

Praxis #7: Meshtastic

Prerequisites

Command Line sessions

Kit

Heltec ESP32 V3 LoRa Wifi Bluetooth

USB-C power supply (or lithium battery) Smart phone or laptop with Meshtastic app installed

Optional - Cloudbusting Pi

Introduction

<https://meshtastic.org/docs/introduction/>

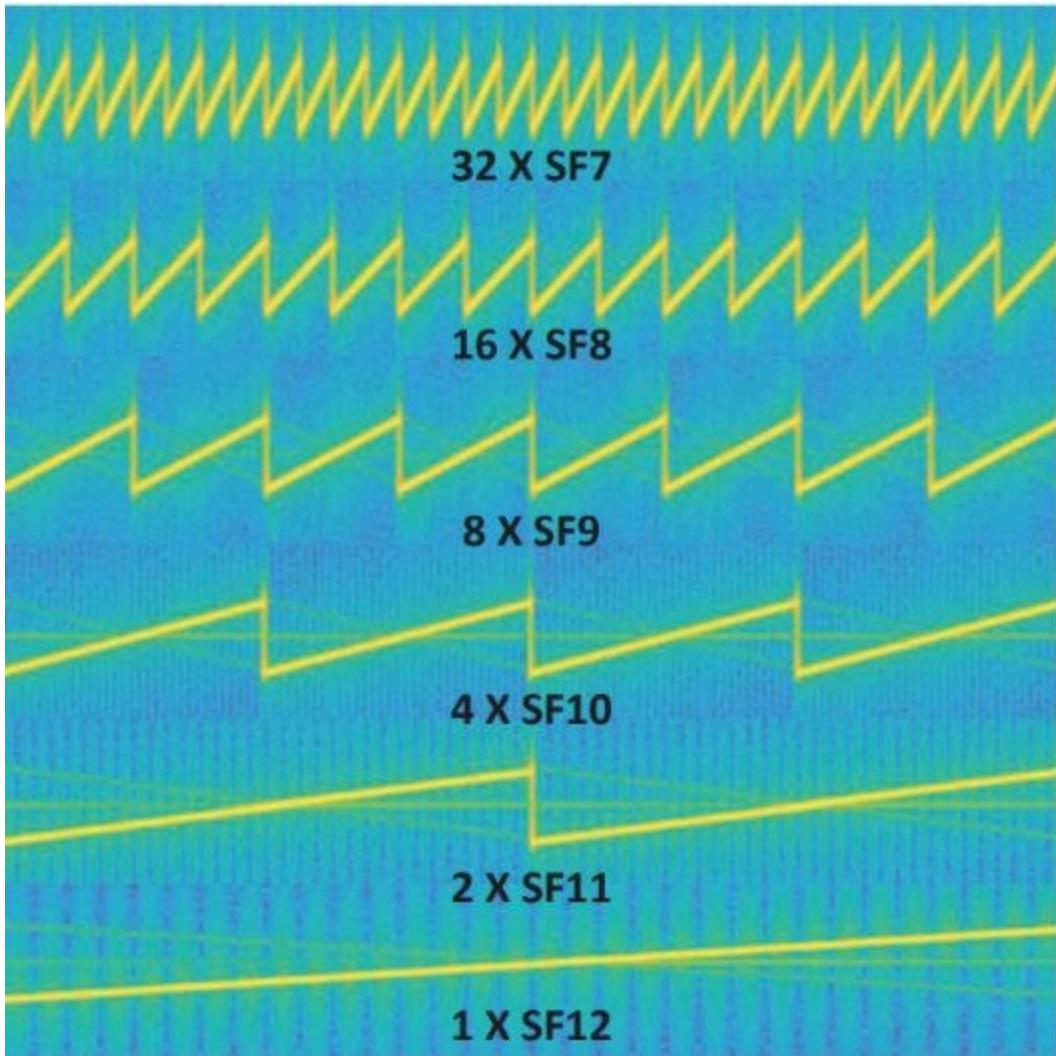
Overview

<https://meshtastic.org/docs/overview/>

- **How does it work?**
- **Underlying infrastructure: What is a mesh?**
 - At the radio level a Meshtastic mesh is a set of nodes that share the same LoRa spreading factor, center frequency, and bandwidth. A node can only be in one radio mesh; it will not see or respond to messages from nodes using different values for these settings. For a mesh to form, nodes need to share the same values.
- **What is LoRa?**
 - LoRa stands for Long Range and is a wireless network protocol that uses radio communication to transmit data over wide areas and bandwidth. LoRa is a type of radio technology, although not like your usual walkie-talkies.
 - LoRa is a wireless modulation technique derived from **Chirp Spread Spectrum (CSS)** technology. It encodes information on radio waves using chirp pulses - similar to the way dolphins and bats communicate! LoRa modulated transmission is robust against disturbances and can be received across great distances. More info:



<https://www.youtube.com/watch?v=dxYY097QNs0>



Ranges of 2-5km in an urban environment or up to 15km in a suburban environment are possible. Over 300km have been seen in open air, peak to peak comms

- Common freqs: 915 MHz, 868 MHz, and 433 MHz
- **Just to note: Meshtastic is different from LoRaWAN**
- **Layered on top: What are channels?**
 - By default, you have one primary channel, which is encrypted with a simple known key ("AQ=="), so to use proper encryption, you must change this key first, or create a new channel and share it with the ones you want to communicate with. However, without a channel using the default key, communication with devices still using it will not be possible.

- Nodes can belong to a maximum of 8 Channels in the radio mesh. A custom Channel can be created for use by a specific group. Only nodes configured with the same Channel name and encryption key will be able to read and display messages on that Channel. However, all nodes in the radio mesh will receive and may retransmit messages (depending on their Role) regardless of the Channel settings for the message.
