Internet Of Things (IOT) - Challenge 3

Mohammadreza Zamani (10869960) – Asal Abbasnejadfard (10974178)

Exercise

EQ1) A LoRaWAN network in Europe (carrier frequency 868 MHz, bandwidth 125 kHz) is composed by one gateway and 50 sensor nodes. The sensor nodes transmit packet with payload size of L byte according to a Poisson process with intensity lambda = 1 packet / minute. Find the biggest LoRa SF for having a success rate of at least 70%. Hint: use https://www.thethingsnetwork.org/airtime-calculator to compute the airtime of a packet.

Solution:

Person code: 10869960

XY = 60

L = 3 + XY bytes = 63

Total transmission rate:

Each node sends 1 packet/minute = 1/60 packet/second

$$\lambda = 1\text{pkt/min} = \frac{1}{60}\text{pkt/sec}$$

50 sensor nodes

Total rate $= \boldsymbol{N} \times \boldsymbol{\lambda}$ = 50 × (1/60) = 0.833 packets/second

ALOHA success model formula:

$$G = N \cdot \lambda \cdot t = 0.833 \cdot t$$

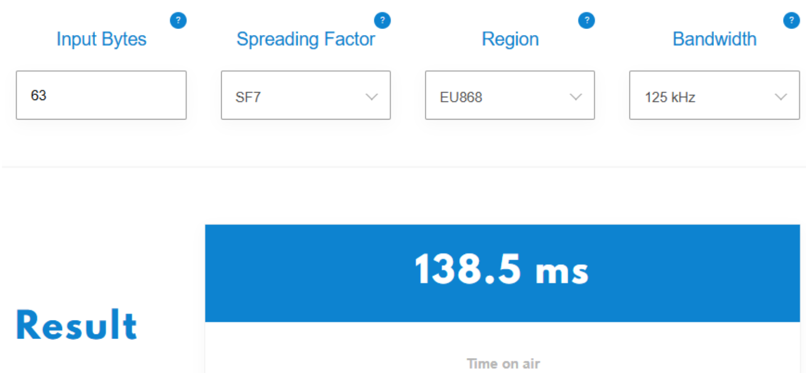
$$\text{Success Rate} = e^{-2G}$$

We want to:

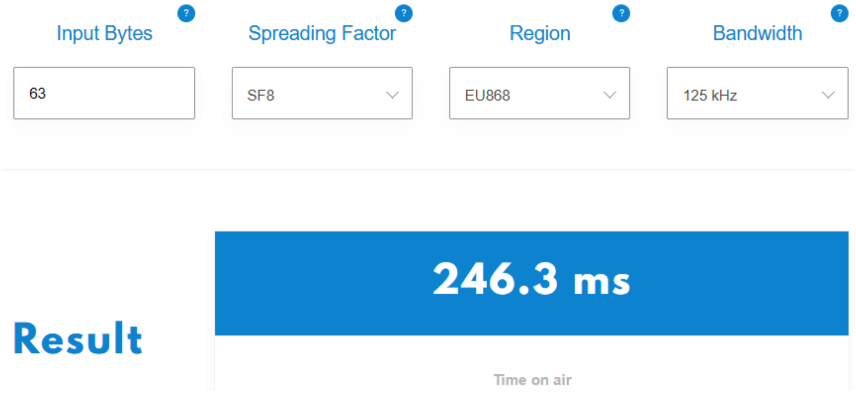
$$\text{Success Rate} \geq 0.70 \Rightarrow G \leq -\frac{\ln(0.7)}{2} \approx 0.178 \Rightarrow t \leq \frac{0.178}{0.833} \approx 0.214 \text{ second}$$

These are the time on airs of different spreading factors (SF7, SF8, SF9, SF10, SF11, SF12):

