# Understanding The Things Network's Software

Mohd Thaqif ABDULLAH HASIM

December 13, 2016

**Abstract**

During a full time preparatory week, I need to have a good understanding of the software provided by The Things Network in order to establish a LoRa network for the use of IoT.

## 1 Architecture of TTN

Data sent by the things (Nodes/devices) to the application will be routed through different routing service components. The gateway will act as a router by forwarding the packets sent by the devices to The Things Network. The Things Network is positioned between the gateways and the applications (see the figure below) and takes care of these routing and processing steps.

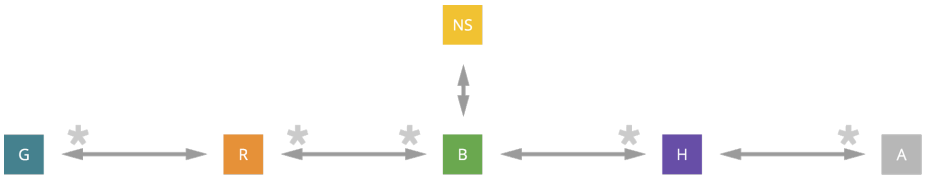


Figure 1: The Things Network's different routing service components: Gateway, Router, Broker, NetworkServer, Handler and Application

Nodes (N in the top Figure) broadcast LoRaWAN messages over the LoRa radio protocol. These messages are received by a number of Gateways (G). The Gateway is a piece of hardware that forwards radio transmissions to the backend. It is connected to one Router (R). The Router is responsible for managing the gateway's status and for scheduling transmissions. Each Router is connected to one or more Brokers (B). Brokers are the central part of The Things Network. Their responsibility is to map a device to an application, to forward uplink messages to the correct application and to forward downlink messages to the correct Router (which forwards them to a Gateway). The Network Server (NS) is responsible for functionality that is specific for LoRaWAN. A Handler (H) is responsible for handling the data of one or more Applications (A). To do so, it connects to a Broker where it registers applications and devices. The Handler is also the point where data is encrypted or decrypted.

The are two options to deploy a network :

1. Public Community Network

2. Private Network

We can run The Things Network on our own server by downloading TTN's open source routing services from GitHub `https://github.com/TheThingsNetwork/ttn`. I have downloaded it and tried to run TTN's routing services on my computer. I have passed the first step which is **to prepare the development environment** by installing some packages. Next step is **to set up TTN's backend for development**. At this step, after installing the dependencies for the development, the tests ran fail and the next steps can't be proceeded.

After visiting the TTN's FAQ, it is recommanded to just use the public community network, as this will save a lot of time. We can run a private deployment TTN's open source routing services if we know what we are doing.

# 2 Gateway

By using the pubic community network, I can concentrate on how to set up the gateway and make it communicate with TTN's backend. LIG owns some Kerlink gateways that need to be configured. A documentation on how to configure this type of gateway can be consulted at `https://www.thethingsnetwork.org/wiki/Hardware/Gateways/Kerlink`

## 2.1 Configure the Gateway

Kerlink's repository on GitHub contains the software en tools to use the Kerlink Gateway with TTN.

Steps of the configuration :

1. Download TTN's patch/update file and produsb.zip
2. Install the patch/update downloaded :
   (a) Extract the compressed files downloaded onto on empty USB flash drive formatted in *FAT-32*.Make sure there is no *.log* file.
   (b) Plug the USB flash drive into the gateway.
   (c) Wait for 5 min. During this time the gateway will reboot itself.
   (d) Unplug the key and check that a *.log* file has appeared. The file should contain the following text *WirmaV2 0x080XXXXX updated*. This log file prevents any further installation on the gateways to avoid cyclic reboots. To redo the update on same gateway, remove this log file from the flash drive reinsert it into the gateway USB. This is not needed if you update another gateway.
3. Logon to the gateway by using the SSH protocol, run
   ssh root@*Gateway's IP*
4. At your firewall system make sure the external IP-address used will map port 1700 to the internal IP-address of the gateway. This is needed when using NAT for internal IP-address translation.

5. Use the following command on the gateway to check whether data is being sent and received:
   tcpdump -i eth0 -n -vvvX host *hostname*
6. There is a documentation on how to regain *root* access on Kerlink

## 2.2 Connect the Gateway to TTN's backend

### 2.2.1 Server Address

Choose the router instance depanding on the region.

router.eu.thethings.network #EU 433 and EU 863-870

### 2.2.2 Configuring the Gateway

*NOTE: Please use a packet forwarder based on version 2.2.x. The new lora_pkt_fwd version 3.0.0 is not supported yet*

In the local_conf.json of the packet forwarder, update the fields server_address as follows:

```
"gateway_conf": {
        ...
        "servers": [{
            "server_address": "<insert server address here>",
            "serv_port_up": 1700,
            "serv_port_down": 1700,
            "serv_enabled": true,
            ...
        }]
}
```

Figure 2:

### 2.2.3 Checking Connectivity

1. Gateway communicates with the routers via an UDP connection on port 1700; So if we listen to this port we should see packets being transferred.
   *sudo tcpdump -AUq port 1700*
2. The packet forwarder periodically writes its status to a log file located at: /var/log/syslog
   *sudo tail -f /var/log/syslog*