

Configuration

Setting device role, rebroadcast mode, and power optimization.

- [Setting the Device Role](#)
- [Rebroadcast Mode: ALL_SKIP_DECODING](#)
- [Power Optimization Settings](#)
- [Choosing a Modem Preset](#)

Setting the Device Role

Configuring a device as a Meshtastic repeater involves two key settings: the device role and the rebroadcast mode. Both can be set using the Meshtastic mobile app or the Python CLI.

Using the mobile app (Android/iOS)

1. Connect to your device in the [Meshtastic app](#)
2. Navigate to **Config** → **Device** → **Role** (app layouts change between versions; this is the current path)
3. Find **Device Role** and select your role. Note that **REPEATER is deprecated as of firmware ~2.7.x**; for most infrastructure nodes the recommended role is now **Router** (which stays visible in the node list), and for the overwhelming majority of ordinary nodes the official guidance is **Client**.
4. Save settings - the device will reboot to apply changes

Using the Python CLI

Install the CLI if you have not already:

```
pip install meshtastic
```

Connect the device via USB, then set the role. The setting key is `device.role` and the value is the bare enum name (for example `ROUTER` or `REPEATER`):

```
# REPEATER is deprecated as of firmware ~2.7.x; ROUTER is recommended for most infrastructure
meshtastic --set device.role ROUTER

# To set the (deprecated) REPEATER role explicitly:
meshtastic --set device.role REPEATER
```

Verify the change:

```
meshtastic --info
```

After setting the role

Once set to REPEATER, the device will no longer appear in other nodes' node lists and does not send NodeInfo (it is anonymous on the mesh) - this is by design. It relays packets without broadcasting its own data, and unlike ROUTER it does not force power-saving sleep. To confirm it is operating, check the serial output or observe that messages are being relayed through it. (If you instead choose ROUTER, the node *does* appear in the node list and the firmware automatically enables power-saving sleep that cannot be turned off, with BLE/WiFi/Serial off by default.)

Flashing Meshtastic firmware

If your device does not have Meshtastic installed, use the [Meshtastic Web Flasher](#) to install firmware directly from Chrome or Edge - no software installation required. Connect via USB, select your device type, and click Flash.

Setting the region

Before the radio will transmit, you must set the correct region. The region setting is the master legal control - it determines both your authorized frequency band *and* the maximum power the firmware will permit:

```
meshtastic --set lora.region US
```

Or in the app: **Config → LoRa → Region → US**. Setting the wrong region (for example leaving a US deployment on EU868) will cause the device to transmit out-of-band on frequencies you are not authorized to use - a clear violation of FCC rules, not just a possibility. In the US, only 902-928 MHz is available for these unlicensed Part 15 devices. Always set the region that matches your physical location before transmitting, and leave `lora.tx_power` at its default of `0` so the firmware applies the region-legal maximum.

Rebroadcast Mode:

ALL_SKIP_DECODING

The rebroadcast mode controls how a node handles incoming packets for retransmission. For repeater deployments, `ALL_SKIP_DECODING` is strongly recommended. Note that `ALL_SKIP_DECODING` is only available when the device role is REPEATER - on any other role (including ROUTER) the setting behaves as `ALL`. If your node uses the ROUTER role, use rebroadcast mode `ALL` instead.

What ALL_SKIP_DECODING does

In this mode, the node rebroadcasts valid LoRa packets without attempting to decode or decrypt them. This mode is only available when `device.role` is REPEATER. It has several important advantages:

- **Lower CPU usage and power draw** - the node skips the packet-decoding/duplicate-check processing entirely, and it never decrypts or re-encrypts payloads
- **Relays channels whose key it does not have** - Meshtastic relays based on the unencrypted packet header, so a repeater forwards packets on any channel that shares its modem settings (preset/frequency), even private channels whose pre-shared key (PSK) - the encryption key that secures a private channel - the repeater does not hold. This lets one public repeater carry traffic for multiple user groups, provided they are all on the same modem preset.
- **Preserves end-to-end encryption** - messages are never decrypted at the repeater; the original encrypted packet is forwarded unchanged, maintaining the security model
- **Faster retransmission** - less processing time per packet means faster relay

Setting ALL_SKIP_DECODING via app

1. Connect to your device in the [Meshtastic app](#)
2. Make sure the device role is set to REPEATER first - the `ALL_SKIP_DECODING` option only applies on the REPEATER role

