE2LoRa: Cross-Technology Communication from Bluetooth to LoRa via Chirp Emulation," in *2020 17th IEEE SECON*, Jun. 2020.

[5] P. Gawlowicz, A. Zubow, and F. Dressler, "Wi-Lo: Emulating LoRa using COTS WiFi," *arXiv:2105.04998 [cs]*, May 2021, arXiv: 2105.04998. [Online]. Available: http://arxiv.org/abs/2105.04998

When the SF decreases, the sensitivity decreases too.

SF	SNR limit (in dB)
12	-20
11	-17,5
10	-15
9	-12,5
8	-10
7	-7,5
6	-5
5	-2,5

	SF-BW	Signal RSSI (in dBm)	Channel RSSI (in dBm)
LoRa sub-GHz	7-500	-116	-111
	12-125	-137	-117
	8-125	-127	-117
LoRa 2.4 GHz	5-1625	-99	-106
	12-203	-130	-115
	7-812	-112	-109
	11-812	-123	-109

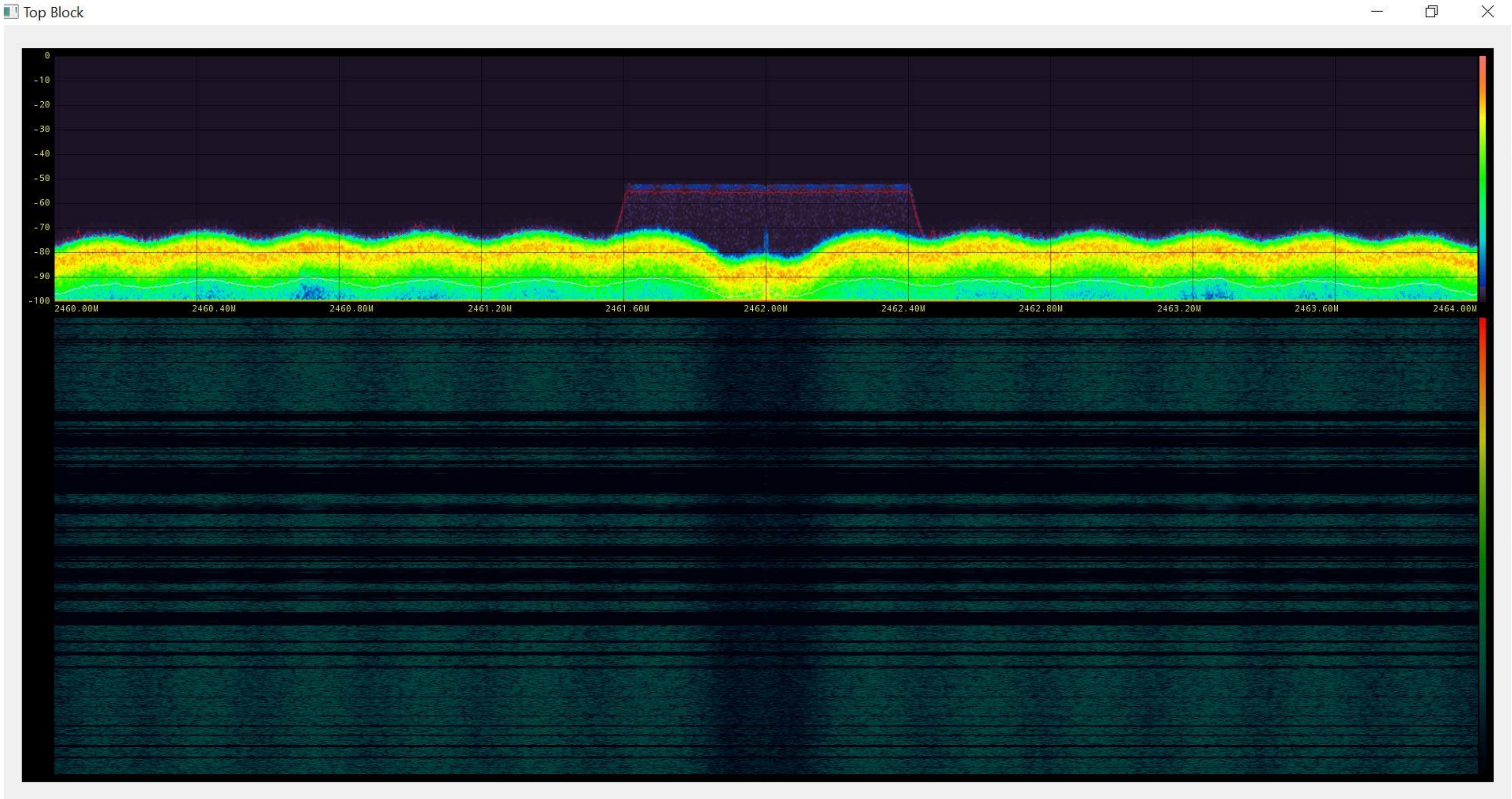
Steps of the signal processing:

- FFT on binary file containing data
- Apply the Perceval theorem to compute energy ($\sum abs(z^2)$ with z = I +Q) of each sample
- Convert the 2D Matrix, containing energies, into a 1D vector and sort the resulting vector
- Compute a threshold
  - **If energy ≥ threshold ➔ signal**
  - **If energy < threshold ➔ noise**
- Convert the energy vector into a figure according to the compute threshold
- Associate a signal to a known wireless technology of the 2.4 GHz ISM band

3

[10]:
```python
# START FFT loop
# Define the sub-band size by splitting into various part the total bandwith
Fs = sample_rate
Fs_sub = Fs/n_subchannel

# What BW represent one bin ?
freq_bin = Fs/fft_len

# give the scale of FFT results <=> from -Fs/2 to Fs/2
f= Fs*(np.arange(0, fft_len, 1) - fft_len/2)/fft_len

# How many bins are necessary to represent the sub band (Fs_sub) ?
nb_bin= int(Fs_sub/freq_bin)

nrg_tot_time_vec = np.zeros((math.floor(duration/granularity),1))

for k in range (0,math.floor(duration/granularity)):
    x_tmp = x[k*fft_len:(k+1)*fft_len-1]
    y_tmp = fft(x_tmp,fft_len)/np.sqrt(fft_len) # make FFT on fft_len samples of x
    y = np.fft.fftshift(y_tmp) # shift the zero-frequency component to the center of the spectrum

    # Parceval theory (energy time domain = energy frequency domain)
    nrg_tot_time = np.sum(np.abs(x_tmp)**2)
    nrg_tot_freq= np.sum(np.abs(y)**2)

    nrg_tot_time_vec [k] = nrg_tot_time

    nrg_acc = 0
    nrg_vec=[] # create an empty vector to put in it each value of nrg calculate in the loop
    for m in range (0, n_subchannel):
        bin_vec = np.arange(m*nb_bin,(m+1)*nb_bin-1, 1)
        nrg = sum(abs(y[bin_vec])**2)
        nrg_acc = nrg_acc + nrg
        nrg_vec.append(nrg)

    Matrix_total[k][0:n_subchannel]= nrg_vec
#END FFT loop
```

3


Top Block
0
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
2460.00M
2460.40M
2460.80M
2461.20M
2461.60M
2462.00M
2462.40M
2462.80M
2463.20M
2463.60M
2464.00M