- **You can select a channel preset depending on your application**
 - long range (IoT sensors), high speed (real-time data), or maximum reliability (critical systems)
 - [Meshtastic pre-set selections](https://meshtastic.org/docs/overview/mesh-tastic-pre-set-selections/)
- **Range**
 - <https://meshtastic.org/docs/overview/range-tests/>
- **Ham mode (unlocks more transmission power, but encryption is not allowed)**
- **What frequencies are used per region?**
 - <https://meshtastic.org/docs/overview/radio-settings/>
 - <https://meshtastic.org/docs/configuration/region-by-country/>

Getting Started

<https://meshtastic.org/docs/getting-started/>

- **Install drivers**
 - <https://meshtastic.org/docs/getting-started/serial-drivers/esp32/>
- **Flash firmware**
 - <https://meshtastic.org/docs/getting-started/flashing-firmware/esp32/web-flasher/>
- **Software**
 - <https://meshtastic.org/docs/software/>

Configuration

<https://meshtastic.org/docs/getting-started/initial-config/>

<https://meshtastic.org/docs/configuration/>

- **Radio config**
 - <https://meshtastic.org/docs/configuration/radio/>
 - Channels: <https://meshtastic.org/docs/configuration/radio/channels/>
 - Device (and roles): <https://meshtastic.org/docs/configuration/radio/device/>
 - LoRa: <https://meshtastic.org/docs/configuration/radio/lora/>
 - Network: <https://meshtastic.org/docs/configuration/radio/network/>
 - Position (GPS): <https://meshtastic.org/docs/configuration/radio/position/>
 - User: <https://meshtastic.org/docs/configuration/radio/user/>

- **Tips**

- <https://meshtastic.org/docs/configuration/tips/>

Software: <https://meshtastic.org/docs/software/>

- **Client App**

- <https://meshtastic.org/#get-connected>

- **Web: <http://meshtastic.local>**

- configure your device to join your wifi network, then in a browser open up <http://meshtastic.local>

MQTT Installation + Configuration

- **Module (MQTT) config**

- <https://meshtastic.org/docs/configuration/module/mqtt/>

- **Example with default mqtt broker**

- Radio Configuration >> Module Configuration >> MQTT
 - MQTT enabled: true
 - Address: <mqtt.meshtastic.org>
 - u: meshdev
 - p: large4cats
 - Encryption enabled: true
 - JSON output enabled: true
 - Root topic: msh/EU_868
 - Proxy to client enabled: true (or connected device to wifi network)

- Channels

- channel 0 name: LongFast
- PSK: AQ==
- on this primary channel, set Uplink and Downlink to "enabled"

