Setting Up A LoRaWAN Network For Remote Data Transmission

Jose M. Level, ***Student, Florida International University***
Kirklan Williams, ***Student, Florida International University***

EEE4717 Intro to Security of IoT, Electrical and Computer Engineering Department, Florida International University, 10555 West Flagler Street, Miami, FL 33174, USA

This lab and its corresponding code was finalized on December 5th, 2020 to be used in EEE4717 Intro to Security of IoT. This lab and its corresponding code is free to use and modify.

Abstract: What is LoRaWAN? LoRaWAN is a low power wide area network protocol that is being used for bidirectional communication for Internet of Things devices. LoRaWAN brings to the industry very low consumption across a large network with millions of devices. This project aims to create a lab (step-by-step guide) for setting up a LoRaWAN using RaspberryPi and a LoRa Shield and send temperature data from a remote lake to an online endpoint to be analyzed using the things network. In the future this lab could have a LoRaWAN gateway at the FIU campus and so students could connect their LoRa device to it. In this project we go through the steps of identifying and implementing the supported security measures available to the LoRaWAN through the documentation from the LoRaWAN alliance including key distribution protocols commonly used for this kind of network.

Index Terms: LoRa, LoRaWAN, IoT.

1. Objective

- Understanding the difference between LoRa and LoRaWAN
- Understanding how LoRaWAN is secured and how it remains secure while using public access points.
- Understanding the use cases for LoRaWAN and its basic implementation using Raspberry Pi
- Understanding what The Things Network is and its role in LoRaWAN implementation

2. Lab Scenario

You will use the LoRa Radio Bonnet to communicate with a gateway and transmit data from a remote node to The Things Network.You will set up an account on The Things Network and relay the required settings in your code. Once completed, your code will send data from your device to the end point of The Things Network where it will be visible and decoded as a payload. You are expected to install any dependencies including but not limited to those within the instructions in this lab.

3. Procedures

3.1. Preliminary Steps

Make sure you have all of the hardware for the lab:

- Raspberry Pi Zero
- LoRa Radio Bonnet with OLED - RFM95W @ 915MHz
- (optional) SMA to uFL/u.FL/IPX/IPEX RF Adapter Cable Antenna for LoRa Bonnet

Next, Configure your Raspberry Pi for the lab:
More information on setup here:
(https://learn.adafruit.com/circuitpython-on-raspberrypi-linux/installing-circuitpython-on-raspberry-pi)

I have provided for you a bash file to help install some of the prerequisite modules (Installer.sh)

Update the Pi with the following commands:
sudo apt-get update
sudo apt-get upgrade
sudo pip3 install –upgrade setuptools

Make python3 default:
sudo apt-get install -y python3 git python3-pip

Enable i2c and spi:
sudo raspi-config
Go to option 5 or option 3 “interfacing options”

Enable spi and i2c
Select finish
sudo reboot

Next we are gonna install some libraries for python:
pip3 install RPI.GPIO
pip3 install adafruit-blinka

Install CircuitPython Libraries:
sudo pip3 install
adafruit-circuitpython-ssd1306
sudo pip3 install
adafruit-circuitpython-framebuf
sudo pip3 install
adafruit-circuitpython-rfm9x

Now Download the sample code (Rfm9x_check.py) to verify your setup.

You’ll also want to download the font file, font5x8.bin, and copy it into the same directory as the script:
wget https://github.com/adafruit/Adafruit_CircuitPython_framebuf/raw/master/examples/font5x8.bin
Run the sample code:
python3 rfm9x_check.py

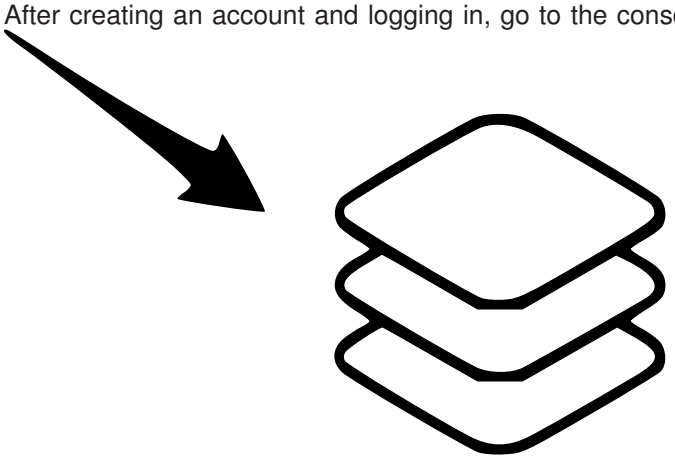
3.2. Setting up the Things Network

When discussing LoRa, there are a few different terms that are commonly used and often misunderstood. LoRa is a low-power wide-area network protocol, and it is used to transmit packets over a long range while consuming insignificant amounts of power. LoRaWAN uses gateways and multiple LoRa devices to create a network that is used to transmit packets faster and across a

wider area.
The Things Network is the connection between the things devices and its applications. The LoRaWAN gateways connect to the things network and allow the LoRa devices to communicate on the network delivering usable data to the user.

