IOT Exercise #3 report

Francesco Spangaro - Luca Tosetti

20 March 2025

Academic Year 2024 - 2025

Contents

1 EQ1

We were tasked with finding the biggest LoRa SF for having a success rate of at least 70% in a European LoRaWAN network (carrier frequency 868 MHz, bandwidth 125 kHz). The network is composed of one gateway and fifty sensor nodes, each transmitting packets with payload size of L bytes, according to a Poisson process with intensity lambda equal to one packet per minute. L is evaluated with the formula: $L = XY + 3$. XY being the last two digits of the leader's person code. In our case: $L = 44 + 3 = 47\ bytes$.
The airtime is computed using the following formulas:

$$Transmission\ duration\ (airtime) = T$$

$$Number\ of\ Nodes = N$$

$$Packet\ arrivals\ poisson\ process\ parameter = \lambda$$

$$\frac{Channel\ load}{traffic} = G = T * \lambda * N$$

$$Successful\ packet\ transmission = P_s = e^{-2G}$$

We computed all the airtimes for the different spreading factors, obtaining the following:

Input bytes	Spreading Factor	Region	Bandwidth	Airtime
47	SF7	EU868	125 kHz	112.9 ms
47	SF8	EU868	125 kHz	205.3 ms
47	SF9	EU868	125 kHz	369.7 ms
47	SF10	EU868	125 kHz	698.4 ms
47	SF11	EU868	125 kHz	1478.7 ms
47	SF12	EU868	125 kHz	2629.6 ms

After evaluating the airtimes, we computed the success probability, obtaining the following:

Spreading Factor	G	Probability	
SF7	0.094	82.86%	✓
SF8	0.171	71.03%	✓
SF9	0.308	54.01%	χ
SF10	0.582	31.22%	χ
SF11	1.232	8.51%	χ
SF12	2.191	1.25%	χ

The biggest LoRa SF found that has a success rate of at least 70% is SF8.

2 EQ2

The steps needed to get the system fully operational are:

1. Setup the hardware that will perform the measurements by connecting the DHT22 temperature and humidity sensor to the Arduino MKR WAN 1310 board. Connect the DHT22's VCC, GND and DATA pins to the 3.3V, GND and one of the digital pins on the Arduino board, respectively.

2. Install the required Arduino libraries, such as MKRWAN to allow usage of the 'Murata' LoRa transceiver on the Arduino board, and DHT sensor, needed to use the DHT22 sensor.

3. Retrieve the DeviceEUI by loading on the Arduino board an example sketch provided by the MKRWAN library. The sketch, once loaded, allows us access to the Serial Monitor of the Arduino board, containing the DeviceEUI.

4. Connect to a public LoRaWAN gateway. Find a TTN gateway, in range for communication, that supports the reception of LoRa packets. This way the Arduino board will connect to the gateway, sending the measurements. The gateway will then forward them to the TTN Network Server. Gateways can be found online on websites such as `https://ttnmapper.org/heatmap/`.

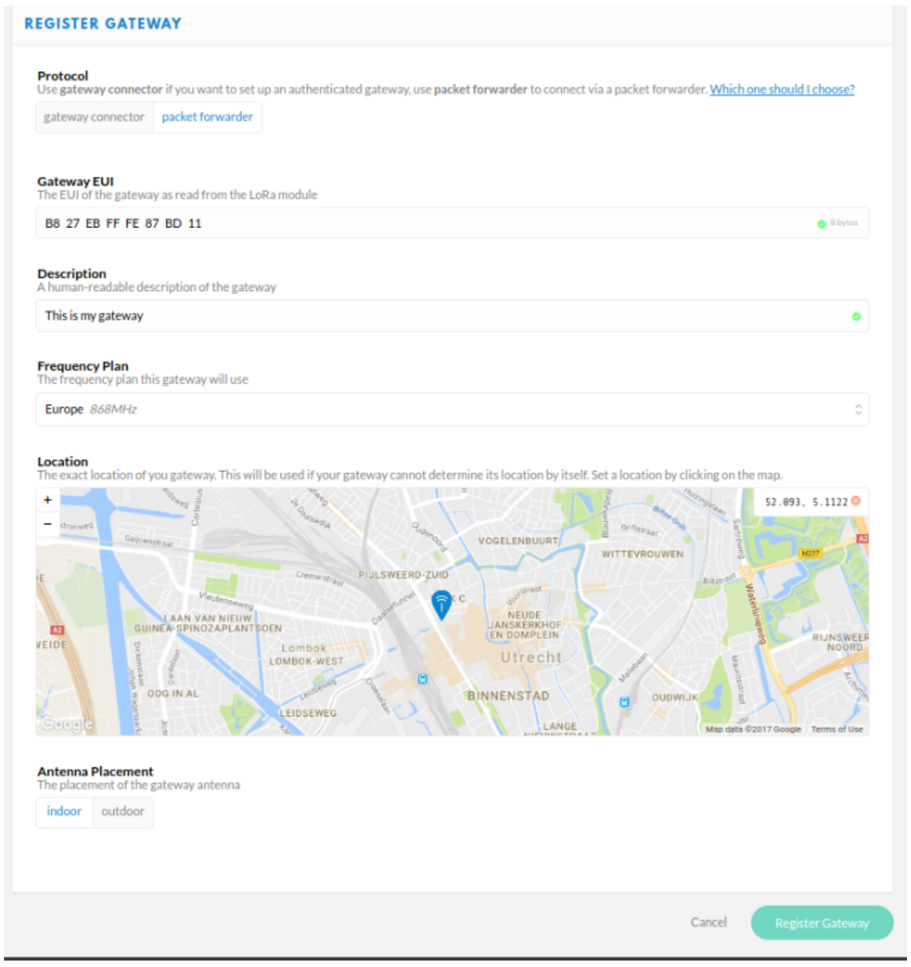
5. (Optional) Buy and setup a LoRaWAN gateway. If there are no public LoRaWAN TTN gateways in range, create an account on the TTN website (`https://www.thethingsnetwork.org/`), log in and access the TTN website console. Then register the gateway by following the setup configuration (e.g. selecting 'packet forwarder', inserting the gateway EUI, Europe Frequency plan, the location, description of the antenna placement, and all information needed). For more info on the process, see [1].

