

Meshtastic Repeater Setup

Step-by-step setup, role selection, and channel configuration for Meshtastic repeaters.

- [Router vs. Repeater Role — Which to Choose](#)
- [Initial Setup Walkthrough](#)
- [Channel Configuration for Infrastructure Nodes](#)

Router vs. Repeater Role — Which to Choose

Overview of Device Roles

Meshtastic firmware supports several device roles that control how a node behaves on the mesh. For infrastructure nodes the historically relevant roles are **ROUTER** and **REPEATER**, but note that **REPEATER is deprecated as of firmware ~2.7.x**. For most stationary, well-placed infrastructure the current recommendation is **ROUTER** (or **ROUTER_LATE** for a node that should rebroadcast only after others have had a chance); for the overwhelming majority of ordinary nodes the official guidance is simply **CLIENT**.

ROUTER Role

- Always rebroadcasts each packet once (does not defer to a neighbor that rebroadcasts first), with prioritized routing. The scope of what it rebroadcasts is governed by `rebroadcast_mode` (default ALL), not by the role itself.
- Appears in the app node list of nodes that hear it. It will only appear on meshmap.net if its packets reach the internet through an MQTT gateway somewhere on the mesh - merely being a ROUTER does not put it on the public map.
- At the protocol level a ROUTER can still originate and receive messages, but BLE/WiFi/Serial app connectivity is OFF by default, so you do not normally text it directly to check status without first enabling an interface.
- Uses more airtime than deferring roles because it always rebroadcasts (it never skips when a neighbor already did), not because it sends extra copies of a packet.
- Rebroadcasts with priority by using a shorter contention delay ("cutting in line") so it relays before other nodes, which helps extend range. It does not prioritize by remaining hop count.
- Automatically enables power-saving sleep on ESP32 (this cannot be turned off) and is designed for stationary, well-placed nodes.
- **Good for:** permanent, strategically placed stationary nodes acting as backbone relays; medium-traffic networks where a preferred relay improves reach.

REPEATER Role

REPEATER is deprecated as of firmware ~2.7.x. Prefer ROUTER (or ROUTER_LATE) for new infrastructure. The behavior below is retained for reference and for existing deployments.

- Like ROUTER, it is a preferred relay that always rebroadcasts each packet once with a shorter delay; it does not prioritize by remaining hop count and is not a "hop maximiser."
- Does **not** broadcast its own position or send NodeInfo - it is anonymous on the mesh and does not appear in the node list, which reduces network traffic.
- Is not normally addressed for "self" - you do not text it directly. This is a default-configuration behavior, not the firmware omitting a send code path.
- Does not force power-saving sleep, so it keeps its LoRa radio on. Baseline power draw is therefore similar to ROUTER; differences between the two roles are marginal (self-generated airtime, CPU/RAM) rather than one drawing dramatically less than the other.
- **Good for (legacy):** permanent unattended relay-only infrastructure where the node should stay anonymous on the mesh. For new builds, use ROUTER instead.

When to Use Each Role

Scenario	Recommended Role
Temporary deployment or field relay that doubles as a usable client	CLIENT
Node needs to appear in the node list for coordination	ROUTER
Permanent mountaintop or rooftop installation	ROUTER
Solar-powered, unattended backbone node	ROUTER
Ordinary user node	CLIENT

CLIENT_MUTE Role

CLIENT_MUTE prevents a regular client device (phone, laptop) from rebroadcasting mesh traffic - a normal CLIENT still rebroadcasts under managed flooding, but CLIENT_MUTE does not relay at all. This is useful for devices that are on the mesh but should not consume airtime acting as relays, such as tablets used only for monitoring.

Setting the Role

In the [Meshtastic app](#): on Android, **Settings → Device → Role**; on iOS/macOS, **Settings → Device Configuration → Device → Role**. The exact menu path can vary by app version.

Via the CLI: `meshtastic --set device.role ROUTER` (the key is `device.role` and the value is the bare enum name, e.g. `ROUTER`, `ROUTER_LATE`, `CLIENT`).

Via the web client at client.meshtastic.org: connect via USB, navigate to Device Config, and choose the desired role from the drop-down.