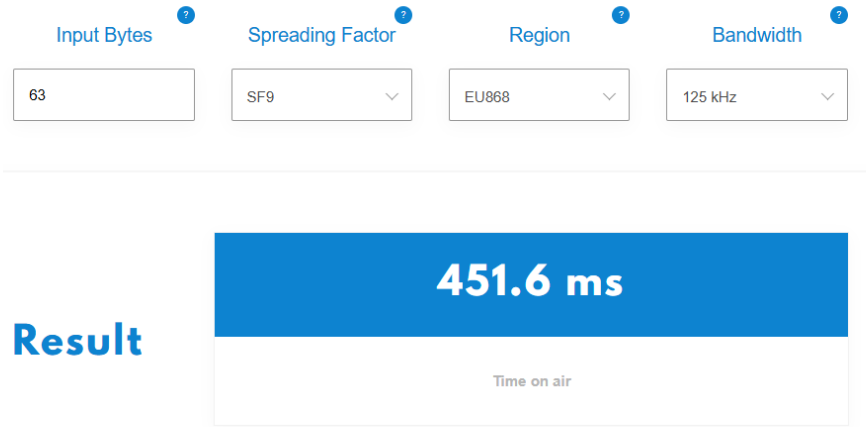
SF7:



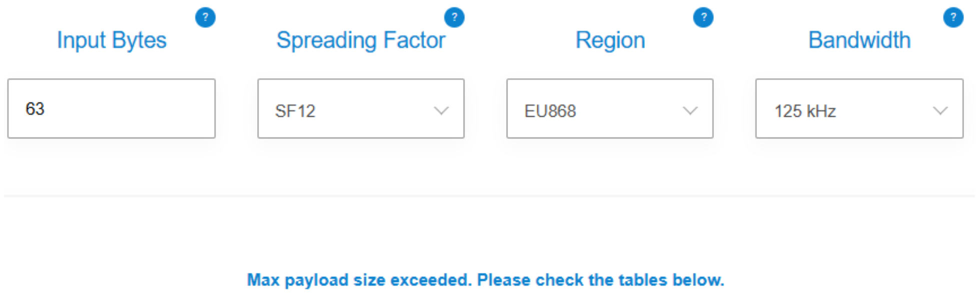
SF8:

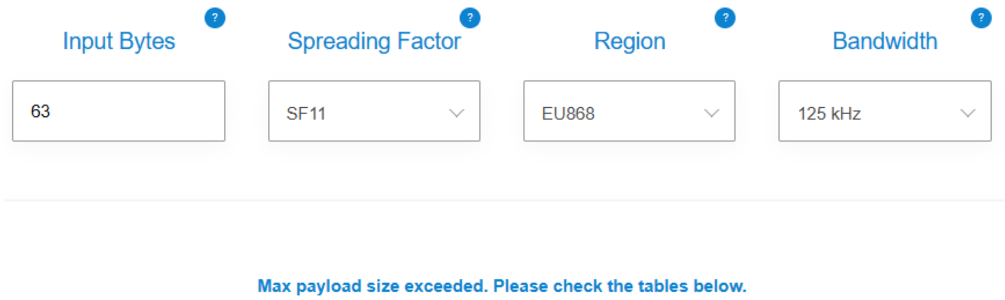


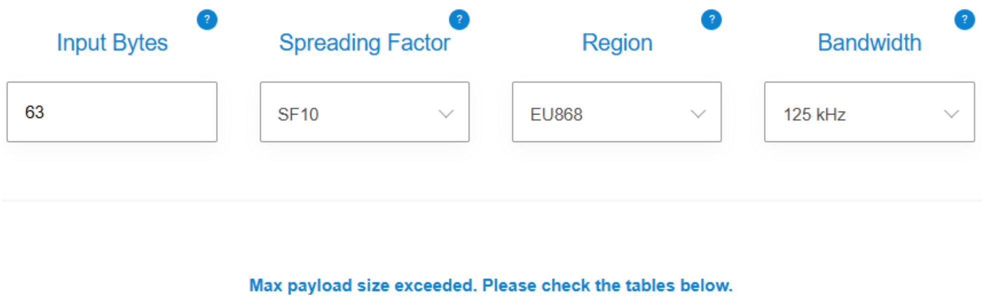
SF9:



For SF10, SF11 and SF 12:







SF	Airtime (ms)	Airtime (s)	G	Success Rate	≥70%?
7	**138.5**	**0.139**	**0.116**	**0.793**	**YES**
8	246.3	0.246	0.205	0.664	NO
9	451.6	0.452	0.377	0.470	NO

The largest spreading factor that is allowed is **SF7**

EQ2) You have purchased an Arduino MKR WAN 1310 and wish to create a system that reads temperature and humidity data from a DHT22 sensor and sends this data wirelessly to ThingSpeak over LoRaWAN. Design a complete system block diagram (sketch in Node-Red) and describe, in detail, the steps you would need to take to get the system fully operational.