6. Register the device on The Things Network. If an account has not yet been created, create one on TTN's website and log in. Navigate to the TTN console and create a new application. Register the Arduino board as a new device by inserting the DeviceEUI obtained at step 3 into the application. Set as join method 'OTAA' (Over The Air Activation) and generate a new AppKey and AppEUI. Save the AppEUI, DeviceEUI and AppKey for later use in the Arduino code.

7. Write the Arduino code. The Arduino code needs to:
   - Read temperatures and humidity values from the DHT22 sensor.
   - Encode the readings into a byte array, in order to reduce the payload size for LoRa transmissions.

   - Connect to a LoRaWAN network using OTAA. The AppEUI and AppKey are needed in this step.

8. Load the Arduino code to the board.
9. Set up a TTN payload formatter. Navigate to the uplink Payload Formatters section in the TTN application. Select a custom JS formatter, needed for decoding the byte array sent from the Arduino board back into comprehensible readings of temperature and humidity.
10. Configure ThingSpeak by creating an account at ThingSpeak's website (`https://thingspeak.mathworks.com/`). Create a new channel with 'Temperature' and 'Humidity' as fields. Obtain the channel's credentials, particularly the 'Write API Key' and 'Channel ID'.
11. Configure WebHook Integration on TTN, since TTN offers HTTP integration with ThingSpeak. To make use of this integration, navigate to the Webhooks page in the integration page of the TTN application. Add a new webhook by selecting the ThingSpeak template and configure it by specifying a unique Webhook ID, the Channel ID and Write API Key previously obtained on ThingSpeak.
12. Now deploy the Arduino board and start the program. In the TTN application the sensed data should appear inside the 'Live Data', as well as inside the newly created ThingSpeak channel.


REGISTER GATEWAY

Protocol
Use **gateway connector** if you want to set up an authenticated gateway, use **packet forwarder** to connect via a packet forwarder. Which one should I choose?
gateway connector [✓ packet forwarder]

Gateway EUI
The EUI of the gateway as read from the LoRa module
B8 27 EB FF FE 87 BD 11 — 8 bytes

Description
A human-readable description of the gateway
This is my gateway

Frequency Plan
The frequency plan this gateway will use
Europe 868MHz

Location
The exact location of you gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.
52.093, 5.1122

Antenna Placement
The placement of the gateway antenna
[✓ indoor] outdoor

Cancel | Register Gateway

Figure 1: TTN gateway setup procedure.

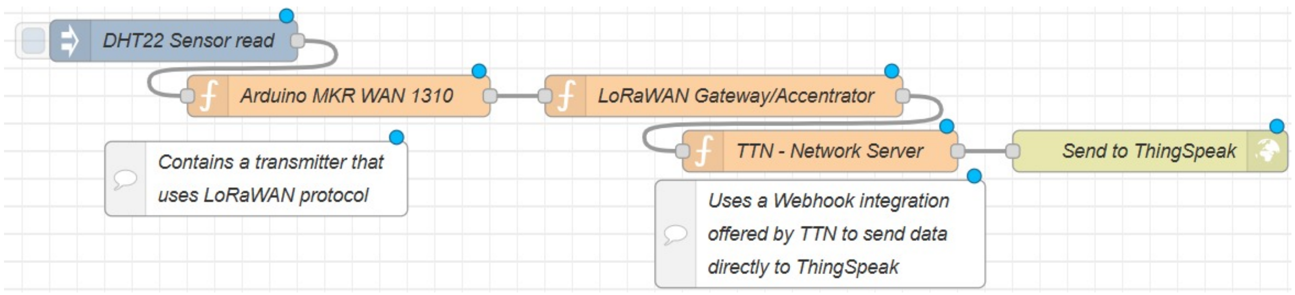

DHT22 Sensor read
Arduino MKR WAN 1310
LoRaWAN Gateway/Accentrator
Contains a transmitter that uses LoRaWAN protocol
TTN - Network Server
Send to ThingSpeak
Uses a Webhook integration offered by TTN to send data directly to ThingSpeak

Figure 2: Node-Red sketch of the system-block diagram describing the needed steps.

