

FAQ

Answers to common questions about LoRa mesh networking, MeshCore, Meshtastic, hardware, range, privacy, and legality.

- [📖 Start Here — FAQ Guide](#)
- [General Questions](#)
- [Hardware and Setup](#)
- [Battery and Power](#)
- [Common Questions](#)
 - [Setup and Configuration Questions](#)
- [Glossary](#)
 - [Glossary of Mesh Networking Terms](#)
- [MeshCore Official FAQ](#)
 - [FAQ: 1. Introduction](#)
 - [FAQ: 2. Initial Setup](#)
 - [FAQ: 3. Server Administration](#)
 - [FAQ: 4. T-Deck Related](#)
 - [FAQ: 5. General](#)
 - [FAQ: 6. Troubleshooting](#)
 - [FAQ: 7. Other Questions:](#)
- [Troubleshooting Guide](#)
 - [My node cannot connect to others](#)
 - [My messages are not getting through](#)
 - [My battery drains too fast](#)
 - [My solar node keeps going offline](#)
- [Common Beginner Questions](#)

- [Do I need an amateur radio license?](#)
- [How far can my node reach?](#)
- [MeshCore or Meshtastic - which should I choose?](#)
- [Hardware FAQ](#)
 - [Which board should I buy as a beginner?](#)
 - [Why isn't my GPS getting a fix?](#)
 - [Can I use my node inside my house or vehicle?](#)
- [Solar and Power FAQ](#)
 - [How big a solar panel do I need?](#)
 - [Why does my solar node keep dying at night?](#)
 - [What battery chemistry should I use outdoors?](#)
- [Networking and Range FAQ](#)
 - [How many hops can a message travel?](#)
 - [Why do I see duplicate messages?](#)
 - [What is channel utilization and why does it matter?](#)
- [Firmware and Software FAQ](#)
 - [How do I update my Meshtastic firmware?](#)
 - [What Meshtastic firmware version should I run?](#)
 - [How do I factory reset my node?](#)
- [Security and Privacy FAQ](#)
 - [Is Meshtastic encrypted? Can anyone read my messages?](#)
 - [Who can see my location on the mesh?](#)
 - [Can I trust MeshCore encryption for sensitive communications?](#)
- [Troubleshooting Advanced Issues](#)
 - [My node is flooding the channel with repeated messages](#)
 - [My Meshtastic node and MeshCore node cannot communicate](#)
 - [RF interference is affecting my node — how do I diagnose it](#)
- [MeshCore-Specific FAQ](#)

- [What is the difference between a Repeater and Room Client in MeshCore?](#)
- [How do I connect to a MeshCore room server from the app?](#)
- [Can I run MeshCore and Meshtastic simultaneously on the same hardware?](#)

- [Community and Social FAQ](#)
 - [How do I find other mesh users in my area?](#)
 - [How do I start a mesh network where there are none?](#)
 - [Can I use Meshtastic or MeshCore for commercial purposes?](#)

- [Antenna and RF FAQ](#)
 - [Do I need an external antenna?](#)
 - [What is the difference between dBi and dBd antenna gain?](#)

Start Here — FAQ Guide

This book answers the most common questions about LoRa mesh networking, MeshCore, and Meshtastic. Use the sections below to jump straight to what you need.

Most Asked Questions

- [MeshCore or Meshtastic - which should I choose?](#)
- [Do I need an amateur radio license?](#)
- [How far can my node reach?](#)
- [Which board should I buy as a beginner?](#)
- [Is Meshtastic encrypted? Can anyone read my messages?](#)

FAQ Categories

Troubleshooting - Something's Not Working

- [My node cannot connect to others](#)
- [My messages are not getting through](#)
- [My battery drains too fast](#)
- [My solar node keeps going offline](#)
- [My node is flooding the channel with repeated messages](#)
- [RF interference is affecting my node - how do I diagnose it?](#)

Hardware Questions

- [Which board should I buy as a beginner?](#)
- [Why isn't my GPS getting a fix?](#)

- [Can I use my node inside my house or vehicle?](#)
- [Do I need an external antenna?](#)

☐☐ Solar and Power Questions

- [How big a solar panel do I need?](#)
- [Why does my solar node keep dying at night?](#)
- [What battery chemistry should I use outdoors?](#)

☐☐ Networking and Range Questions

- [How many hops can a message travel?](#)
- [Why do I see duplicate messages?](#)
- [What is channel utilization and why does it matter?](#)

☐☐ Security and Privacy Questions

- [Is Meshtastic encrypted? Can anyone read my messages?](#)
- [Who can see my location on the mesh?](#)
- [Can I trust MeshCore encryption for sensitive communications?](#)

☐☐ Community Questions

- [How do I find other mesh users in my area?](#)
- [How do I start a mesh network where there are none?](#)
- [Can I use Meshtastic or MeshCore for commercial purposes?](#)

☐☐ MeshCore-Specific Questions

- [What is the difference between a Repeater and Room Client in MeshCore?](#)
- [How do I connect to a MeshCore room server from the app?](#)

☐☐ Firmware and Software Questions

- [How do I update my Meshtastic firmware?](#)
- [What Meshtastic firmware version should I run?](#)
- [How do I factory reset my node?](#)

Full Reference

- [Glossary of Mesh Networking Terms](#) - Definitions for every term used in this wiki
- [MeshCore Official FAQ](#) - Adapted from official MeshCore documentation; some entries have been edited or paraphrased and may not match the current upstream docs verbatim. Check the official MeshCore docs for the authoritative version.

General Questions

Answers to the most common general questions about LoRa mesh networking, MeshCore, and Meshtastic.

Do I need a license to use Meshtastic or MeshCore?

No. Both protocols operate in the 902-928 MHz band in the US and Canada, which is license-free for compliant unlicensed devices - in the US under FCC Part 15, and in Canada under ISED's RSS-247 rules. This is the band commonly called the "915 MHz ISM band," but note that it is a *shared* band: the unlicensed mesh use is a secondary, non-interference allocation, not an exclusive ISM allocation. You do not need an amateur (ham) radio license to buy, own, or operate a LoRa mesh node for MeshCore or Meshtastic.

The FCC Part 15 rules that apply (47 CFR 15.247(b)(3) and (b)(4)): maximum 1 W (30 dBm) conducted transmit power. Antenna gain up to 6 dBi is allowed at full power; above 6 dBi, conducted power must be reduced dB-for-dB, so EIRP stays at roughly 36 dBm. There is no standalone "4 W EIRP" allowance that lets you run full power into a high-gain antenna. Mass-market LoRa boards ship at FCC-compliant default power, but some hardware (external PA modules, Station G2) can be configured above legal limits - verify your settings.

Note: if you are a licensed amateur radio operator and choose to operate on amateur frequencies using Part 97 rules, different rules apply - but standard ISM band operation requires no license at all.

Does this work without internet or cell service?

Yes - the radio layer needs no internet or cell service. That is the entire point of LoRa mesh networking. The mesh operates entirely on LoRa radio signals transmitted directly between devices. There is no internet connection, no cell network, no infrastructure, and no central server

involved in basic mesh operation.

Each node talks directly to nearby nodes via radio. Messages hop from node to node until they reach the destination. As long as there is a path of nodes in range of each other between sender and receiver, messages get through - regardless of whether any internet or cell service exists.

Important caveat: the mesh only works if other nodes are within radio range, either directly or via repeaters. A single node with no peers in range talks to no one. In an area with no nearby nodes - or in a wide-area disaster where nodes lose power and paths collapse - an off-grid node may have no working path. Build and verify local coverage before relying on the mesh for critical communications.

Some users optionally connect gateway nodes to the internet via Wi-Fi (primarily for Meshtastic's MQTT bridging feature), but this is entirely optional and not required for the mesh to function.

How far can messages travel?

Range depends heavily on environment, antenna height, and channel settings. The figures below are approximate field observations, not manufacturer guarantees, and real-world results vary widely:

- **Urban, near ground level (node to node):** roughly 1 - 5 km - highly site-dependent; dense urban can be well under 1 km
- **Rural, open terrain (node to node):** roughly 5 - 20 km
- **Elevated repeater on a hilltop or tower:** 20 - 50+ km with line of sight. Line-of-sight distance scales with antenna height (roughly the square root of height via the radio-horizon relationship), so these numbers depend on how high the node is, not a fixed figure.

Those are *single-hop* figures. With a mesh network of repeater nodes, a message can travel beyond any single-hop range by relaying across a chain of nodes. In practice, reliable multi-hop range is limited by the firmware's hop limit (Meshtastic defaults to 3, with a hard cap of 7), per-hop latency, and accumulating packet loss, so long cross-region delivery is the exception rather than the rule.

The single most effective way to increase range is elevation. Because line-of-sight distance follows the radio horizon (which scales with the square root of antenna height), a hilltop or tower repeater can reach far-off nodes that a ground-level node cannot reach at all - elevated repeaters commonly reach tens of kilometres to other elevated or distant nodes. (As an illustration, the radio horizon from about 500 ft to ground level is roughly 30 km.)

Can I use MeshCore and Meshtastic devices together on the same network?

No. MeshCore and Meshtastic use different protocols and different packet formats. A MeshCore device cannot decode Meshtastic packets, and vice versa. You need all devices on the same protocol to form a communicating network.

If you want to communicate with someone using a different protocol than you, one of you needs to switch. Check what protocol your local network and community uses before purchasing or flashing hardware.

What does this cost? Are there any ongoing fees?

Hardware cost: **\$20 - \$150** depending on the device. A basic Heltec V3 for MeshCore costs around \$22. A LILYGO T-Deck Plus with built-in keyboard and screen runs around \$100-115. (Prices are approximate and as of June 2026; hardware pricing changes often, so confirm against a current retailer listing.)

There are no ongoing fees, subscriptions, data plans, or service charges. Once you have the hardware, the network is free to use indefinitely. The firmware is open source and free. The apps are free. There are no cloud services you are paying for.

Optional costs: a better antenna (\$15 - \$30), a weatherproof enclosure (\$10 - \$20), and power hardware for a permanent outdoor node. None of these are required to get started.

Can I use this without a smartphone?

Most LoRa mesh devices require a smartphone running the companion app ([Meshtastic app](#) or MeshCore app) for messaging. The phone connects via Bluetooth and provides the user interface.

However, some devices have built-in keyboards and screens and can operate completely standalone (prices approximate and as of June 2026; verify against a current listing):

- **LILYGO T-Deck** (around \$55-75) - built-in QWERTY keyboard and color display
- **LILYGO T-Deck Plus** (around \$100-115) - T-Deck with larger battery and improved hardware

These standalone devices are popular for go-bag use, field operations, and anyone who prefers not to depend on a phone. They are fully functional mesh nodes without any external device required.

There is also a desktop application and web interface available for Meshtastic for use with a computer connected via USB or Bluetooth.

Is this secure? Can others read my messages?

Both platforms encrypt traffic, but they use different schemes, so it helps to keep them separate:

- **Meshtastic:** channel messages are encrypted with a per-channel pre-shared key (PSK) using AES-128 or AES-256 in CTR mode. Direct messages can additionally use public-key (PKI) encryption (see below).
- **MeshCore:** channel messages use AES (AES-128) with a shared channel key, while direct messages use Curve25519 ECDH public-key encryption between the two parties.

In both systems, the **default public channel** uses a well-known key that anyone can configure - it is essentially unencrypted in practice, similar to a public radio channel. For private communication, create a private channel with a custom key and share it only with intended recipients.