Solution:

This is the flowchart that we should design.

DHT22 Sensor

↓

Arduino MKR WAN 1310 (LoRa library + DHT library)

↓

LoRa Transmit (868 MHz, SF configurable)

↓

LoRa Gateway (e.g., TTN)

↓

TTN Network Server

↓

Integration via HTTP/MQTT

↓

ThingSpeak (Data Storage & Plotting)

The following steps describe in detail the full system flow for sending temperature and humidity data from a DHT22 sensor to ThingSpeak over a LoRaWAN network, assuming all hardware components are physically available:

1. **DHT22 Sensor**

   The DHT22 sensor measures the ambient temperature and humidity. It generates digital output signals corresponding to these environmental parameters, which can be directly read by a microcontroller.

2. **Arduino MKR WAN 1310 (with LoRa and DHT Libraries)**

   The Arduino MKR WAN 1310 serves as the microcontroller unit (MCU) that reads the data from the DHT22 sensor. It uses specific libraries: A DHT sensor library to interpret DHT22 data signals. A LoRaWAN library to prepare and send the data wirelessly. The Arduino collects the measurements and encodes them into packets for LoRaWAN transmission.

3. **LoRa Transmission (868 MHz, Spreading Factor Configurable)**

   Using its built-in LoRaWAN module, the Arduino MKR WAN 1310 transmits the collected sensor data over a frequency of 868 MHz (standard for Europe). Transmission parameters such as Spreading Factor (SF) can be configured to balance data rate, range, and power consumption depending on the network design.

4. **LoRa Gateway (e.g., The Things Network - TTN Gateway)**

   The LoRa gateway receives the transmitted LoRa signals from the Arduino. It acts as a bridge between the LoRaWAN network and the Internet. The gateway decodes the LoRa packets and forwards them securely to a network server.

5. **TTN (The Things Network) Network Server**

   The TTN network server processes the incoming packets: It authenticates the device; It checks the data integrity. It prepares the sensor data to be made available through cloud integration (e.g., via HTTP or MQTT protocols).

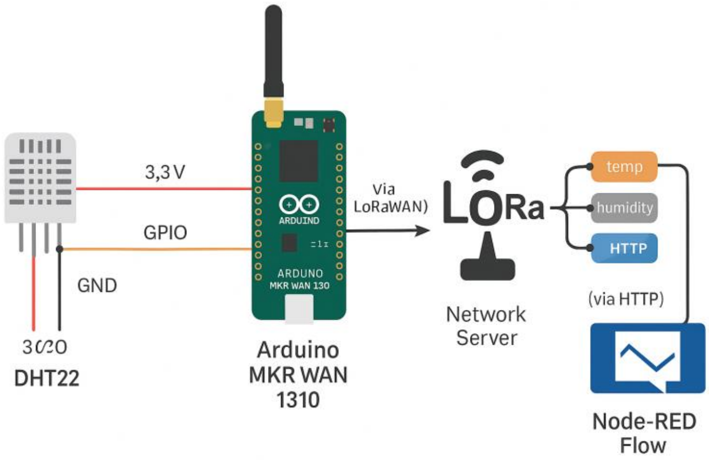
6. **Integration via HTTP/MQTT**

   The server forwards the sensor data through either: HTTP requests: Sending POST requests with the data. MQTT protocol: Publishing the data to an MQTT broker. Depending on the integration method, the data will be packaged in the required format for the next destination.