3. Go to **Settings → Device**
4. Find **Rebroadcast Mode** and select **ALL_SKIP_DECODING**
5. Save and allow the device to reboot

Setting ALL_SKIP_DECODING via CLI

```
meshtastic --set device.rebroadcast_mode ALL_SKIP_DECODING
```

This is only valid when `device.role` is REPEATER. The setting lives under the `device.` namespace, not `lora.`

Other rebroadcast modes (for reference)

Mode	Behavior	Use case
<code>ALL_SKIP_DECODING</code>	Rebroadcasts without decoding or decrypting	Recommended for all repeater deployments; requires REPEATER role
<code>ALL</code>	Processes/validates packets (including duplicate detection) before rebroadcasting the original encrypted packet; does not decrypt or re-encrypt payloads	Default; used on ROUTER and other roles that should relay
<code>LOCAL_ONLY</code>	Does not relay to other nodes	Not suitable for repeaters
<code>NONE</code>	No rebroadcasting (only permitted for SENSOR/TRACKER/TAK_TRACKER roles)	Not suitable for repeaters

Power Optimization Settings

Repeater and Router role nodes keep the LoRa radio on continuously, which draws significantly more power than a client device that sleeps between uses. These settings minimize everything else to extend runtime on battery or solar power.

Disable Bluetooth

Once configured, a dedicated repeater does not need Bluetooth. Disabling it saves power and removes an unnecessary wireless interface.

```
meshtastic --set bluetooth.enabled false
```

Disable GPS (REPEATER role handles this automatically)

The REPEATER role does not broadcast position data, but the GPS module may still draw power unless you disable it explicitly in the Position config. If your board has a GPS module, disable it to save power:

```
meshtastic --set position.gps_mode DISABLED
```

Disable the screen

If your device has a display, minimize the screen-on time. Display backlights draw meaningful power. Note that `display.screen_on_secs 0` does **not** turn the screen off - 0 means the 10-minute default. To minimize screen-on time, set a small non-zero value instead:

```
meshtastic --set display.screen_on_secs 1
```

Set appropriate transmit power

Transmit power is the largest variable power draw. In most cases, leave `tx_power` at its default of **0**, which tells the firmware to use the maximum power your region legally permits for this hardware - the region setting is what enforces the legal cap. Only set an explicit value if you are deliberately reducing power (for example, to limit interference with nearby nodes). Higher power is not always better.

```
meshtastic --set lora.tx_power 0
```

If you do set an explicit value, `tx_power` is conducted PA power; stay at or below 30 dBm. If you use an antenna over 6 dBi gain, you **MUST** reduce conducted power by 1 dB for every dB of gain above 6 dBi (47 CFR 15.247(b)(4)). Actual radiated power also depends on your board's PA capability and feedline loss, so tie any fixed value to your antenna-gain / EIRP calculation rather than picking a number arbitrarily.

Use the right modem preset - but match your local network

Modem preset affects how long the radio is transmitting each packet, which directly impacts power consumption:

- Slower, longer-range presets (Long Slow, Long Moderate) keep the radio transmitting longer per packet - higher average power draw
- Faster presets (Medium Slow, Medium Fast) transmit each packet more quickly - lower average power draw, and better network capacity in dense areas
- Long Fast (firmware default) sits in the middle

Do not choose a preset based on power alone - you must use the same preset as the rest of your local network. Check what preset your local community or regional network uses before deploying. See the [Choosing a Modem Preset](#) page for guidance.

Managed Mode (optional)

For a deployed repeater that should not be reconfigured by whoever is near it, you can enable Managed Mode (`security.is_managed`). This blocks client apps from writing config; changes must then come via PKC Remote Admin (firmware 2.5+) or the legacy Admin channel. **Verify that**

remote admin works before enabling this, or you risk locking yourself out - once managed mode is on, you cannot reconfigure the node from a normal client app.

```
meshtastic --set security.is_managed true
```

Choosing a Modem Preset

Your modem preset is one of the most consequential configuration decisions for your repeater. It must match every other node on your channel - and different regional communities have standardized on different presets.