Direct messages (DMs) between specific nodes are encrypted with the recipient's public key in MeshCore. In Meshtastic (firmware v2.5 and later), DMs are also encrypted with the recipient's public key using public-key cryptography (X25519 key exchange + AES-CCM), the same general approach as MeshCore. Pre-2.5 Meshtastic firmware fell back to the shared channel key.

Hardware and Setup

Answers to common questions about choosing hardware, flashing firmware, and configuring your first LoRa mesh node.

What is the difference between MeshCore and Meshtastic?

Both are LoRa mesh networking protocols that run on similar hardware, but they differ in routing architecture and community:

- **Meshtastic** uses *flooding*: every relay node rebroadcasts every message. Simple, reliable in sparse networks, large global user base, extensive hardware support.
- **MeshCore** uses *path-discovery routing*: the network finds a route first, then only necessary nodes relay subsequent messages. More efficient in dense networks; less channel congestion at scale. Also includes Room Server (store-and-forward) functionality.

They are **not compatible with each other**. Pick the one your local community uses. See the full [MeshCore vs Meshtastic comparison page](#) for a detailed breakdown.

Which device should I buy as my first device?

The right choice depends on your protocol and use case:

- **MeshCore beginners:** The **Heltec V3** (~\$20-30 depending on source - lower on AliExpress, higher at US retailers; prices as of June 2026) is the most popular starting point. Compact, inexpensive, ESP32-based with a small display. Available on AliExpress or direct from Heltec.
- **Meshtastic beginners:** Check flasher.meshtastic.org for the current supported device list. The RAK WisBlock and LILYGO T-Beam are popular choices.

- **Standalone device (no phone needed): LILYGO T-Deck Plus** (~\$80). Built-in keyboard, color display, large battery. Fully self-contained.
 - **Best battery life in a pocket-carry device: LILYGO T-Echo** (~\$65). E-ink display, nRF52840-based; battery life ranges from several days to a couple of weeks depending on settings.
 - **Dedicated outdoor repeater:** Any ESP32 device in a weatherproof enclosure with a good antenna and solar power. The Heltec V3 or a WisBlock in a RAK enclosure are popular choices.
-

My device won't show up in the web flasher. What do I do?

This is the most common setup problem. Work through these steps in order:

1. **Try a different USB cable.** The single most common cause is a charge-only cable. Use a cable that you know works for data transfer (e.g., the cable that came with your phone and is used for file transfers).
 2. **Try a different USB port.** Some ports, particularly USB hubs or front-panel ports, are unreliable for serial devices. Use a direct port on the back of a desktop or side of a laptop.
 3. **Install drivers (Windows).** ESP32-based devices use a USB-to-serial chip, either CH340 or CP2102. If Windows doesn't automatically install the driver, download it manually:
 - CH340: search for "CH340 driver Windows"
 - CP2102: available from Silicon Labs website
 4. **Force bootloader mode manually:**
 - **ESP32 devices (Heltec V3, T-Beam, etc.):** Hold the BOOT button while plugging in the USB cable. Continue holding for 2 - 3 seconds after connecting. Release. The device should now appear in the flasher.
 - **nRF52 devices (T-Echo, RAK4631, T114):** Double-tap the reset button quickly. The device enters DFU mode and appears as a USB mass storage drive. Drag the .uf2 firmware file onto it.
 5. **Use Chrome or Edge.** The web flashers use the WebSerial API, which is only supported in Chrome and Chromium-based browsers (Edge, Brave, Opera). Firefox and Safari will not work.
-

I flashed the wrong firmware. Is my device bricked?

No. LoRa mesh devices are not easily bricked by flashing incorrect firmware. You can always re-flash with the correct firmware.

Procedure to recover:

- **ESP32 devices:** Force the device into bootloader mode by holding the BOOT button while connecting USB. Open the web flasher, select the correct firmware, and flash again.
- **nRF52 devices:** Double-tap reset to enter DFU mode, then drag the correct .uf2 firmware file onto the USB drive that appears.

In rare cases where the device seems completely unresponsive after a bad flash, fully erasing the flash memory and reflashing from scratch usually recovers it. The web flashers include an "Erase" option for this purpose.

How do I know which firmware variant to flash?

For MeshCore, there are several firmware variants with different roles:

- **Companion:** For personal handheld use. Connects to your phone via Bluetooth. This is what most users flash for their personal device.
- **Repeater:** For a dedicated infrastructure relay node. Optimized to relay messages; minimal user interface. Flash this for a hilltop or rooftop installation.
- **Room Server:** A store-and-forward bulletin board on the mesh. Retains messages and delivers them to connecting nodes. Flash this for a community message board node.

For Meshtastic, the firmware is the same regardless of role - the *role* is configured in the app settings after flashing (Client, Router, Repeater, etc.).

When in doubt: Flash Companion (MeshCore) or the standard firmware (Meshtastic) for a personal device. You can always re-flash with a different variant.

What antenna should I use?

The antenna that ships with most devices is a basic low-gain (roughly 1 - 3 dBi) stub antenna. It works but limits your range. Upgrading the antenna is the single highest-impact change you can make.

Recommendations by use case:

- **Handheld/portable:** A flexible rubber duck antenna (roughly 1 - 3 dBi) is fine for portability. The stock antenna is usually adequate.
- **Desktop or vehicle:** A 5 - 6 dBi fiberglass whip antenna with a magnetic base significantly improves range, typically in the ~\$15 - \$30 range (prices vary by supplier; as of June 2026).
- **Outdoor repeater:** A 5 - 8 dBi vertical omnidirectional antenna on a mast, with low-loss coax. Brands like Taoglas, Linx Technologies, and various suppliers on Amazon/AliExpress offer suitable options.

Before you install an outdoor/mast antenna - safety: Any outdoor antenna with coax running into a building needs **lightning/surge protection and grounding:** a coax surge arrestor (antenna discharge unit) on the lead-in, bonded to the building's grounding electrode system, plus a grounded mast per NEC Article 810. This is required, not optional - do not run coax from an outdoor antenna into your home without it. When erecting a mast, keep yourself, the mast, and the antenna at least **10 ft from any overhead power line** (if it could fall into a line, pick another spot), use a properly footed ladder, avoid wet/icy roofs, and have a second person present.

Verify the connector type on your device before purchasing. Most LoRa mesh devices (e.g., Heltec V3, T-Beam, T-Deck, T-Echo) use a standard **SMA** connector (female jack on the board, center pin in the female body). RP-SMA is mainly a Wi-Fi/Bluetooth convention and is *not* what these boards use. Standard SMA and RP-SMA are not interchangeable - they look similar but have the center pin gender reversed - so check your specific board before buying an antenna or pigtail.

How do I configure my region?

Region must be set before the device will transmit. In both MeshCore and Meshtastic apps, region is a required first-time setup step. Select **US** for the United States and Canada (915 MHz). The region setting is a regulatory control - it constrains the device to the frequencies and power limits legal in your country (902-928 MHz for US/Canada under FCC Part 15 / ISM RSS-247). Selecting the wrong region can make the device transmit on frequencies you are not authorized to use, which is unlawful, in addition to preventing communication with local nodes.

Battery and Power

Answers to common questions about battery life, power management, battery chemistry, solar sizing, and long-term deployments.

How long does the battery last?

Battery life varies significantly by device type, display type, messaging activity, and whether the device is acting as a relay. The figures below are approximate community estimates - actual runtime varies widely with screen use, GPS, and transmit duty cycle:

Device Type	Typical Battery Life	Notes
ESP32 client node (e.g., Heltec V3, T-Beam)	~1 - 3 days (approx.)	3000 mAh battery, moderate messaging. ESP32 nodes are power-hungry; T-Beam with GPS active is toward the shorter end.
nRF52 client node (e.g., T114, RAK4631)	~3 - 7 days (approx.)	Lower power than ESP32 (efficient MCU, no Wi-Fi radio)
E-ink display device (T-Echo, Wireless Paper)	~7 - 14 days (approx.)	E-ink uses power only when updating; excellent for always-on carry
Repeater node (always receiving)	Varies by MCU: ESP32 ~1 - 2 days on 3000 mAh; nRF52 several days	The always-on radio is the main draw. An nRF52 repeater (~10 - 15 mA) lasts far longer than an ESP32 one; for permanent sites plan for continuous/solar power rather than battery-only.

Power-saving tips: reduce TX power to the minimum needed for your use case; increase the sleep interval between beacon transmissions; disable GPS if not needed; use an e-ink device for always-on carry.

What battery chemistry should I use for outdoor deployments?

LiFePO4 (Lithium Iron Phosphate) is strongly preferred for any outdoor, unattended, or cold-weather deployment.

Why LiFePO4 over LiPo:

- **Temperature performance:** LiFePO4 operates reliably from about - 4°F (- 20°C) to 140°F (60°C), while LiPo degrades significantly below 32°F (0°C) and risks damage below - 4°F. Note: this is the *discharge* range - see the cold-weather note below for the separate, stricter charging limit.
- **Safety:** LiFePO4 is far more resistant to thermal runaway than LiPo, with a much higher onset temperature (~270°C vs ~150-210°C for LiPo/NMC), so it is much less likely to ignite under abuse. It is **not** completely immune, however - under severe abuse (puncture, dead short, gross overcharge, or charging a frozen cell) any lithium cell can vent flammable gas, overheat, and in extreme cases ignite. Still use a proper BMS and fusing, and avoid puncture or overcharge. LiPo combusts far more readily under these conditions.
- **Cycle life:** LiFePO4 typically lasts 2,000 - 5,000+ charge cycles. LiPo lasts roughly 300 - 1,000 cycles. For a permanently deployed solar-charged node, LiFePO4 can last a decade; LiPo may degrade in 1 - 3 years.
- **Voltage characteristics:** LiFePO4 has a flatter discharge curve (steady ~3.2V per cell vs. LiPo's declining curve), which means more consistent performance through the discharge cycle.

Critical cold-weather note: Do **not charge** any lithium chemistry, including LiFePO4, below 0°C (32°F) - sub-freezing charging causes lithium plating, which permanently damages the cell and can create internal shorts (a fire hazard on later cycles). For winter solar deployments, use a BMS or charge controller with low-temperature charge cutoff, or a self-heating/insulated battery. Separately, LiFePO4 also loses a large fraction of its usable capacity in extreme cold (roughly half by around - 40°F / - 40°C, as a rough estimate) - and note that many LiFePO4 cells are only rated for discharge down to about - 20°C, not - 40°C. If deploying in Minnesota, the Dakotas, Canada, or similar climates, check your cell's datasheet and size your battery bank for worst-case winter temperatures, not just rated capacity.

Are cheap 18650 batteries from Amazon OK?

Be very careful. First, for reference: a real 18650 cell holds roughly 2,500 - 3,500 mAh. Treat any listing above about 3,600 mAh as false advertising, regardless of price. The 18650 market on Amazon is saturated with counterfeit and overstated-capacity cells - a cell listed as "9800 mAh" for a few dollars is physically impossible.

Counterfeit cells often have:

- Actual capacity 20 - 50% of stated capacity
- Poor protection circuitry or none at all. (Note: many legitimate 18650 cells are intentionally sold "unprotected" - bare cells without a built-in protection PCB - which is normal and expected when they are used inside a pack that has its own BMS. The concern with counterfeits is the absence of *any* protection in a context that needs it, plus generally poor quality control.)
- Higher internal resistance = poor performance under load
- Increased fire/damage risk

Buy 18650 cells from reputable sources:

- **18650batterystore.com** - US-based, genuine cells, good selection
- **illumn.com** - US-based specialty battery retailer
- **Brand-name cells:** Samsung 30Q, Samsung 40T, Molicel P26A, Molicel P42A, Panasonic NCR18650B, LG MJ1

For outdoor deployments where capacity and reliability matter, buying genuine cells from a reputable source is worth the modest price premium over Amazon mystery cells.