3.2.1. Create an Application with TTN

Go to The Things network Login and make an account.
(https://account.thethingsnetwork.org/register)
After creating an account and logging in, go to the console and click “Applications”



Enter an application-ID and device description (It can be whatever you want)
Set your handler Registration section to us-west and click add Application.

3.2.2. Device Registration

Click “Register Device” in the devices section.
Once in the page, make a unique device ID.
The Device EUI and App Key will be generated for you.
You must select the App EUI front the drop down list.
Click Register on the bottom right hand side.

3.2.3. Change activation method and disable Frame Counter Checks

Click on the settings tab in device overview.
Navigate to the activation method section and select ABP.
Go to frame counter width and select 16bit and disable frame counter checks.

What are frame counter checks?

Frame counter checks are used to prevent spoofing. As a LoRa packet is sent, it contains a frame counter value. The network is looking for the frame counter to always be getting bigger.

Why is it useful to turn in off during development?

The frame counter is checking for frame packets to continuously get bigger. However, with simple nodes, every time the device is restarted, the frame counter resets. So, the packets will be rejected

until the counter exceeds the previous frame count.

Why is it important to turn this back on for production?

Once your network and devices have been developed, it is in best practice to save the counter in memory. You can also incorporate a way for the counter to work in conjunction with data and time, so resets will no longer create issues.

3.3. Download and modify the Code

sudo pip3 install adafruit-circuitpython-tinylora
Download the code provided for you:
radio_lorawan.py
Make sure you have the code in the same directory in your Pi as the font and setup check code (rfm9x_check.py) we used before.

Open the code in an editor to make the following additions:

- Fill out the “devaddr “ variable in the code with the Device Address from The Things Network Console you set up earlier. (*Note that the code must use Brackets[] instead of curly braces)
- Do the same for the Network Session Key to fill out “nwkey” in the code.
- Then again for the “app” variable make sure to use the Application Session Key from the console.

3.3.1. Running the Code

Make sure you are in an area within the coverage of a LoRa Gateway. Here is a link to the coverage map. *The installation/setup of a multi channel LoRa Gateway is not in the scope of this lab. Once in an appropriate area, run the code with the following command: python3 radio_lorawan.py

3.4. Building a Gateway

Configure another Raspberry pi same as above

- Connect the gateway Raspberry Pi to internet
- Make sure everything is setup properly on the new raspberry pi
- Run: python3 rfm9x_check.py

3.4.1. Single Channel Pi Gateway Setup

- Check if WiringPi is pre installed
- Run: gpio -v
- If something is returned then continue, if not then run sudo apt-get install wiringpi
- Run git clone https://github.com/adafruit/single_chan_pkt_fwd.git
- And cd into single_chan_pk_fwd/
- sudo make all
- If you are using a Raspberry Pi ZERO configuration the global_conf.json file. Replace ”is_pi_zero”: false with ”is_pi_zero”: true

3.4.2. Configuring the Things Network for Gateway

- Run python3 lorawan_gateway.py, This will display the devices EUI on the adafruit oled screen.
- Login in to your TTN account
- Click on Console - Gateways - Register Gateway.
- Tick I’m using legacy packet forwarder
- Enter the gateway EUI, fill out the description, and set frequency to US.

- Run/test the gateway run python3 lorawan_gateway.py
- On the status page you should now see connected

3.4.3. Transmitting Data

- Select the middle button on your adafruit bonnet. You'll notice that it is now waiting to receive packets.
- On your other raspberry pi you will want to run your code that you configured for the temperature sensor and transmit packets.
- Run python3 tempMonitor.py
- You will notice in TTN the data being received through the data tab, however it is important to notice the data is unreadable. When data is transmitted through the gateway you can see a packet has been transmitted but not necessarily it contents. This is done to make sure the users data is secured. The final application will have its own decode key to be able to obtain the transmitted data.
- Log on to the application side of the user that setup the sensor
- You will notice on the application side the data has been decoded and displays the proper values once you enter the payload ID.

3.5. References

1) "About LoRaWAN®." About LoRaWAN® — LoRa Alliance®, lora-alliance.org/about-lorawan.
2) GEMALTO, et al. "LoRaWAN™ SECURITY." Lora-Alliance.org, 2017, lora-alliance.org/sites/default/files/2019-05/lorawan_security_whitepaper.pdf
3) The Things Network, www.thethingsnetwork.org/.
4) Rubell, Brent. "Single Channel LoRaWAN Gateway for Raspberry Pi." Adafruit Learning System, learn.adafruit.com/raspberry-pi-single-channel-lorawan-gateway