What a preset controls

Each preset is a named combination of three LoRa parameters:

- **Spreading Factor (SF):** Higher SF = longer range, longer airtime per packet. Range from 7 (fastest) to 12 (longest range).
- **Bandwidth (BW):** Wider bandwidth = faster data rate, slightly less range.
- **Coding Rate (CR):** Higher CR adds forward error correction overhead.

The full preset table

Preset	SF	BW	CR	Data Rate	Link Budget
Short Turbo	7	500 kHz	4/5	21.9 kbps	140 dB
Short Fast	7	250 kHz	4/5	10.9 kbps	143 dB
Short Slow	8	250 kHz	4/5	6.25 kbps	145.5 dB
Medium Fast	9	250 kHz	4/5	3.52 kbps	148 dB
Medium Slow	10	250 kHz	4/5	1.95 kbps	150.5 dB
Long Fast	11	250 kHz	4/5	1.07 kbps	153 dB
Long Moderate	11	125 kHz	4/8	0.34 kbps	156 dB
Long Slow	12	125 kHz	4/8	0.18 kbps	158.5 dB

The link-budget figures above assume 22 dBm TX power and 0 dBi antennas; your real budget shifts with your actual transmit power and antenna gain. `VERY_LONG_SLOW` (SF12 at the narrowest bandwidth) is the only preset offering even more range than Long Slow, but it is deprecated and not recommended by the Meshtastic project for regular use.

Link budget is roughly the maximum single-hop path loss the signal can survive, computed at 22 dBm TX and 0 dBi antennas; real budgets shift with your actual TX power and antenna gain. Higher

is more range. Data rate is the raw channel throughput - faster data rate means each transmission occupies less airtime, reducing collisions in busy networks.

Step 1: Ask your local community

Before choosing a preset, check what your local or regional network has standardized on. **All nodes on a channel must use the same preset to communicate.** If your area already has an established mesh, deploying on a different preset will isolate your repeater from it entirely.

Check local Discord servers, third-party tools such as the community [network map at meshmap.net](https://meshmap.net) (unofficial; only shows nodes reporting to the public MQTT server), or regional mesh community pages for your area's standard.

Step 2: Match your network density

If no local standard exists, choose based on your expected network density:

Sparse / rural networks (under ~60 nodes in range)

Long Fast (the firmware default) is a reasonable choice. Range is maximized and the lower data rate is not a problem when relatively few nodes are generating traffic.

Larger networks (more than ~60 nodes)

Consider Medium Slow or Medium Fast. The official guidance is to consider moving away from Long Fast once a mesh exceeds roughly 60 nodes. The 3 - 4x higher data rate significantly reduces collision probability while still maintaining respectable, only slightly reduced range compared to Long Fast.

Dense urban networks (well over 60 nodes)

Faster presets are strongly preferred. Long Fast in a 100+ node network can cause congestion as packet airtime accumulates. Medium Slow has been reported as the standard for at least one large community mesh: the Meshtastic Bay Area community (reportedly 150+ nodes, as of 2025) is reported to have migrated to Medium Slow with improved reliability, and the Wellington Region Mesh (New Zealand) is reported to use Short Fast. These community figures are unsourced and change over time - check your local community's current standard rather than relying on these examples.

Presets to avoid for repeater deployment

- **Long Slow / Very Long Slow** - maximum range but extremely long airtime per packet. Can saturate a network even at low message rates. Long Slow and the even slower, deprecated `VERY_LONG_SLOW` are not recommended for regular use by the Meshtastic project.
- **Short Turbo** - highest throughput, but it uses 500 kHz bandwidth. This is legal in the US 902-928 MHz band, but some narrower regional bands (for example EU 868 MHz) do not permit it. Verify legality for your region before use.

Setting the preset via CLI

```
meshtastic --set lora.modem_preset MEDIUM_SLOW
```

Valid preset names: `SHORT_TURBO`, `SHORT_FAST`, `SHORT_SLOW`, `MEDIUM_FAST`, `MEDIUM_SLOW`, `LONG_FAST`, `LONG_MODERATE`, `LONG_SLOW`, `VERY_LONG_SLOW` (`VERY_LONG_SLOW` is deprecated / not recommended).

Frequency slot and channel name

Meshtastic derives the transmit frequency from the channel name hash by default (frequency slot 0). This means two nodes with the same channel name and same preset will automatically land on the same frequency. You can override this with a specific slot number if needed, but the default hash-based behavior is correct for most deployments.