How big a solar panel do I need for a repeater node?

A typical LoRa repeater node in the continental United States requires a surprisingly modest solar setup. Rules of thumb:

- **Panel:** 5 - 10 W is adequate for most locations during summer. A 10 W panel provides comfortable margin for cloudy days.
- **Battery:** Size for 3 - 5 days of runtime without any solar input. Worked example for a node drawing an assumed ~150 mA average (your node's actual average draw varies by role and TX duty - measure it if you can): $150 \text{ mA} \times 24 \text{ h} \times 3 \text{ days} = 10,800 \text{ mAh} = 10.8 \text{ Ah}$ minimum. A 20 Ah LiFePO4 battery provides good margin.

Safety: fuse the battery. Install an appropriately-rated fuse between the battery and the charge controller/load. An unfused lithium bank (especially a 20 - 40 Ah LiFePO4 pack) can deliver very high fault current into a wiring fault or a shorted controller and start a fire; a fuse is standard practice and is not optional.

Regional considerations:

- **Southern US (Texas, Arizona, California):** Ample sun year-round; 5 - 7 W panel is usually sufficient.
- **Northern US (Minnesota, North Dakota, Montana):** December peak sun hours typically drop to about **2.5 - 3.5 hours/day** (e.g. Minnesota winter is roughly 3 - 5 h/day) - significantly less than the 4 - 6 hours/day you get in summer. A 10 W panel and a larger battery bank (30 - 40 Ah) is recommended for year-round operation without manual intervention.
- **Pacific Northwest:** Low winter sun and frequent overcast push peak sun hours down toward ~2.5 h/day or less in deep winter; plan for 2 - 3 hours/day. Size accordingly or accept that the node may need occasional charging in deep winter.

Practical formula: Daily energy consumption (Wh) ÷ peak sun hours ÷ panel efficiency (typically 80% for a real system) = panel wattage needed. Always add 50 - 100% margin for real-world inefficiency, dirty panels, and suboptimal panel angle.

Can I charge LiFePO4 batteries with a standard LiPo charger?

No - use only a charger designed for LiFePO4. LiFePO4 cells have a different charge voltage profile than LiPo cells (3.65V/cell max for LiFePO4 vs. 4.2V/cell for LiPo). Charging LiFePO4 with a LiPo charger will overcharge the cells, reducing their life and potentially causing damage.

Purpose-built LiFePO4 solar charge controllers and battery management systems (BMS) are widely available and not expensive. Many solar charge controllers include a LiFePO4 mode.

Should I run my node from a USB power bank?

USB power banks work well for **portable and temporary deployments**. They are convenient, inexpensive, and widely available.

Limitations for permanent deployment:

- Most USB power banks shut off when they detect a low-current draw (like a standby LoRa node). This is called "low-current cutoff." The node will stop running after a short time even if the power bank is not depleted.

- Power banks are not designed for continuous solar charging - charging and discharging simultaneously (known as "pass-through") degrades many power banks quickly.

For permanent outdoor deployment, use a dedicated LiFePO4 battery with a proper solar charge controller rather than a consumer power bank.

My device gets warm during operation. Is this normal?

Mild warmth is normal, particularly during active transmission or when running at high TX power. Most ESP32 LoRa boards (e.g. Heltec V3, which use the SX1262) transmit at up to about 22 dBm conducted and run slightly warm at that level - this is not a concern at normal operating temperatures. Higher outputs (27 dBm and above, i.e. 0.5 W) are beyond the SX1262's native ~22 dBm capability and require an external power-amplifier module; a bare Heltec cannot reach them.

Concerns to watch for:

- Excessive heat from the battery area may indicate a failing or improperly charged lithium battery. If a cell is hot to the touch or swelling, **stop charging and disconnect it immediately**. Do not puncture, crush, or continue to use a swollen cell - it is a fire and venting hazard. Move it to a non-combustible surface away from flammable material, let it cool, and dispose of it at a battery-recycling or hazardous-waste facility - never in the household trash.
- Sustained high temperature inside a sealed enclosure can shorten component life. A weatherproof box is by definition sealed, so don't drill open holes (that defeats the weatherproofing) - instead use vapor-permeable vents (e.g. Gore vents) or a sun shield, and keep the enclosure out of direct sun where practical.

Common Questions

Setup and Configuration Questions

My device won't show up in the app. What do I check?

1. **Is Bluetooth enabled?** The app connects via BLE. Ensure Bluetooth is on in your phone settings and the app has Bluetooth permission.
2. **Is the device powered and running?** Check for activity LED or screen (if present). Some devices show no activity when running normally - this doesn't mean they're off.
3. **Driver installed?** For ESP32 boards such as the Heltec V3: the board may carry either a CH340 or a CP2102/CP210x USB-to-serial chip, so install whichever driver matches your specific unit (on Windows; some Macs may need manual installation). For nRF52840 devices (RAK4631, T-Echo): standard USB CDC driver, usually auto-installs.
4. **Reboot the device:** Hold the reset button or power-cycle. Some devices get stuck during initialization.
5. **Is the device in setup mode?** Some devices require a specific button sequence to enter BLE pairing mode. Check your device's documentation.

What preset should I use?

For MeshCore in the US and Canada: Use the USA/Canada preset. This is the community standard across all major North American MeshCore networks. In the MeshCore app: Settings → Radio → Choose Preset → USA/Canada (Recommended).

For Meshtastic: Check what your local community uses. Long Fast is the Meshtastic firmware default and is widely used in sparse networks. Medium Slow is increasingly common in dense urban networks. Nodes on different presets cannot hear each other even on the same channel name. Always confirm with your local community first.

My messages are sending but nobody replies. Is it working?

Possible explanations:

- Nobody else is on the network in your area right now. Mesh communities are most active in areas with established infrastructure.
- You're on a different preset from others in the area. Verify your preset matches the community standard.
- You're on a private or custom channel that others aren't monitoring. Switch to the default public channel.
- You're sending but no nodes are in range to receive. Try moving to higher elevation and try again.

To test whether your node is working at all: pair two devices you control, confirm they hear each other, then check signal quality (RSSI/SNR).

What role should I set my node to?

- **Client** (Meshtastic) / **Client** (MeshCore): Your default for a phone-paired portable node. Participates in the mesh, sends and receives messages. (On Meshtastic, the default CLIENT role stays awake and intelligently rebroadcasts to help the mesh; it does not sleep periodically by default - battery sleep is governed by separate power-config settings, not the role. MeshCore client behavior differs.)
- **Router** (Meshtastic): Like Client but always rebroadcasts and gains rebroadcast priority (it "cuts in line" ahead of other nodes). Meshtastic recommends ROUTER **only** for nodes in high, line-of-sight locations left running semi-permanently; misusing it on a poorly-placed node harms the mesh.
- **Repeater** (Meshtastic): Full-time infrastructure role that behaves like ROUTER but goes further - it rebroadcasts other nodes' packets while completely disabling its own broadcast traffic (telemetry/position). Use only for permanently deployed nodes in good locations.
- **Repeater** (MeshCore): A MeshCore repeater is **selective** - it does not blindly forward every packet the way a Meshtastic REPEATER rebroadcasts. It does not broadcast its own position by default. Use for permanently deployed infrastructure nodes.

When in doubt: use Client. Putting a poorly-placed node in Router or Repeater mode can actually degrade network performance by increasing traffic without improving coverage.

How do I know my repeater is actually working?

After deploying a new repeater:

1. Check whether it appears in the network map (meshmap.net for Meshtastic; your regional MeshCore map). Note: meshmap.net only shows nodes that report to the public Meshtastic MQTT server, so a working repeater with MQTT disabled or no internet uplink will *not* appear there - absence on the map does not mean the repeater is down. Verify locally instead.
2. Verify another node within range shows the repeater in its contact list (MeshCore) or node list (Meshtastic)
3. Send a message from 2+ miles away and confirm it routes through the repeater (check the hop count)
4. For MeshCore: enable flood advertisements so the repeater is visible across the network, then check that remote nodes can see it

How do I update firmware on my device?

The easiest method for most devices:

1. Connect device to computer via USB
2. For MeshCore: open flasher.meshcore.io in Chrome or Edge desktop (requires the Web Serial API)
3. For Meshtastic: open flasher.meshtastic.org in Chrome or Edge
4. Select your device type, select the latest firmware version, click Flash
5. Wait for completion - device will reboot automatically

After flashing, your radio settings are preserved but verify them before putting the node back in service. A firmware update can occasionally reset settings to defaults on some devices.

Glossary

Glossary of Mesh Networking Terms

A reference for terminology used throughout this wiki and in the mesh networking community.

A

Advertisement (advert)

A packet broadcast by a MeshCore node to announce its existence on the network. Advertisements contain the node's identity, position (if configured), and routing credentials. Other nodes use advertisements to discover repeaters and build routing tables.

APRS (Automatic Packet Reporting System)

An amateur radio protocol for broadcasting GPS positions and short messages. Some mesh gateways bridge position data to the APRS network, making mesh nodes visible on aprs.fi.

Autonomy period

How many days a battery-powered node can operate without solar input or recharging. This wiki uses 5 days of autonomy as a recommended design target for solar-powered repeaters; it is an editorial rule of thumb rather than a documented industry standard, so size your own deployment for your local worst-case weather.

B

Bandwidth (BW)

In LoRa, the width of the frequency channel used for transmission. Wider bandwidth = faster data rate but less range. Common values: 62.5, 125, 250, 500 kHz. The MeshCore USA/Canada preset uses 62.5 kHz BW.

BLE (Bluetooth Low Energy)

The wireless connection method used by most LoRa mesh apps to communicate with companion devices. BLE has a typical range of 30 - 100 feet and consumes very little power.

C

CAD (Channel Activity Detection)

A LoRa radio feature that listens for activity on the channel before transmitting, reducing collisions. Also called listen-before-talk (LBT).

Channel

A logical grouping of nodes that can communicate with each other. Nodes must be on the same channel to exchange messages. Channels can be public (no PSK) or private (encrypted with a shared PSK). Both MeshCore and Meshtastic support multiple channels.

Coding Rate (CR)

A LoRa parameter that controls forward error correction overhead. Higher CR (e.g., 4/8) adds more redundancy, allowing the receiver to reconstruct damaged packets. Lower CR (4/5) has less overhead but requires cleaner signals. Affects airtime and link budget.

CascadiaMesh

A MeshCore community mesh network covering the Pacific Northwest (Oregon, Washington, and into British Columbia). Operates a backbone infrastructure along the I-5 corridor. See cascadiamesh.org.

D

dBi (decibels relative to isotropic)

A measure of antenna gain relative to a theoretical perfect omnidirectional antenna. A 6 dBi antenna radiates $\sim 4\times$ more power in its main direction compared to an isotropic antenna ($6\text{ dB} = \sim 4\times$ power), which corresponds to roughly $2\times$ the range in free space (range scales with the square root of power). Higher dBi = more directional (for verticals: more concentrated horizontally).

dBm (decibels relative to milliwatt)

A measure of power level. $0\text{ dBm} = 1\text{ mW}$. $27\text{ dBm} \approx 500\text{ mW}$. $30\text{ dBm} = 1000\text{ mW} = 1\text{ W}$. Used to express transmit power and signal strength (RSSI).