Initial Setup Walkthrough

Prerequisites

- A Meshtastic-compatible device (T-Beam, Heltec, RAK WisBlock, etc.) flashed with current firmware - use the official flasher at flasher.meshtastic.org.
- A phone with the [Meshtastic app](#) (iOS or Android) **or** a Chrome-based browser for the web client.
- An antenna connected *before* powering the device.

Step 0 - Set the Region (do this first)

Config → LoRa → Region - set this to your country (e.g.) before transmitting. The region is the master legal control: it determines both your authorized frequency band *and* the maximum power the firmware will permit. The device should not be used to transmit until the region is set correctly - in the US, only 902-928 MHz is available for these unlicensed Part 15 devices, and an incorrect region will transmit out-of-band on frequencies you are not authorized to use. Leave at its default of so the firmware applies the region-legal maximum.

Step 1 - Connect to the Device

Via Bluetooth: Open the Meshtastic app, tap the + button, and scan for your device. Pair when prompted.

Via USB: Navigate to client.meshtastic.org in Chrome, click **New connection → Serial**, and select the device's COM/serial port.

Step 2 - Set the Device Name

Navigate to **Config → Device**:

- **Long Name:** Your full identifier for this node, e.g. `KD9XYZ-Hilltop-1`. This appears in message headers and node lists.
- **Short Name:** Up to 4 bytes (typically 4 ASCII characters), shown on the map, e.g. `H1`. Note that multi-byte UTF-8 characters such as emoji or non-ASCII text consume more than one byte each, so fewer than 4 will fit.

Step 3 - Set the Device Role

Config → Device → Role. Use **ROUTER** when you want the infrastructure node to stay visible in the Nodes list and topology (the recommended choice for most infrastructure); use **REPEATER** for a pure relay that is invisible in the Nodes list and originates no NodeInfo, Position, or Telemetry. Note that **REPEATER is deprecated as of firmware ~2.7.x**, so ROUTER is preferred for new deployments; for ordinary non-infrastructure nodes the official guidance is CLIENT. Both ROUTER and REPEATER are infrastructure relay roles - neither is specifically a "monitoring point." See the *Router vs. Repeater Role* page for guidance.

Step 4 - Configure the Channel

The default channel (*LongFast* or *Default*) works for joining the public mesh. To match a local community's private channel:

1. Navigate to **Channels → Channel 0**.
2. Set the **Name** and **PSK** to match the local standard. The **PSK** (pre-shared key) is the channel's encryption key - all nodes that share a channel must use the same key to read each other's messages. Your local community provides this key, usually as a shared QR code or channel URL that imports the name and key together; you rarely need to type a raw key by hand. A PSK can be 0 bytes (no encryption), 16 bytes (AES-128), or 32 bytes (AES-256). See the channel-configuration page for details.
3. Contact your local Meshtastic community (Discord, or a local club) for the channel name and key.

Step 5 - Set the Modem Preset

Radio Config → LoRa → Modem Preset - select the preset your local network uses (typically *Long Fast* or *Medium Slow*). **Critical: all nodes on the same network must use the same modem preset or they cannot hear each other.**

Step 6 - Configure a Fixed Position

For an unattended repeater without GPS, set a fixed position so the node can report its location:

1. **Config → Position → Fixed Position → Enable.**
2. Enter the latitude, longitude, and altitude of the deployment site (look up coordinates with any map app).
3. Set **Position Broadcast Interval** to a long value for a static node - `43200` seconds (12 hours) or `86400` seconds (24 hours) is appropriate, since a fixed node's position never changes and frequent rebroadcasts waste airtime.

Step 7 - Power Optimisations (Battery / Solar)

- **Config → Power** - set sleep and minimum wake intervals to the lowest practical values.
- Automatic forced power-saving is tied to the **ROUTER** role (ESP32 only), not REPEATER - the firmware enables it for ROUTER and it cannot be turned off. The REPEATER role does not force power-saving sleep; tune power settings manually where the platform supports it.
- **Screen Timeout** (`display.screen_on_secs`) controls how long the display stays on after activity. Lowering it reduces draw, but note that a value of `0` selects the firmware default (about 10 minutes), *not* "always off" - set a small non-zero value if you want the screen to turn off quickly.

Step 8 - Disable Bluetooth (Optional)

Config → Bluetooth → Enabled → false. This saves power and removes an attack surface on unattended nodes. **Note:** once Bluetooth is disabled, you will need a USB/serial connection or remote admin (PKC admin keys, firmware ≥ 2.5) to re-enable it.