- LoRa Config

- Ignore MQTT: false
- OK to MQTT: true

- **Try it out**

- **Receive**

- `mosquitto_sub -h mqtt.meshtastic.org -t msh/EU_868/2/json/LongFast/!da64adac`

- **Send (convert your hex Node ID to decimal)**

- `mosquitto_pub -h mqtt.meshtastic.org -t msh/EU_868/2/json/LongFast/ -m '{"from":3664031148,"channel":1,"type":"sendtext","payload":"Hallo from Edinburgh"}'`

- **Configure your own MQTT broker on your yunohost pi (from cloudbusting) or laptop**

<https://meshtastic.org/docs/software/integrations/mqtt/>

- **Install on your yunohost pi (from Cloudbusting)**

- Install mosquitto from applications dashboard
 - or cli: sudo yunohost app install mosquitto
- Set a username and password
 - u: mosquitto
 - p: dweb2025mqtt
- SSH into your pi and sudo nano /etc/mosquitto/conf.d/default.conf
 - change allow_anonymous from false to true
 - sudo systemctl restart mosquitto

- **Install on your mac**

- <https://meshtastic.org/docs/software/integrations/mqtt/mosquitto/>
- brew install mosquitto

- **Configure the custom broker and channel on your Meshtastic device**

- Radio Configuration >> Module Configuration >> MQTT
 - Address = <The publicly reachable URL of your MQTT broker>
 - u: mosquitto (taken from the example "Install on your yunohost pi" above)
 - p: dweb2025mqtt (taken from the example "Install on your yunohost pi" above)
 - Root topic = msh/dweb2025 (you can create your own topic after "msh/")
- Channels
 - channel 0 name: mqtt
 - PSK: AQ==
 - on this primary channel, set Uplink and Downlink to "enabled"

- **Try it out**

- **Receive**

- mosquitto_sub -h <URL of MQTT broker> -t msh/<your topic>/2/json/mqtt\!da64adac

- **Send (convert your hex Node ID to decimal)**

- mosquitto_pub -h <URL of MQTT broker> -t msh/<your topic>/2/json/mqtt/ -m


```
'{"from":3664031148,"channel":0,"type":"sendtext","payload":"Hallo from Edinburgh"}'
```

Further Notes

Connecting it all together

create python script that takes mqtt messages and pushes them to web page

<https://meshtastic.org/docs/software/integrations/mqtt/mqtt-python/>

<https://techtutorialsx.com/2017/04/23/python-subscribing-to-mqtt-topic/>

<https://hivemq.github.io/mqtt-cli/docs/publish/>

<https://github.com/arankwende/meshtastic-mqtt-client>

<https://www.rapidtables.com/convert/number/decimal-to-hex.html?x=4201583640>

<https://testclient-cloud.mqtt.cool/>

https://meshtastic.au/wp/?page_id=47

<https://docs.rakwireless.com/product-categories/meshtastic/meshtastic-basic-device-setup/>

Local Communication

Each group, using Meshtastic devices, can communicate freely within their local mesh network. This means that devices in Group A (located in one part of the world or far community) can send messages to each other without any internet or Wi-Fi connection. The same applies to Group B in another location.

Gateway Device

To connect these two groups, a Meshtastic device/s (More than one device can act as a gateway for redundancy) within each group can be designated as a gateway. This gateway device will have an additional connection to the internet or a local network capable of running an MQTT broker.

MQTT Integration

The gateway device in Group A connects to an MQTT broker (a server that routes messages between MQTT clients). This broker can be hosted on a local server or a cloud service. The gateway device publishes messages from the local mesh network to a specific MQTT topic. Similarly, it subscribes to another topic to receive messages from the MQTT broker.

Message Relay

The MQTT broker acts as an intermediary. When the gateway device in Group A publishes a message to the MQTT topic, the MQTT broker receives it and then forwards it to the gateway device in Group B, and vice versa. This allows messages to be relayed between the two groups despite the vast distances separating them.

Global Communication

Through this setup, Group A and Group B can now communicate with each other as if they were part of the same local network. The use of MQTT as a bridge ensures that messages are efficiently transmitted between the two distant groups, leveraging the strengths of both Meshtastic for local communication and MQTT for long-range message relaying, creating a versatile and powerful communication network.