DFU (Device Firmware Update)

A firmware update mode on nRF52840-based devices (T-Echo, RAK4631, etc.). Entered by double-tapping the reset button. The device appears as a USB drive, and firmware is updated by copying a .uf2 file to it.

E

EIRP (Effective Isotropic Radiated Power)

The total RF power broadcast in the main direction of the antenna, accounting for both transmit power and antenna gain. Under FCC Part 15.247(b)(3)-(b)(4) for the 902-928 MHz band, conducted power is capped at 1 W (30 dBm) and full power is allowed with antenna gain up to 6 dBi; above 6 dBi the conducted power must be reduced dB-for-dB, so EIRP is effectively held near 36 dBm rather than there being a standalone "4 W EIRP" allowance.

With a 6 dBi antenna at 30 dBm TX, EIRP = 36 dBm - the boundary of the gain allowance.

EasySkyMesh

A community fork of MeshCore firmware optimized for ultra-low power consumption, targeting a Heltec board (commonly referred to as the Heltec V3/V4 family; the exact revision should be confirmed against the project's own documentation). Reported to achieve roughly 5.5 mA average idle current. Not the official MeshCore firmware.

F

Flooding

A routing approach in which every received packet is re-broadcast by every router node (up to the hop limit). Simple and robust but creates significant traffic in dense networks.

Meshtastic uses flooding as its primary routing method. MeshCore relies mainly on path-discovery routing, but it still floods traffic that has no specific destination - notably adverts and channel (group) messages - so it is not accurate to say MeshCore never floods.

Firmware

The software that runs on a LoRa device. Determines which protocol (MeshCore or Meshtastic), which features are active, and how the radio is configured. Flashed to the device via USB using a web flasher or PlatformIO.

G - H

Gateway

A node that bridges the mesh network to the internet. In Meshtastic, gateways typically use MQTT. In MeshCore, gateways are usually implemented via room servers with internet uplinks.

HAL (Hardware Abstraction Layer)

A software layer in MeshCore firmware that separates platform-specific hardware code (ESP32, nRF52840) from the common protocol logic. Allows the same firmware core to run on different MCU families.

Hop

One relay of a message from one node to the next. A message with 3 hops has been relayed through 3 intermediate nodes between sender and receiver. Each hop adds latency and reduces reliability. Most deployments target 3 - 5 hop maximum.

Hop limit

The maximum number of times a packet is relayed before being discarded. Prevents packets from circulating indefinitely in the mesh.

I - L

ISM band

Industrial, Scientific, and Medical frequency bands designated for unlicensed use. In North America, the 902 - 928 MHz band (commonly called "915 MHz") is used for LoRa mesh. In Europe, 863 - 870 MHz ("868 MHz").

LiFePO4 (Lithium Iron Phosphate)

A rechargeable battery chemistry preferred for outdoor mesh deployments. Advantages over LiPo: wider temperature range, longer cycle life (2000+ vs 300 - 500 cycles), much safer - highly resistant to thermal runaway and far less prone to fire than LiPo, though not categorically fireproof under severe abuse, and it still requires proper charging and protection. Lower energy density. Nominal voltage: 3.2V per cell.

Link budget

The total signal loss a radio link can absorb while still achieving reliable communication. Calculated as TX power + antenna gains – required receive sensitivity. Higher link budget = more range or better penetration through obstacles. A larger spreading factor increases the link budget (but reduces data rate).

LoRa (Long Range)

A proprietary chirp spread-spectrum radio modulation developed by Semtech. LoRa achieves exceptional range at low power and low data rates. The physical layer technology underlying both MeshCore and Meshtastic.

LoRaWAN

A network protocol that uses LoRa radio. Different from LoRa mesh - LoRaWAN requires centralized gateways and a network server. MeshCore and Meshtastic are not LoRaWAN.

M - N

MCU (Microcontroller Unit)

The processor chip in a LoRa node. Common MCUs: ESP32-S3 (higher power, has Wi-Fi), nRF52840 (ultra-low power, BLE only). Choice of MCU significantly affects power consumption and available interfaces.

MeshCore

A free, open-source LoRa mesh networking protocol and firmware. Uses path-discovery routing. Per community technical write-ups, its channel encryption is an Encrypt-then-MAC construction using AES-128 in ECB mode with an HMAC-SHA256 MAC; note that ECB is a known cryptographic weakness. Refer to the MeshCore repository/official docs for the authoritative cipher details. Strong North American community with several regional networks. Not to be confused with Meshtastic.

Meshtastic

A free, open-source LoRa mesh networking protocol and firmware. Uses flooding-based routing. Larger global community. Not interoperable with MeshCore.

MPPT (Maximum Power Point Tracking)

A solar charge controller algorithm that continuously adjusts the operating point to extract maximum power from a solar panel. More efficient than PWM controllers, especially on larger systems and in cold climates; on very small, low-current LoRa nodes the advantage is

marginal and an inexpensive PWM controller is often adequate and more cost-effective.

MQTT (Message Queuing Telemetry Transport)

A lightweight publish/subscribe messaging protocol. Used by Meshtastic gateways to bridge mesh traffic to the internet. Some MeshCore room servers also support MQTT bridging for monitoring and integration.

NoDakMesh

A MeshCore community mesh network covering North Dakota and the Northern Plains. One of the regional networks using the standard USA/Canada MeshCore preset. See nodakmesh.org.

P - R

Path-discovery routing

The routing approach used by MeshCore. When a node wants to reach a destination, it broadcasts a route request (path discovery packet). Repeaters append themselves to build a route record. The destination responds with a route reply (path acknowledgment). Subsequent traffic follows the discovered path rather than flooding. Note that MeshCore still floods channel (group) messages and advertises that have no specific destination. More efficient than flooding for directed traffic in large or busy networks.

Preset

A named set of radio parameters (spreading factor, bandwidth, coding rate) that determines the tradeoff between range and data rate. Meshtastic ships roughly 9-10 modem presets (Short Turbo through Very Long Slow); the exact count changes between firmware versions, so check meshtastic.org/docs/configuration/radio/loras for the current list (as of 2026-06-08). MeshCore uses regional presets (USA/Canada, Europe, etc.). Nodes must use the same preset to communicate.

PSK (Pre-Shared Key)

A shared secret key used to encrypt messages on a private channel. All nodes on the channel must have the same PSK configured. Meshtastic encrypts each channel with AES-CTR (128- or 256-bit) keyed by the channel PSK; MeshCore uses per-contact / Diffie-Hellman-derived keys with AES-128 and an HMAC-SHA256 MAC.

RegionMesh

A United States community mesh network built on MeshCore; community-run and open-source with no usage fees. See regionmesh.com for current coverage.

Room server

A MeshCore infrastructure component that runs on a server (Raspberry Pi, VPS, etc.) and provides message persistence, node discovery, and optionally internet bridging. The MeshCore equivalent of a Meshtastic MQTT gateway, but with more features.

RSSI (Received Signal Strength Indicator)

The power level of a received radio signal, measured in dBm. More negative = weaker. As a rule of thumb, usable LoRa RSSI ranges from roughly -40 to -60 dBm (very strong) down to about -120 to -137 dBm at high spreading factors (the Semtech SX1262 datasheet quotes sensitivity near -137 dBm at SF12). SNR is more informative than RSSI alone for assessing

link quality.

S - Z

SNR (Signal-to-Noise Ratio)

The ratio of signal strength to background noise, measured in dB. Unlike RSSI, SNR indicates how far above the noise floor the signal is. LoRa can decode signals below the noise floor - roughly -7.5 dB at SF7 down to about -20 dB at SF12 under ideal conditions (per the Semtech SX1262 datasheet). Use conservative margins for emergency link planning in noisy environments. SNR is a better indicator of link quality than RSSI for LoRa links.

Spreading Factor (SF)

A LoRa parameter (SF7 to SF12) that controls how long each symbol is transmitted. Higher SF = longer range, longer airtime, lower data rate. SF12 roughly quadruples range over SF7 in open terrain but is on the order of $32\times$ slower (each SF step doubles airtime), drastically reducing data rate.

SWR (Standing Wave Ratio)

A measure of impedance mismatch between a transmitter and antenna. Perfect match = 1:1 SWR. $<1.5:1$ is excellent for LoRa; $>3:1$ indicates a significant problem (damaged connector, wrong frequency antenna, etc.). Importantly, a very high SWR - such as transmitting with the antenna missing or disconnected - reflects power back into the power amplifier and can damage the transmitter, which is why a node should never transmit without an antenna attached.

WCMesh (West Coast Mesh)

A MeshCore community mesh network covering the US West Coast, with a strong presence in Oregon, Washington, and northern California. Uses the USA/Canada preset (with some regional frequency variations). See wcmesh.com.

MeshCore Official FAQ

FAQ: 1. Introduction

1.1. Q: What is MeshCore?

A: MeshCore is a multi platform system for enabling secure text based communications utilising LoRa radio hardware. The project lists its intended use cases as Off-Grid Communication, Emergency Response & Disaster Recovery, Outdoor Activities, Tactical Security including law enforcement and private security, and IoT sensor networks. ([source](#)) *Editorial note: these are vendor-listed use cases. For emergency, disaster, or tactical use, treat MeshCore as a supplemental best-effort text channel only - it offers no guaranteed delivery or quality-of-service and its channel encryption has known limitations, so it should not be relied on as a primary channel for life-safety traffic.*

MeshCore is free and open source:

- MeshCore is the routing and firmware etc, available on GitHub under MIT license
- There are clients made by the community, such as the web clients, these are free to use, and some are open source too
- The cross platform mobile app developed by [Liam Cottle](#) for Android/iOS/PC etc is free to download and use
- The T-Deck firmware is developed by Scott at Ripple Radios, the creator of MeshCore, is also free to flash on your devices and use

Some more advanced, but optional features are available on T-Deck if you register your device for a key to unlock. On the MeshCore smartphone clients for Android and iOS/iPadOS, you can unlock the wait timer for repeater and room server remote management over RF feature.

These features are completely optional and aren't needed for the core messaging experience. They're like super bonus features and to help the developers continue to work on these amazing features, they may charge a small fee for an unlock code to utilise the advanced features.

Anyone is able to build anything they like on top of MeshCore without paying anything.

1.2. Q: What do you need to start using MeshCore?

A: Everything you need for MeshCore is available at:

- Main Website: <https://meshcore.io>
- Firmware Flasher: <https://flasher.meshcore.io>

- MeshCore Firmware on GitHub: <https://github.com/meshcore-dev/MeshCore>
- MeshCore Companion Web App: <https://app.meshcore.nz>
- MeshCore Map: <https://map.meshcore.io>
- Liam Cottle's [MeshCore Technical Presentation](#)

You need LoRa hardware devices to run MeshCore firmware as clients or server (repeater and room server).

1.2.1. Hardware

MeshCore is available on a variety of 433MHz, 868MHz and 915MHz LoRa devices. For example, Lilygo T-Deck, T-Pager, RAK Wireless WisBlock RAK4631 devices (e.g. 19003, 19007, 19026), Heltec V3, Xiao S3 WIO, Xiao C3, Heltec T114, Station G2, Nano G2 Ultra, Seeed Studio T1000-E. More devices are being added regularly.