3 EQ3

3.1 Introduction

During research, we have found two different versions of the paper provided on LoRaSim, each containing slightly different graphs. The first, provided on WeBeep, is the oldest, while the second is the newer version. The newer version contains updated graphs that differ from the older ones due to a bug in the LoRaSim simulator. Information about the different simulator parameters has been taken from here.

3.2 Figure 5

In order to replicate figure 5 of the newer LoRaSim paper, we have adopted the following parameters:

- Number of nodes: $range(1,\ 10)\ +\ range(10,\ 300,\ 10)\ +\ range(300,\ 1650,\ 100)$.
- Transmission rate: $1e6$.
- Experiment (exp): different based on the configuration set SN used. For SN3 we used exp=4, for SN3 we used exp=3 and for SN5 we used exp=5. Note: by using exp=1 for SN3 its graph representation is more similar to the graph present in the older version of the paper even though we can infer from the paper that the parameter value used in the displayed image experiment is exp=4.
- Duration: 86.400.000 ms = 1 day.
- Collision flag set to 1.

In figure is the obtained result side by side the paper's original.

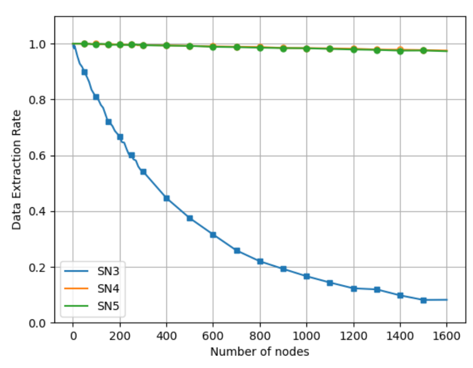


Figure 3: Replica of the paper's figure 5 graph.

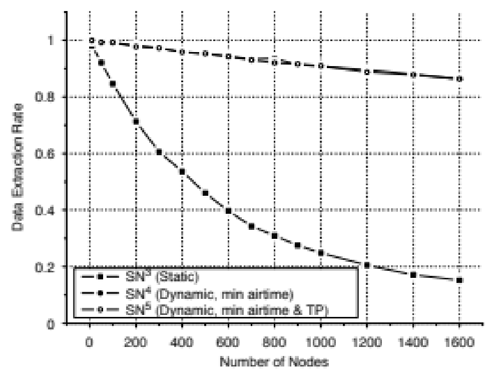


Figure 4: Paper's figure 5 graph.

3.3 Figure 7

In order to replicate figure 7 of the newer LoRaSim paper, we have adopted the following parameters:

- Number of nodes: $range(1,\ 10)\ +\ range(10,\ 300,\ 10)\ +\ range(300,\ 1650,\ 100)$.
- Transmission rate: $1e6$.
- Experiment (exp): we can infer from the paper that exp=0 was used for the experiment, since it is stated that they used the same configuration as for SN1. This allows us to obtain a graph that is only *similar* to the one contained in the paper, since here the success rate decreases faster than in the paper.
- Duration: 86.400.000 ms = 1 day.
- Collision flag set to 1.
- Base station number: [1, 2, 3, 4, 8, 24] as used in the paper.

In figure is the obtained result side by side the paper's original.

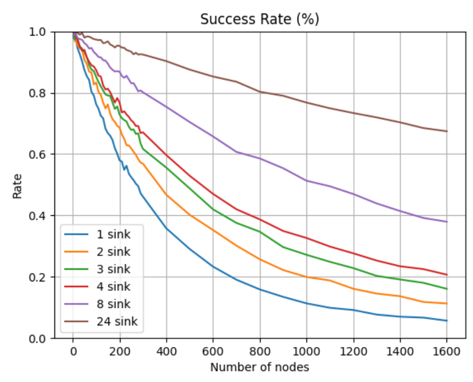


Figure 5: Replica of the paper's figure 7 graph.

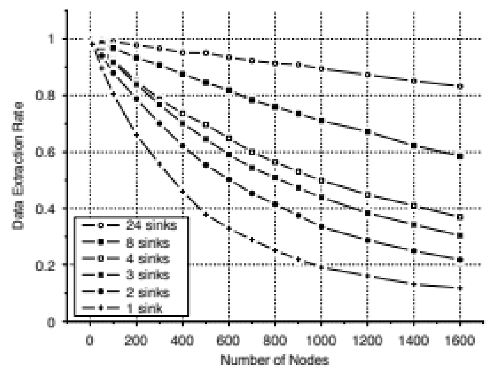


Figure 6: Paper's figure 7 graph.