Step 9 - Verify Operation

- Watch the node list for other nearby nodes appearing - this confirms receive is working.

- Send a test message and verify it is received on another device.
- If the node is uplinked to MQTT through an internet gateway, it may also appear on the third-party community map meshmap.net. Note that meshmap.net is not an official Meshtastic property and only shows nodes whose data reaches it via an MQTT gateway - a node with no internet gateway may never appear there, and appearance timing is not guaranteed.

Channel Configuration for Infrastructure Nodes

What Are Channels?

Meshtastic supports up to **8 simultaneous channels** (numbered 0 - 7). Channel 0 is the *primary channel* used for most mesh traffic. Channels 1 - 7 can carry separately encrypted traffic for specific groups or purposes.

PSK - Pre-Shared Key

Each channel has a **name** and a **pre-shared key (PSK)**. The PSK can be **0 bytes** (no encryption), **16 bytes** (AES-128), or **32 bytes** (AES-256) - it is not always a 32-byte AES-256 key. Two nodes can communicate on a channel only if they share the same name *and* the same PSK (same key length and value). The PSK encrypts the channel payloads with AES (AES-128 for a 16-byte key, AES-256 for a 32-byte key).

The Default Public Key

The *Default* public channel (LongFast) uses a **well-known, publicly published default key** (the 1-byte key `AQ==`, 0x01) - it is a defined weak key, not an empty/no-encryption PSK, so traffic on it is effectively public. Any node running Meshtastic with the default channel can participate in the public mesh. Using a **custom PSK** creates a private channel readable only by nodes that hold that key.

Channel Strategy for Community Infrastructure Nodes

- **Channel 0 - Default PSK:** Keep the primary channel on the public Default key so all community users benefit from your repeater's coverage.
- **Channel 1 - Private PSK:** Adding a secondary channel with a private key for your personal use or club coordination is acceptable. The repeater will relay packets on both channels.

Remote Administration

Meshtastic supports remote administration so that nodes can send configuration commands - changing settings without physical access to the device. In firmware 2.5 and later, the recommended method is **PKC admin keys** configured under **Security Config**: you add the public key(s) of trusted administrator nodes to the remote node's admin-key list, and those nodes can then administer it. The legacy **admin channel** (a secondary channel named exactly `admin`, case-sensitive) exists only for managing pre-2.5 nodes; there is *no* "Is Admin" toggle on a channel.

Setting up remote administration is strongly recommended for unattended permanent deployments.

1. On firmware 2.5+, open **Security Config** and add the public key of each trusted administrator node to the node's admin-key list. (For legacy pre-2.5 nodes only: create a secondary channel named exactly `admin`.)
2. Verify the exact app path before you publish or commit the change - getting remote admin wrong can lock you out of a remote node entirely.
3. Back up the configuration (export the config, and for legacy admin channels save the QR code) and store it securely - anyone with the admin keys or legacy admin-channel QR code can reconfigure your node remotely.

Channel Propagation

What a relay rebroadcasts depends on its **Rebroadcast Mode**, not just on which channels it has configured. Meshtastic relays based on the *unencrypted* packet header, so under the default `ALL` mode a repeater rebroadcasts **all** packets that match its modem settings and frequency - *including packets on channels whose PSK it does not have and cannot decrypt*. This means a private channel can be carried by a public repeater. Only the `LOCAL_ONLY` or `KNOWN_ONLY` rebroadcast modes restrict relaying to the repeater's own configured/known channels. (A node still has to be on the same modem preset and frequency to hear and relay at all.) For a public community repeater, keeping only the public Default channel configured is the standard approach, but be aware that under default `ALL` mode it will still relay other traffic on the same modem settings.

Changing Channels on a Deployed Node

Options for modifying channel config after deployment:

- **Via remote admin** (preferred for remote nodes) - send a channel update from an administrator node whose key is in the remote node's admin-key list (or, on legacy nodes, that shares the admin channel).
- **Via serial/USB** - connect a laptop directly to the node.
- **Via Bluetooth** - only if Bluetooth was left enabled.

Common pitfall: losing the admin keys / admin channel config (or disabling Bluetooth and having no USB access) leaves a remote node inaccessible without a physical site visit. Always back up admin keys and any admin-channel QR codes before deployment.