For an up-to-date list of supported devices, please go to <https://flasher.meshcore.io>

To use MeshCore without using a phone as the client interface, you can run MeshCore on a Lilygo's T-Deck, T-Deck Plus, T-Pager, T-Watch, or T-Display Pro. MeshCore Ultra firmware running on these devices are a complete off-grid secure communication solution.

1.2.2. Firmware

MeshCore offers four firmware roles: BLE Companion, USB Serial Companion, Repeater, and Room Server. Each is described below.

1.2.3. Companion Radio Firmware

Companion radios are for connecting to the Android app or web app as a messenger client. There are two different companion radio firmware versions:

1. **BLE Companion**

BLE Companion firmware runs on a supported LoRa device and connects to a smart device running the Android or iOS MeshCore client over BLE

1. **USB Serial Companion**

USB Serial Companion firmware runs on a supported LoRa device and connects to a smart device or a computer over USB Serial running the MeshCore web client

1.2.4. Repeater

Repeaters are used to extend the range of a MeshCore network. Repeater firmware runs on the same devices that run client firmware. A repeater's job is to forward MeshCore packets toward their destination. Unlike a simple flood-everything mesh, MeshCore repeaters follow embedded paths for direct messages once a path is known - but they do still forward flood-routed traffic such as adverts and channel (group) messages that have no specific destination.

A repeater can be remotely administered using a T-Deck running the MeshCore firmware with remote administration features unlocked, or from a BLE Companion client connected to a smartphone running the MeshCore app.

1.2.5. Room Server

A room server is a simple BBS server for sharing posts. T-Deck devices running MeshCore firmware or a BLE Companion client connected to a smartphone running the MeshCore app can connect to a room server.

Room servers store message history on them and push the stored messages to users. Room servers allow roaming users to come back later and retrieve message history. With channels, messages are either received when it's sent, or not received and missed if the channel user is out of range. Room servers are different and more like email servers where you can come back later and get your emails from your mail server.

A room server can be remotely administered using a T-Deck running the MeshCore firmware with remote administration features unlocked, or from a BLE Companion client connected to a smartphone running the MeshCore app.

When a client logs into a room server, the server pushes up to the 32 most recent messages the client has not yet seen.

Although room server can also repeat with the command line command `set repeat on`, it is not recommended nor encouraged. A room server with repeat set to `on` lacks the full set of repeater and remote administration features that are only available in the repeater firmware.

The recommendation is to run repeater and room server on separate devices for the best experience.

FAQ: 2. Initial Setup

2.1. Q: How many devices do I need to start using MeshCore?

A: If you have one supported device, flash the BLE Companion firmware and use your device as a client. You can connect to the device using the Android or iOS client via Bluetooth. You can start communicating with other MeshCore users near you.

If you have two supported devices, and there are not many MeshCore users near you, flash both to BLE Companion firmware so you can use your devices to communicate with your near-by friends and family.

If you have two supported devices, and there are other MeshCore users nearby, you can flash one of your devices with BLE Companion firmware and flash another supported device to repeater firmware. Place the repeater high above ground to extend your MeshCore network's reach.

After you flashed the latest firmware onto your repeater device, keep the device connected to your computer via USB serial, use the console feature on the web flasher and set the frequency for your region or country, so your client can remote administer the repeater or room server over RF:

```
set freq {frequency}
```

The repeater and room server CLI reference is here: <https://github.com/meshcore-dev/MeshCore/wiki/Repeater-&-Room-Server-CLI-Reference>

If you have more supported devices, you can use your additional devices with the room server firmware.

2.2. Q: Does MeshCore cost any money?

A: All radio firmware versions (e.g. for Heltec V3, RAK, T-1000E, etc) are free and open source developed by Scott at Ripple Radios.

The native Android and iOS client uses the freemium model and is developed by Liam Cottle, developer of meshtastic map at meshtastic.liamcottle.net on [GitHub](https://github.com) and [reticulum-meshchat on github](https://reticulum-meshchat.ongithub.com).

The T-Deck firmware is free to download and most features are available without cost. To support the firmware developer, you can pay for a registration key to unlock your T-Deck for deeper map

zoom and remote server administration over RF using the T-Deck. You do not need to pay for the registration to use your T-Deck for direct messaging and connecting to repeaters and room servers.

2.3. Q: What frequencies are supported by MeshCore?

A: It supports the 868MHz range in the UK/EU and the 915MHz range in New Zealand, Australia, and the USA. Countries and regions in these two frequency ranges are also supported. Exact legal frequencies, channel plans, and duty-cycle limits vary by country (e.g., EU 863-870 MHz has duty-cycle restrictions; US uses 902-928 MHz). Always select the preset for your specific country and confirm it against your national regulator before transmitting.

Use the smartphone client, or the repeater setup feature on the web flasher, to set your radio's RF settings by choosing the preset for your region.

Recently, as of October 2025, many regions have moved to the "narrow" setting, aka using BW62.5 and a lower SF number (instead of the original SF11). For example, USA/Canada (Recommended) preset is 910.525MHz, SF7, BW62.5, CR5. These exact values change over time - confirm the current recommended preset with your regional community or the live MeshCore preset list before relying on a specific frequency.

After extensive testing, many regions have switched or about to switch over to BW62.5 and SF7, 8, or 9. Narrower bandwidth setting and lower SF setting allow MeshCore's radio signals to fit between interference in the ISM band, provide for a lower noise floor, better SNR, and faster transmissions.

If you have consensus from your community in your region to update your region's preset recommendation, please post your update request on the [#meshcore-app](#) channel on the [MeshCore Discord server](#) to let Liam Cottle know.

2.4. Q: What is an "advert" in MeshCore?

A:

Advert means to advertise yourself on the network. In Reticulum terms it would be to announce. In Meshtastic terms it would be the node sending its node info.

MeshCore allows you to manually broadcast your name, position and public encryption key, which is also signed to prevent spoofing. When you click the advert button, it broadcasts that data over LoRa. MeshCore calls that an Advert. There's two ways to advert, "zero hop" and "flood".

- Zero hop means your advert is broadcasted out to anyone that can hear it, and that's it.
- Flooded means it's broadcasted out and then repeated by all the repeaters that hear it.

MeshCore clients only advertise themselves when the user initiates it. A repeater sends a flood advert once every 12 hours by default. This interval can be configured using the following command:

```
set flood.advert.interval {hours}
```

The separate `set advert.interval {minutes}` command controls the local zero-hop advert timer.

2.5. Q: Is there a hop limit?

A: Internally the firmware has maximum limit of 64 hops. In real world settings it will be difficult to get close to the limit due to the environments and timing as packets travel further and further. Practical reliable delivery is a handful of hops; latency and packet loss accumulate quickly, so do not design emergency relay chains around large hop counts. We want to hear how far your MeshCore conversations go.

FAQ: 3. Server Administration

3.1. Q: How do you configure a repeater or a room server?

A: - When MeshCore is flashed onto a LoRa device for the first time, it is necessary to set the server device's frequency to make it utilize the frequency that is legal in your country or region.

Repeater or room server can be administered with one of the options below:

- After a repeater or room server firmware is flashed on to a LoRa device, go to and use the web user interface to connect to the LoRa device via USB serial. From there you can set the name of the server, its frequency and other related settings, location, passwords etc.

[!image](#)

- Connect the server device using a USB cable to a computer running Chrome on the MeshCore web tool (as of 2026, configuration is at <https://config.meshcore.io> and flashing at <https://flasher.meshcore.io>; verify which host currently provides the serial console), then use the `console` feature to connect to the device
- Use a MeshCore smartphone clients to remotely administer servers via LoRa.
- A T-Deck running unlocked/registered MeshCore firmware. Remote server administration is enabled through registering your T-Deck with Ripple Radios. It is one of the ways to support MeshCore development. You can register your T-Deck through the Ripple Radios registration page (check the current MeshCore / Ripple Radios site for the active registration link).

3.2. Q: Do I need to set the location for a repeater?

A: While not required, with location set for a repeater it will show up on the MeshCore map in the future. Set location with the following command:

```
set lat
```

```
set lon
```

You can get the latitude and longitude from Google Maps by right-clicking the location you are at on the map.

3.3. Q: What is the password to administer a repeater or a room server?

A: The default admin password to a repeater and room server is `password`. Use the following command to change the admin password:

```
password {new-password}
```

3.4. Q: What is the password to join a room server?

A: The default guest password to a room server is `hello`. Use the following command to change the guest password:

```
set guest.password {guest-password}
```

⚠ **Security warning — change both defaults before deploying.** The default admin password (`password`) and guest password (`hello`) are publicly known. A repeater or room server left on these defaults can be remotely administered, reconfigured, or hijacked by **anyone within radio range** — a serious risk for community or emergency infrastructure. Immediately set a strong, unique admin password (and guest password) on every deployed node, before it goes on the air, using the commands above.

3.5. Q: Can I retrieve a repeater's private key or set a repeater's private key?

A: You can issue these commands to get or set a repeater's private key using a USB serial connection.

```
get prv.key
```

 to print a repeater's private key on the serial console

```
set prv.key
```

 to set a repeater's private key on the serial console

Reboot the repeater after `set prv.key` command for the new private key to take effect.

3.6. Q: The first byte of my repeater's public key collides with an existing repeater on the mesh. How do I get a new private key with a matching public key that has its first byte of my choosing?

A: You can generate a new private key and specific the first byte of its public key here:
<https://gessaman.com/mc-keygen/>

Having multiple repeaters with the same first byte ID does not negatively affect the mesh or its functionality. Flood and pathed packets will still reach their destinations. First byte ID collision makes traceroute and path analysis harder because these tools don't know exactly which of the two (or more) colliding repeaters is the one in the path.

Best practice is when you set up a new repeater, choose a public key that is not in use. If it is not possible to find a unique first byte for your repeater's public key, choose one that is unique within about 10 miles (16 km) to minimize collision with nearby repeaters.

3.7. Q: My repeater maybe suffering from deafness due to high power interference near my mesh's frequency, it is not hearing other in-range MeshCore radios. What can I do?

A: This appears to be related to the SX1262 radio's automatic gain control behavior (see the Semtech SX1262 datasheet for how AGC operates). As an empirical mitigation, you can periodically reset its AGC with the command below.

```
set agc.reset.interval {seconds}
```

The interval is specified in seconds and must be a multiple of 4. Example: `set agc.reset.interval 4` resets AGC every 4 seconds, which works well to cure deafness.

This is a very low cost operation. In the firmware, the AGC reset is implemented by setting `state = STATE_IDLE;` in the function `RadioLibWrapper::resetAGC()` in `RadioLibWrappers.cpp`.

3.8. Q: How do I make my repeater an observer on the mesh?

A: The observer instruction is available here: <https://analyzer.letsmesh.net/observer/onboard>

3.9. Q: What is multi-byte support? What do 1-byte, 2-byte, 3-byte adverts and messages mean?

A:

The original MeshCore protocol design uses the first byte of a repeater's public key to denote the repeater in a path. And with 1 byte for each repeater in the path, MeshCore packets can travel as many as 64 hops.

However, with 1 byte, there are only 254 unique IDs (exclude 00 and FF which are reserved). Many meshes group have multiple repeaters with the same first byte in their public keys. Packets

continue to pass through repeaters and the mesh is not harmed in anyway. It does make it harder for tools to analyze paths with duplicated repeater IDs.