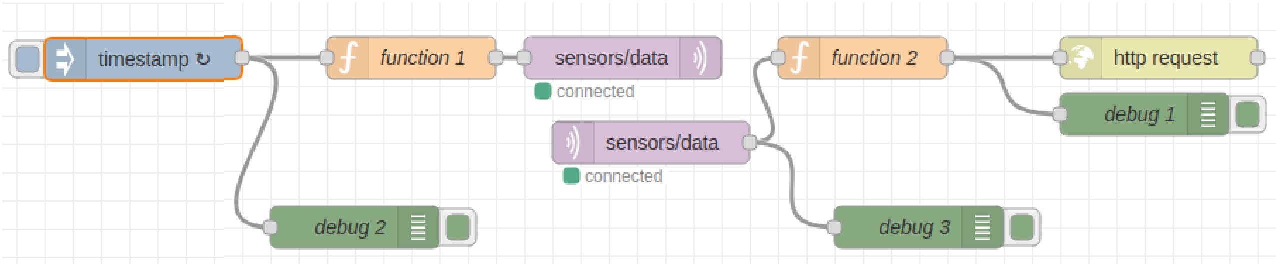
7. **ThingSpeak (Data Storage and Visualization)**

   ThingSpeak acts as the final cloud platform where: The data is stored into defined fields (such as Field 1 for temperature and Field 2 for humidity). Real-time visualization (e.g., charts) is generated for monitoring and analysis. Optional data analysis and trigger actions can be implemented based on incoming values.

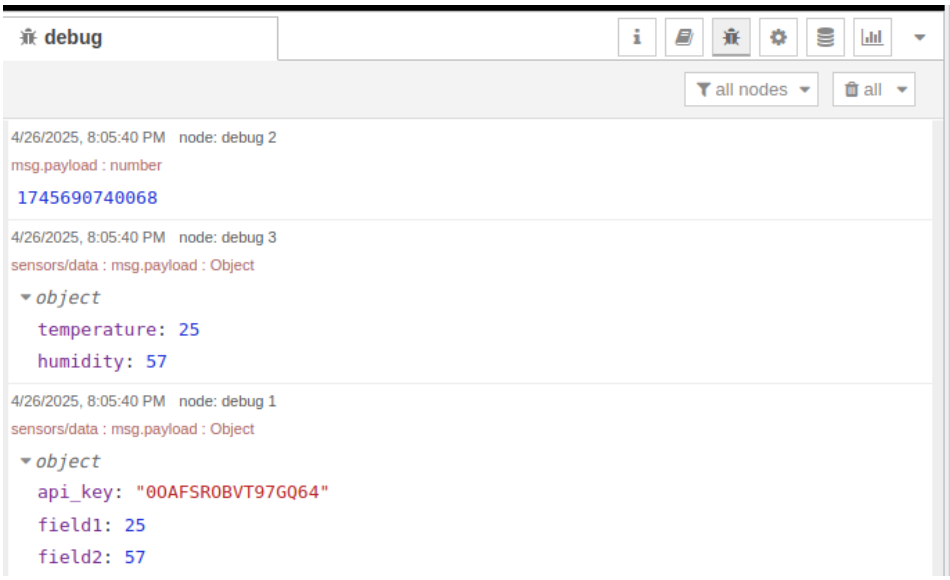
In fact, graphically, the figure below is considered.



Since we don't have physical access to components like Arduino and DHT22 sensor, we will implement it in node red as a simulation. This is our flow in node red.

