Setting Up a Meshtastic MQTT Gateway

Any Meshtastic device with a direct internet connection can act as an MQTT gateway. This includes ESP32-based boards with onboard Wi-Fi (such as the LILYGO T-Beam or Heltec WiFi LoRa 32) and the nRF52-based RAK4631 paired with the RAK13800 Ethernet module. A node does *not* have to provide its own internet connection: with **MQTT Client Proxy**, any node (including nRF52 boards) can reach the broker through the connected phone app's internet connection. Note that nRF52 boards (e.g., RAK4631) cannot emit JSON over MQTT - JSON output requires an ESP32 gateway.

This guide covers enabling MQTT through the official Meshtastic app and verifying the connection with MQTT Explorer.

Prerequisites

- A Meshtastic node with a path to the internet: onboard Wi-Fi (ESP32), Ethernet (RAK4631 + RAK13800), or the phone's connection via MQTT Client Proxy.
- The Meshtastic Android / iOS app, or the Meshtastic Web Client.
- For an on-device internet connection, the node already joined to a Wi-Fi network (configure under *Network settings*: Settings > Network in the app, or Radio Config > Network on Web). With MQTT Client Proxy you can skip on-device Wi-Fi entirely.
- An MQTT broker - either the public `mqtt.meshtastic.org` or your own Mosquitto instance. The public broker is **not anonymous**: it requires credentials (default username `meshdev` / password `large4cats`), which the firmware auto-fills when the fields are left blank.

Step 1 - Connect the Node to Wi-Fi

In the Meshtastic app, open the device's config:

1. Navigate to *Network settings* (Settings > Network in the app, or Radio Config > Network on Web).
2. Toggle *Wi-Fi Enabled* on.
3. Enter your SSID and WPA2 passphrase.
4. Tap *Send*. The device will reboot and connect to Wi-Fi; confirm by checking the IP address shown on the device's OLED display or in the app's *Node Info* panel.

Alternatively, skip on-device Wi-Fi and use **MQTT Client Proxy** so the node connects to the broker through the connected phone app's internet connection.

Step 2 - Enable the MQTT Module

1. In the app, open *Settings* > *MQTT* (on Apple: *Settings* > *Module Configuration* > *MQTT*).
2. Toggle **MQTT Enabled** on.
3. Set **MQTT Server Address**:
   - Public broker: `mqtt.meshtastic.org`
   - Self-hosted: `192.168.1.50` or `mqtt.mynetwork.local`
4. For the public broker, leave **Username** and **Password** blank - the firmware will automatically use the built-in default credentials (`meshdev` / `large4cats`). This is **not** anonymous access. For an authenticated self-hosted broker, fill in your own credentials.
5. Set **Root Topic** - this prefixes all published/subscribed topics. Default: `msh`. Change only if you are segregating traffic (e.g., a private regional prefix like `msh/myregion`).
6. Toggle **JSON Output Enabled** if you intend to consume messages from Node-RED, Home Assistant, or custom scripts. Leave off for device-to-device bridging. (JSON is not supported on nRF52 gateways.)
7. Toggle **TLS Enabled** on if connecting to port 8883 (required for TLS brokers).
8. Tap *Send*. The device will apply the config and initiate MQTT connection.

Note: **Uplink Enabled** and **Downlink Enabled** are *per-channel* settings (configured under *Settings* > *Channels*), not toggles on the MQTT module screen. They default to *off*. See "Advanced: Uplink/Downlink per Channel" below - MQTT only carries traffic when the module is enabled *and* a channel has uplink (or downlink) enabled.

Step 3 - Configure Channel Encryption Key Handling

Messages are encrypted with the channel key on the LoRa air interface. The same encrypted payload is published to MQTT. The gateway does *not* decrypt messages before uplinking - it forwards the ciphertext as-is. This means:

- Subscribers on the MQTT broker who do not know the channel key see only opaque binary blobs.
- Remote Meshtastic devices with the same channel key can decrypt messages downlinked from the broker.
- For JSON mode, the gateway *does* decrypt before serialising to JSON. JSON output therefore exposes plaintext to whoever can subscribe on the broker - use a private broker with ACLs for JSON mode. Note that JSON only serializes a subset of packet types, is **not**

available on nRF52 boards, and is published *alongside* (not instead of) the encrypted protobuf topic - enabling JSON does not stop the encrypted `/e/` topic.

To share a channel key with a remote user, use the channel QR code or URL share feature in the app: *Channels → Share* produces a deep link containing the encoded key.

Step 4 - Verify with MQTT Explorer

MQTT Explorer is a free, cross-platform GUI client for exploring broker traffic.

1. Download from mqtt-explorer.com.
2. Create a new connection:
   - Protocol: `mqtt://` (port 1883) or `mqtts://` (port 8883).
   - Host: `mqtt.meshtastic.org` (or your broker).
   - For the public broker, enter Username `meshdev` and Password `large4cats`. These are required - unlike the Meshtastic firmware, a third-party client such as MQTT Explorer will be rejected by the broker if they are left blank.
3. Connect and expand the `msh` topic tree. Within seconds you should see topics appearing as your gateway uplinks packets.
4. Click a leaf topic to see the raw payload (raw/binary protobuf, which some clients display as base64) and, for JSON topics, the decoded JSON object.

```
# Alternatively, use the mosquitto_sub CLI tool:
mosquitto_sub -h mqtt.meshtastic.org -p 1883 -u meshdev -P large4cats -t "msh/#" -v
```

Advanced: Uplink/Downlink per Channel

Meshtastic 2.x allows per-channel MQTT uplink/downlink settings, so you can uplink a private community channel while keeping a sensitive channel (e.g., encrypted tactical) RF-only:

1. In the app, navigate to *Settings* > *Channels*.
2. Edit the channel you want to configure.
3. Under *Uplink Enabled* / *Downlink Enabled*, set the per-channel overrides. These default to *off* for all channels.
4. MQTT requires *both* the MQTT module to be enabled *and* at least one channel to have uplink (or downlink) enabled. The module toggle alone does not uplink any channel - if the module is on but no channel is enabled, nothing flows.

Common Issues and Fixes

Symptom	Cause	Fix
Device shows "MQTT connecting" indefinitely	Wrong broker address or firewall blocking port 1883	Ping broker from device's LAN; check firewall rules
No topics appear in MQTT Explorer	Uplink not enabled, or no LoRa traffic to uplink	Confirm Uplink is on; transmit a message from another node
Remote nodes not seen locally after downlink	Channel key mismatch between islands	Share channel URL; confirm both ends use identical key bytes
Duplicate messages on the mesh	Multiple gateways have downlink enabled in same RF area	Disable downlink on all but one gateway per coverage area
TLS handshake failure	Clock skew on device; certificate expired	Enable NTP on node; renew/replace TLS cert on broker

Revision #6
Created 2026-05-03 05:36:31 UTC by Mesh America Admin
Updated 2026-06-10 05:38:26 UTC by Mesh America Admin