Firmware version 1.14 and newer introduces the ability for repeaters to advert with 1-, 2-, or 3-byte adverts. Companions can also send out channel and direct messages with 1-, 2-, or 3-byte path. Adverts and messages sent in 1-byte path is compatible with repeater firmware older or newer than 1.14. Because each extra path-hash byte per hop consumes packet space, 1-byte paths reach up to 64 hops, 2-byte adverts and messages will travel up to 32 hops, and 3-byte adverts and messages will travel up to 21 hops.

3.9.1. Q: **What path hash sizes will my repeater forward?**

Repeaters running firmware 1.14+ repeat packets sent with 1-, 2-, or 3-byte path hash. Repeaters on firmware older than 1.14 only repeat 1-byte path hash packets and silently drop 2- and 3-byte packets.

3.9.2. Q: **What determines a packet's path hash size?**

The original packet sender determines the path hash size. The most common original sender is a companion app. The other common original sender is a repeater, when it broadcasts its advert.

3.9.3. Q: **How do I change my companion's path hash size?**

As of firmware version 1.14 and MeshCore app version 1.41.0, in the MeshCore app, you can set your companion's message path hash size in `Settings (gear icon)`, `Experimental Settings`.

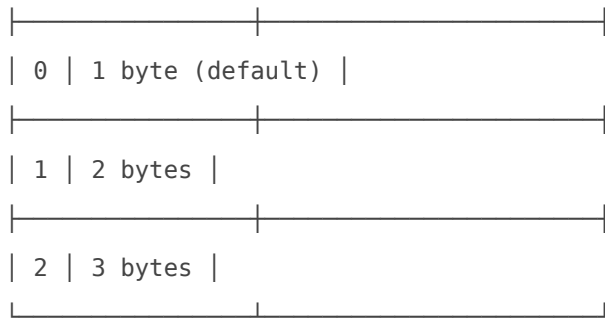
Until your regional mesh has the vast majority of the repeaters updated to 1.14+ firmware, it is recommended to keep your companion at the default 1-byte because pre-1.14 repeaters will silently drop messages with larger path hashes.

3.9.4. Q: **What does the CLI command `path.hash.mode` do on a repeater?**

This CLI command `path.hash.mode` *only* controls the path hash size used in a repeater's own advert broadcasts. It does **NOT** affect which packets the repeater forwards. A repeater with firmware 1.14+ always forward 1-, 2-, and 3-byte packets regardless of this setting.

Usage: `set path.hash.mode {0|1|2}`:

<code>path.hash.mode</code>	Advert path hash size
-----------------------------	-----------------------



It is safe to set your 1.14+ repeaters to mode 1 or 2.

3.9.5. Q: **Why use 2- or 3-byte path hash for adverts?**

A longer path hash helps tools like the LetsMesh.net Analyzer and MeshMapper disambiguate repeaters more reliably. With only 1 byte, the chance of different repeaters having the same first byte in their public key is high, making it harder to tell them apart in mesh network analysis. Since this only affects adverts, there's no downside. 2- and 3-byte adverts don't travel as far as 1-byte adverts, but it is not important for MeshCore nodes to hear a repeater's advert that are 21 or 32 hops away.

3.9.6. Q: **When can we move away from 1-byte path hash for channel and direct messages?**

You should move to send 2-byte or 3-byte channel and direct messages when the vast majority of the repeaters in your regional mesh are updated to firmware version 1.14 or newer. Setting your repeater's `path.hash.mode` to 1 (for 2-byte path hash) or 2 (for 3-byte path hash) now helps the community gauge to how many repeaters have updated to 1.14+. Please work with your MeshCore community together to decide when to switch to 2-byte path or 3-byte path for channel and direct messages.

FAQ: 4. T-Deck Related

4.1. Q: Is there a user guide for T-Deck, T-Pager, T-Watch, or T-Display Pro?

A: Yes, it is available on <https://buymeacoffee.com/ripplebiz/ultra-v7-7-guide-meshcore-users>

4.2. Q: What are the steps to get a T-Deck into DFU (Device Firmware Update) mode?

A:

1. Device off
2. Connect USB cable to device
3. Hold down trackball (keep holding)
4. Turn on device
5. Hear USB connection sound
6. Release trackball
7. T-Deck in DFU mode now
8. At this point you can begin flashing using the MeshCore web flasher at flasher.meshcore.io (use the Console / flash option).

4.3. Q: Why is my T-Deck Plus not getting any satellite lock?

A: For T-Deck Plus, the GPS baud rate should be set to **38400**. Also, some T-Deck Plus devices were found to have the GPS module installed upside down, with the GPS antenna facing down instead of up. If your T-Deck Plus still doesn't get any satellite lock after setting the baud rate to 38400, you might need to open the device to check the GPS orientation.

GPS on T-Deck is always enabled. You can skip the "GPS clock sync" and the T-Deck will continue to try to get a GPS lock. You can go to the `GPS Info` screen; you should see the `Sentences:` counter increasing if the baud rate is correct.

[Source](#)

4.4. Q: Why is my OG (non-Plus) T-Deck not getting any satellite lock?

A: The OG (non-Plus) T-Deck doesn't come with a GPS. If you added a GPS to your OG T-Deck, please refer to the manual of your GPS to see what baud rate it requires. Alternatively, you can try to set the baud rate from 9600, 19200, etc., and up to 115200 to see which one works.

4.5. Q: What size of SD card does the T-Deck support?

A: Users have had no issues using 16GB or 32GB SD cards. Format the SD card to **FAT32**.

4.6. Q: what is the public key for the default public channel?

A:

T-Deck uses the same key the smartphone apps use but in base64

```
iz0H6cXN6mrJ5e26oRXNcg==
```

Note: This is the shared key for the public default channel and is known to everyone - it provides **NO privacy**. Anyone can read default-channel traffic. Never send sensitive information on the default channel; create a private channel with your own key for any confidential traffic.

There is no `=` key on the T-Deck's hardware keyboard. You can use the on-screen software keyboard to enter `=`. Tap the text box to enable the on-screen software keyboard.

In `iz0H6cXN6mrJ5e26oRXNcg==`, the 3rd character is a capital letter `0` (as in Oscar), not the digit zero `0`.

The smartphone app key is in hex:

```
8b3387e9c5cdea6ac9e5edbaa115cd72
```

[Source](#)

4.7. Q: How do I get maps on T-Deck?

A: You need map tiles. Pre-downloaded map tile bundles (for Europe and the US) are sometimes shared as a way to support development. *(Note: the specific download links are not currently available here - check the MeshCore Discord or the official T-Deck guide for the current tile-bundle links.)*

Another way to download map tiles is to use a Python tile-download script to get the tiles for the areas you want. *(The script link is not currently provided here; the MeshCore community shares*

both a base script and a modified version with extra error handling and parallel downloads - ask in the MeshCore Discord for the current links.)

4.8. Q: Where do the map tiles go?

Once you have the tiles downloaded, copy the `\tiles` folder to the root of your T-Deck's SD card.

4.9. Q: How to unlock deeper map zoom and server management features on T-Deck?

A: You can download, install, and use the T-Deck firmware for free, but it has some features (map zoom, server administration) that are enabled if you purchase an unlock code for \$10 per T-Deck device.

Unlock page: *(link not currently provided here - the unlock code is sold via the developer's store; see the official T-Deck guide linked in 4.1 for the current purchase page.)*

4.10. Q: How to decipher the diagnostics screen on T-Deck?

A: Space is tight on T-Deck's screen, so the information is a bit cryptic. The format is :

```
{hops} l:{packet-length}({payload-len}) t:{packet-type} snr:{n} rssi:{n}
```

See here for packet-type:

<https://github.com/meshcore-dev/MeshCore/blob/main/src/Packet.h#L19>

```
#define PAYLOAD_TYPE_REQ 0x00 // request (prefixed with dest/src hashes, MAC) (enc data: timestamp, blob)
```

```
#define PAYLOAD_TYPE_RESPONSE 0x01 // response to REQ or ANON_REQ (prefixed with dest/src hashes, MAC) (enc data: timestamp, blob)
```

```
#define PAYLOAD_TYPE_TXT_MSG 0x02 // a plain text message (prefixed with dest/src hashes, MAC) (enc data: timestamp, text)
```

```
#define PAYLOAD_TYPE_ACK 0x03 // a simple ack #define PAYLOAD_TYPE_ADVERT 0x04 // a node advertising its Identity
```

```
#define PAYLOAD_TYPE_GRP_TXT 0x05 // an (unverified) group text message (prefixed with channel hash, MAC) (enc data: timestamp, "name: msg")
```

```
#define PAYLOAD_TYPE_GRP_DATA 0x06 // an (unverified) group datagram (prefixed with channel hash, MAC) (enc data: data_type, data_len, blob)
```

```
#define PAYLOAD_TYPE_ANON_REQ 0x07 // generic request (prefixed with dest_hash, ephemeral  
pub_key, MAC) (enc data: ...)
```

```
#define PAYLOAD_TYPE_PATH 0x08 // returned path (prefixed with dest/src hashes, MAC) (enc  
data: path, extra)
```

[Source](#)

4.11. Q: The T-Deck sound is too loud, or can you customize the sound?

A: You can customise (and effectively quiet down) the sounds on the T-Deck by placing your own `.mp3` files - including silent or quieter clips - onto the `root` dir of the SD card. Replacing a sound file with a silent or low-volume clip is the way to reduce or disable that alert. The files are:

- `startup.mp3`
- `error.mp3`
- `alert.mp3`
- `new-advert.mp3`
- `existing-advert.mp3`

4.13. Q: What is the 'Import from Clipboard' feature on the t-deck and is there a way to manually add nodes without having to receive adverts?

A: 'Import from Clipboard' is for importing a contact via a file named 'clipboard.txt' on the SD card. The opposite, is in the Identity screen, the 'Card to Clipboard' menu, which writes to 'clipboard.txt' so you can share yourself (call these 'biz cards', that start with "meshcore://...")

4.14. Q: How to capture a screenshot on T-Deck?

A: To capture a screenshot on a T-Deck, long press the top-left corner of the screen. The screenshot is saved to the microSD card, if one is inserted into the device.

FAQ: 5. General

5.1. Q: What are BW, SF, and CR?

A:

BW is bandwidth - width of frequency spectrum that is used for transmission

SF is spreading factor - sets how many chips encode each symbol (2^{SF}); higher SF means longer airtime, greater range and sensitivity, but lower data rate.

CR is coding rate - from: <https://www.thethingsnetwork.org/docs/lorawan/fec-and-code-rate/>

TL;DR: default CR to 5 for good stable links. If it is not a solid link and is intermittent, change to CR to 7 or 8.

Forward Error Correction is a process of adding redundant bits to the data to be transmitted. During the transmission, data may get corrupted by interference (changes from 0 to 1 / 1 to 0). These error correction bits are used at the receivers for restoring corrupted bits.

The Code Rate of a forward error correction expresses the proportion of bits in a data stream that actually carry useful information.

There are 4 code rates used in LoRa:

4/5

4/6

4/7

4/8

For example, if the code rate is 4/7, for every 4 bits of useful information, the coder generates a total of 7 bits of data, of which 3 bits are redundant.

Making the bandwidth 2x wider (from BW125 to BW250) allows you to send 2x more bytes in the same time. Making the spreading factor 1 step lower (from SF10 to SF9) allows you to send 2x more bytes in the same time.