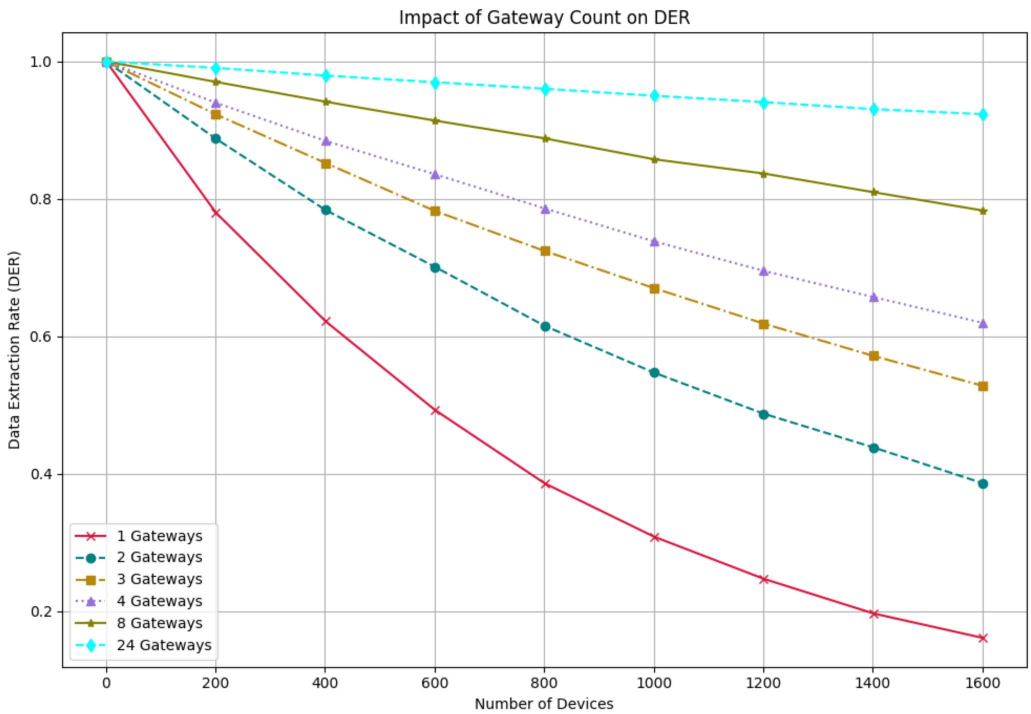


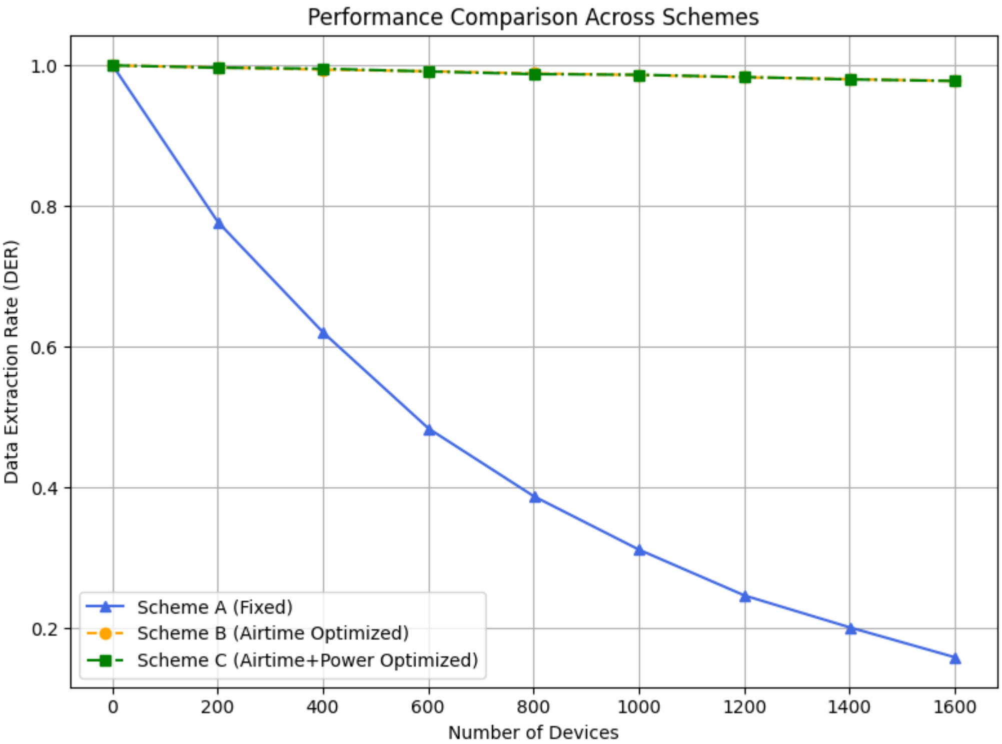
And There are some samples of output.



Instead of reading real temperature and humidity, we generate random values between reasonable ranges. Instead of using LoRa radios, we use MQTT messaging locally to simulate data transmission and reception. We format the received data and send it over real HTTP POST to ThingSpeak, completing the data pipeline. This is the channel link: https://thingspeak.mathworks.com/channels/2923139.

EQ3)




Performance Comparison Across Schemes
Data Extraction Rate (DER)
Number of Devices
1.0
0.8
0.6
0.4
0.2
0
200
400
600
800
1000
1200
1400
1600
Scheme A (Fixed)
Scheme B (Airtime Optimized)
Scheme C (Airtime+Power Optimized)