Lowering the spreading factor reduces the receiver's sensitivity, so the link tolerates less noise and has shorter range, in exchange for faster transmission. You could compare this to two people

taking in a noisy place (a bar for example). If you're far from each other, you have to talk slow (SF10), but if you're close, you can talk faster (SF7)

So, it's balancing act between speed of the transmission and resistance to noise.

things network is mainly focused on LoRaWAN, but the LoRa low-level stuff still checks out for any LoRa project

5.2. Q: Do MeshCore clients repeat?

A: No, MeshCore clients do not repeat. This is the core of MeshCore's messaging-first design. This is to avoid devices flooding the air wave and create endless collisions, so messages sent aren't received. Note the emergency-comms trade-off: because MeshCore clients never repeat, the network depends entirely on surviving repeater infrastructure. In a disaster where repeaters lose power, clients fall back to direct radio range only - plan redundant, powered repeaters.

In MeshCore, only repeaters and room server with `set repeat on` repeat.

5.3. Q: What happens when a node learns a route via a mobile repeater, and that repeater is gone?

A: If you used to reach a node through a repeater and the repeater is no longer reachable, the client will send the message using the existing (but now broken) known path, the message will fail after 3 retries, and the app will reset the path and send the message as flood on the last retry by default. This can be turned off in settings. If the destination is reachable directly or through another repeater, the new path will be used going forward. Or you can set the path manually if you know a specific repeater to use to reach that destination.

In the case if users are moving around frequently, and the paths are breaking, they just see the phone client retries and revert to flood to attempt to re-establish a path.

5.4. Q: How does a node discovery a path to its destination and then use it to send messages in the future, instead of flooding every message it sends like Meshtastic?

Routes are stored in sender's contact list. When you send a message the first time, the message first gets to your destination by flood routing. When your destination node gets the message, it will send back a delivery report to the sender with all repeaters that the original message went through. This delivery report is flood-routed back to you the sender and is a basis for future direct path. When you send the next message, the path will get embedded into the packet and be evaluated by repeaters. If the hop and address of the repeater matches, it will retransmit the message, otherwise it will not retransmit, hence minimizing utilization.

[Source](#)

5.5. Q: Do public channels always flood? Do private channels always flood?

A: Yes. Group/public channels are one-to-many broadcasts with no single destination, so there is no specific path to discover - they must flood. Repeaters can however deny flood traffic up to some hop limit, with the `set flood.max` CLI command. Administrators of repeaters get to set the rules of their repeaters.

[Source](#)

5.6. Q: what is the public key for the default public channel?

A: The smartphone app key is in hex:

```
8b3387e9c5cdea6ac9e5edbaa115cd72
```

T-Deck uses the same key but in base64

```
iz0H6cXN6mrJ5e26oRXNcg==
```

The third character is the capital letter 'O', not zero `0`

[Source](#)

5.7. Q: Is MeshCore open source?

A: Most of the firmware is freely available. Everything is open source except the T-Deck firmware and Liam's native mobile apps.

- Firmware repo: <https://github.com/meshcore-dev/MeshCore>

5.8. Q: How can I support MeshCore?

A: Provide your honest feedback on GitHub and on [MeshCore Discord server](#). Spread the word of MeshCore to your friends and communities; help them get started with MeshCore.

Support Liam Cottle's smartphone client development by unlocking the server administration wait gate with in-app purchase

Support Rastislav Vysoky (recrof)'s flasher web site and the map web site development through [PayPal](#) or [Revolut](#)

5.9. Q: How do I build MeshCore firmware from source?

A: See instructions here:

<https://discord.com/channels/826570251612323860/1330643963501351004/1341826372120608769>

Build instructions for MeshCore:

For Windows, first install WSL and Python+pip via: <https://plainenglish.io/blog/setting-up-python-on-windows-subsystem-for-linux-wsl-26510f1b2d80>

(Linux, Windows+WSL) In the terminal/shell:

```
sudo apt update
sudo apt install libpython3-dev
sudo apt install python3-venv
```

Mac: python3 should be already installed.

Then it should be the same for all platforms:

```
python3 -m venv meshcore
cd meshcore && source bin/activate
pip install -U platformio
git clone https://github.com/meshcore-dev/MeshCore.git
cd MeshCore
```

open platformio.ini and in `[arduino_base]` edit the `LORA_FREQ`. Set `LORA_FREQ` to your region's frequency (e.g. 910.525 for USA/Canada, 867.5 for EU) before building.

save, then run:

```
pio run -e RAK_4631_Repeater
```

then you'll find `firmware.zip` in `.pio/build/RAK_4631_Repeater`

5.10. Q: Are there other MeshCore related open source projects?

A: [Liam Cottle](#)'s MeshCore web client and MeshCore Javascript library are open source under MIT license.

Web client: <https://github.com/liamcottle/meshcore-web>

Javascript: <https://github.com/liamcottle/meshcore.js>

5.11. Q: Does MeshCore support ATAK

A: ATAK is not currently on MeshCore's roadmap.

Meshcore would not be best suited to ATAK because MeshCore:

clients do not repeat and therefore you would need a network of repeaters in place

will not have a stable path where all clients are constantly moving between repeaters

MeshCore clients would need to reset path constantly and flood traffic across the network which could lead to lots of collisions with something as chatty as ATAK.

This could change in the future if MeshCore develops a client firmware that repeats.

[Source](#)

5.12. Q: How do I add a node to the [MeshCore Map](#)

A:

To add a BLE Companion radio, connect to the BLE Companion radio from the MeshCore smartphone app. In the app, tap the `3 dot` menu icon at the top right corner, then tap `Internet Map`. Tap the `3 dot` menu icon again and choose `Add me to the Map`

To add a Repeater or Room Server to the map, go to the Contact List, tap the `3 dot` next to the Repeater or Room Server you want to add to the Internet Map, tap `Share`, then tap `Upload to Internet Map`.

You can use the same companion (same public key) that you used to add your repeaters or room servers to remove them from the Internet Map.

5.13. Q: Can I use a Raspberry Pi to update a MeshCore radio?

A: Yes.

Below are the instructions to flash firmware onto a supported LoRa device using a Raspberry Pi over USB serial.

“ Instructions for nRF devices like RAK, T1000-E, T114 are immediately after the ESP instructions

For ESP-based devices (e.g. Heltec V3) you need:

- Download firmware file from <https://flasher.meshcore.io>
- Go to the web site on a browser, find the section that has the firmware up need
- Click the Download button, right click on the file you need, for example,
`Heltec_V3_companion_radio_ble-v1.7.1-165fb33.bin`
- Non-merged bin keeps the existing Bluetooth pairing database
- `Heltec_v3_companion_radio_usb-v1.7.1-165fb33-merged.bin`
- Merged bin overwrites everything including the bootloader, existing Bluetooth pairing database, but keeps configurations.
- Right click on the file name and copy the link and note it for later use here is an example:
`https://flasher.meshcore.io/releases/download/companion-v1.7.1/Heltec\_v3\_companion\_radio\_ble-v1.7.1-165fb33.bin`
- Run:
- `wget https://flasher.meshcore.io/releases/download/companion-v1.7.1/Heltec_v3_companion_radio_ble-v1.7.1-165fb33.bin` to download the firmware file for your device type. or the version you need - USB, BLE, Repeater, Room Server, merged bin or non-merged bin
- If the above wget command only downloads a very small file (10K bytes instead of more than 100K byte, use this command instead:
- `wget --user-agent="Mozilla/5.0" --content-disposition "https://flasher.meshcore.io/releases/download/companion-v1.7.1/Heltec_v3_companion_radio_usb-v1.7.1-165fb33.bin"`
- Confirm the `ttyXXX` device path on your Raspberry Pi:
- Go to `/dev` directory, run ls command to find confirm your device path
- They are usually `/dev/ttyUSB0` for ESP devices
- For ESP-based devices, install esptool from the shell:
- `pip install esptool --break-system-packages`
- To flash, use the following command:
- For non-merged bin:
- `esptool.py -p /dev/ttyUSB0 --chip esp32-s3 write_flash 0x10000 .bin`
- For merged bin:
- `esptool.py -p /dev/ttyUSB0 --chip esp32-s3 write_flash 0x00000 .bin`

Instructions for nRF devices:

For nRF devices (e.g. RAK, Heltec T114) you need the following:

- Download firmware file from <https://flasher.meshcore.io>
- Go to the web site on a browser, find the section that has the firmware up need
- You need the ZIP version for the adafruit flash tool (below)
- Click the Download button, right click on the ZIP file, for example:
`RAK_4631_companion_radio_ble-v1.7.1-165fb33.zip`
- Right click on the file name and copy the link and note it for later use here is an example:
`https://flasher.meshcore.io/releases/download/companion-v1.7.1/RAK\_4631\_companion\_radio\_ble-v1.7.1-165fb33.zip`
- Run:

- `wget https://flasher.meshcore.io/releases/download/companion-v1.7.1/RAK_4631_companion_radio_ble-v1.7.1-165fb33.zip` to download the firmware file for your device type. or the version you need - USB, BLE, Repeater, Room Server, ZIP file only
- Confirm the `ttyXXX` device path on your Raspberry Pi:
- Go to `/dev` directory, run `ls` command to find confirm your device path
- They are usually `/dev/ttyACM0` for nRF devices
- For nRF-based devices, install `adafruit-nrfutil`
- `pip install adafruit-nrfutil --break-system-packages`
- Use this command to flash the nRF device:
- `adafruit-nrfutil --verbose dfu serial --package RAK_4631_companion_radio_usb-v1.7.1-165fb33.zip -p /dev/ttyACM0 -b 115200 --singlebank --touch 1200`

To manage a repeater or room server connected to a Pi over USB serial using shell commands, you need to install `picocom`. To install `picocom`, run the following command:

- `sudo apt install picocom`

To start managing your USB serial-connected device using `picocom`, use the following command:

- `picocom -b 115200 /dev/ttyUSB0 --imap lfcrLf`

From here, reference repeater and room server command line commands on MeshCore github wiki here:

- <https://github.com/meshcore-dev/MeshCore/wiki/Repeater-&-Room-Server-CLI-Reference>

5.14. Q: Are there are projects built around MeshCore?

A: Yes, there are many. MeshCore's protocol is open source using the MIT license. The MIT license and the open source protocol makes it very easy for the MeshCore community to build new firmware for radios, applications on mobile devices, map tools, and analysis tools, and integration with other projects like Home Assistant.

As new MeshCore community projects become available on a weekly basis, we have stopped tracking them here in this FAQ. [samuk](https://github.com/samuk/awesome-meshcore/blob/main/README.md) maintains a very exhaustive list of MeshCore community project at <https://github.com/samuk/awesome-meshcore/blob/main/README.md>. [samuk](https://github.com/samuk/awesome-meshcore/blob/main/README.md) accepts PRs and merges them regularly.

5.15. Q: Are there client applications for Windows or Mac?

A: Yes, the same iOS and Android client is also available for Windows and Mac. You can find them together with the Android APK here:

<https://files.liamcottle.net/MeshCore>

Both the Windows and Mac versions of the client app are fully unlocked and are free to use.

5.16. Q: Are there any resources that compare MeshCore to other LoRa systems?

A: Here is a list of MeshCore comparison resources:

The Comms Channel on YouTube:

<https://www.youtube.com/watch?v=guDoKGs02Us>

MeshCore Advantages by MCarper:

<https://github.com/mikecarper/meshfirmware/blob/main/MeshCoreAdvantages.md>

[Meshcore vs Meshtastic](#) by austinmesh.org

<https://www.austinmesh.org/learn/meshcore-vs-meshtastic/>

FAQ: 6. Troubleshooting

6.1. Q: My client says another client or a repeater or a room server was last seen many, many days ago.

A: This is almost always a clock/time-sync problem - see the shared answer under 6.2 below, which covers both this "last seen many days ago" symptom and the "not showing up at all" symptom.

6.2. Q: A repeater or a client or a room server I expect to see on my discover list (on T-Deck) or contact list (on a smart device client) are not listed.

A:

- If your client is a T-Deck, it may not have its time set (no GPS installed, no GPS lock, or wrong GPS baud rate).
- If you are using the Android or iOS client, the other client, repeater, or room server may have the wrong time.

You can get the epoch time on and use it to set your T-Deck clock. For a repeater and room server, the admin can use a T-Deck to remotely set their clock (clock sync), or use the `time` command in the USB serial console with the server device connected.

6.3. Q: How to connect to a repeater via BLE (Bluetooth)?

A: You can't connect to a device running repeater firmware via Bluetooth. Devices running the BLE companion firmware you can connect to it via Bluetooth using the android app

6.4. Q: My companion isn't showing up over Bluetooth?

A: make sure that you flashed the Bluetooth companion firmware and not the USB-only companion firmware.

6.5. Q: I can't connect via Bluetooth, what is the Bluetooth pairing code?

A: the default Bluetooth pairing code is `123456`. Note that this is a well-known, fixed default - it is not a secret. For sensitive deployments, be aware that anyone within Bluetooth range (a short

distance) could attempt to pair while the device is in pairing mode.

6.6. Q: My Heltec V3 keeps disconnecting from my smartphone. It can't hold a solid Bluetooth connection.

A: Heltec V3 has a very small coil antenna on its PCB for Wi-Fi and Bluetooth connectivity. It has a very short range, only a few feet. Try repositioning the device closer to the phone first. It is possible to remove the coil antenna and replace it with a 31mm wire (roughly a quarter-wave at 2.4 GHz), and the BT range is much improved with the modification. However, this is an advanced, irreversible mod requiring fine soldering on a tiny PCB antenna feed - it can permanently damage the board and voids the warranty, and soldering an unmatched wire can worsen SWR. Only attempt it if you are experienced.

6.7. Q: My RAK/T1000-E/xiao_nRF52 device seems to be corrupted, how do I wipe it clean to start fresh?

A:

1. Connect USB-C cable to your device, per your device's instruction, get it to flash mode:

- For RAK, click the reset button **TWICE**
- For T1000-e, quickly disconnect and reconnect the magnetic side of the cable from the device **TWICE**
- For Heltec T114, click the reset button **TWICE** (the bottom button)
- For Xiao nRF52, click the reset button once; if that doesn't work, double-click the reset button (two quick presses); if that still fails, disconnect the board from your PC and reconnect again ([seeed studio wiki](#))

1. A new folder will appear on your computer's desktop

2. Download the `flash_erase*.uf2` file for your device on <https://flasher.meshcore.io>

- RAK WisBlock and Heltec T114: `Flash_erase-nRF52_softdevice_v6.uf2`
- Seeed Studio Xiao nRF52 WIO: `Flash_erase-nRF52_softdevice_v7.uf2`

1. drag and drop the uf2 file for your device to the root of the new folder
2. Wait for the copy to complete. You might get an error dialog, you can ignore it
3. Go to <https://flasher.meshcore.io>, click `Console` and select the serial port for your connected device
4. In the console, press enter. Your flash should now be erased
5. You may now flash the latest MeshCore firmware onto your device

Separately, starting in firmware version 1.7.0, there is a CLI Rescue mode. If your device has a user button (e.g. some RAK, T114), you can activate the rescue mode by hold down the user button of the device within 8 seconds of boot. Then you can use the 'Console' on <https://flasher.meshcore.io>

6.8. Q: WebFlasher fails on Linux with failed to open

A: If the usb port doesn't have the right ownership for this task, the process fails with the following error:

```
NetworkError: Failed to execute 'open' on 'SerialPort': Failed to open serial port.
```

Allow the browser user on it:

```
# setfacl -m u:YOUR_USER_HERE:rw /dev/ttyUSB0
```

FAQ: 7. Other Questions:

7.1. Q: How to update nRF (RAK, T114, Seed XIAO) companion, repeater and room server firmware over the air using the new simpler DFU app?

A: The steps below work on both Android and iOS as nRF has made both apps' user interface the same on both platforms:

1. Download nRF's DFU app from iOS App Store or Android's Play Store, you can find the app by searching for `nrf dfu`, the app's full name is `nRF Device Firmware Update`
2. On <https://flasher.meshcore.io>, download the **ZIP** version of the firmware for your nRF device (e.g. RAK or Heltec T114 or Seeed Studio's Xiao)
3. From the MeshCore app, login remotely to the repeater you want to update with admin privilege
4. Go to the Command Line tab, type `start ota` and hit enter.
5. you should see `OK` to confirm the repeater device is now in OTA mode
6. Run the DFU app, tab `Settings` on the top right corner
7. Enable `Packet receipt notifications`, and change `Number of Packets` to 10 for RAK, 8 for T114. 8 also works for RAK.
8. Select the firmware zip file you downloaded
9. Select the device you want to update. If the device you want to update is not on the list, try enabling `OTA` on the device again
10. If the device is not found, enable `Force Scanning` in the DFU app
11. Tab the `Upload` to begin OTA update
12. If it fails, try turning off and on Bluetooth on your phone. If that doesn't work, try rebooting your phone. If you keep getting failures at the "Enabling Bootloader" step, try forgetting the NRF board in your IOS or Andriod device's bluetooth settings and re-pair it through the DFU app.
13. Wait for the update to complete. It can take a few minutes.
14. It is strongly recommended that you install and use the OTAFIX bootloader at https://github.com/oltaco/Adafruit_nRF52_Bootloader_OTAFIX.
15. To update a companion node over OTA, it must be running companion firmware v1.15 or greater.
16. Please see the Meshcore Blog for additional information on OTA firmware flashing:
 - <https://blog.meshcore.io/2026/04/06/otafix-bootloader>
 - <https://blog.meshcore.io/2026/04/02/nrf-ota-update>

7.1.1 Q: Can I update Seeed Studio Wio Tracker L1 Pro using OTA?

A: You can flash this safer bootloader to the Wio Tracker L1 Pro

https://github.com/oltaco/Adafruit_nRF52_Bootloader_OTAFIX

After this bootloader is flashed onto the device, you can trigger over the air update using bluetooth by holding the button next to the D-Pad and then click the reset button. Then follow the same OTA update instructions above. You can skip past the `start ota` instruction and start the update using the DFU app.

7.2. Q: How to update ESP32-based devices over the air?

A: For ESP32-based devices (e.g. Heltec V3):

1. On <https://flasher.meshcore.io>, download the **non-merged** version of the firmware for your ESP32 device (e.g. `Heltec_v3_repeater-v1.6.2-4449fd3.bin`, no `"merged"` in the file name)
2. From the MeshCore app, login remotely to the repeater you want to update with admin privilege
3. Go to the Command Line tab, type `start ota` and hit enter.
4. you should see `OK` to confirm the repeater device is now in OTA mode
5. The command `start ota` on an ESP32-based device starts a wifi hotspot named `MeshCore OTA`
6. From your phone or computer connect to the 'MeshCore OTA' hotspot
7. From a browser, go to <http://192.168.4.1/update> and upload the non-merged bin from the flasher

7.3. Q: Is there a way to lower the chance of a failed OTA device firmware update (DFU)?

A: Yes, developer `oltaco` has an enhanced OTA DFU bootloader for nRF52 based devices. With this bootloader, if it detects that the application firmware is invalid, it falls back to OTA DFU mode so you can attempt to flash again to recover. This bootloader has other changes to make the OTA DFU process more fault tolerant.

Refer to https://github.com/oltaco/Adafruit_nRF52_Bootloader_OTAFIX for the latest information.

Currently, the following boards are supported:

- Heltec Automation Mesh Node T114 / HT-nRF5262
- Nologo ProMicro NRF52840 (aka SuperMini NRF52840)
- Seeed Studio SenseCAP Card Tracker T1000-E
- Seeed Studio Wio Tracker L1
- Seeed Studio XIAO nRF52840 BLE
- Seeed Studio XIAO nRF52840 BLE SENSE

- RAK 4631
- RAK WisMesh Tag (new 28/11/2025)

7.4. Q: are the MeshCore logo and font available?

A: Yes, it is on the MeshCore github repo here:

<https://github.com/meshcore-dev/MeshCore/tree/main/logo>

7.5. Q: What is the format of a contact or channel QR code?

A:

Channel:

```
meshcore://channel/add?name=&secret=
```

Contact:

```
meshcore://contact/add?name=&public_key=&type=
```

where `&type` is:

```
chat = 1
```

```
repeater = 2
```

```
room = 3
```

```
sensor = 4
```

7.6. Q: How do I connect to the companion via WIFI, e.g. using a heltec v3?

A:

WiFi firmware requires you to compile it yourself, as you need to set the wifi ssid and password.

Edit WIFI_SSID and WIFI_PWD in `./variants/heltec_v3/platformio.ini` and then flash it to your device.

7.7. Q: I have a Station G2, or a Heltec V4, or an [Ikoka Stick](#), or a radio with a EByte E22-900M30S or a E22-900M33S module, what should their transmit power be set to?

A:

For companion radios, you can set these radios' transmit power in the smartphone app. For repeater and room server radios, you can set their transmit power using the command line command `set tx`. You can get their current value using command line comand `get tx`

⚠ WARNING: Set these values at your own risk. Incorrect power settings can permanently damage your radio hardware.

⚠ US FCC compliance — read before setting power: In the United States, FCC Part 15.247 caps the conducted output of unlicensed 902–928 MHz LoRa devices at **1 W (30 dBm)** and effective isotropic radiated power (EIRP) at **4 W (36 dBm)** (the 1 W conducted limit assumes an antenna of up to 6 dBi gain; gain above 6 dBi requires a dB-for-dB reduction in conducted power). **Any setting above 30 dBm conducted is not legal for unlicensed US operation**, and pushing the radio's power amplifier to its hardware maximum can also damage the PA. Keep US settings at or below 30 dBm conducted (1 W), minus any antenna gain over 6 dBi. The rows in the table below that exceed 30 dBm are marked accordingly. Outside the US, follow your own region's limits — note that EU868 limits are far lower (typically 25 mW / 14 dBm ERP on most sub-bands), so the high-output figures shown for EU868 below are well above what EU rules permit and are listed only to document the hardware's capability.

Note on the "In-App Setting" column: The value you enter in the app or via `set tx` is a module-internal index, *not* the actual emitted power. It maps to the real radio output in a non-obvious, non-linear way (for example, an in-app value of 19 dBm on a Station G2 produces about 36.5 dBm of actual output, while 22 dBm in-app on a Heltec V4 produces only about 28 dBm). Always judge legality and safety by the **Target Radio Output** column, not the in-app number. The actual-output figures below come from community/vendor measurements; confirm against your specific board's datasheet before relying on them.

Device / Model	Region / Description	In-App Setting (dBm)	Target Radio Output	Notes
Station G2 Reference	US915 Max Output	19 dBm	36.5 dBm (4.46W) — EXCEEDS US FCC Part 15 limit (not legal for unlicensed US use)	
	US915 Max at 1dB compression point	16 dBm	35 dBm (3.16W) — EXCEEDS US FCC Part 15 limit (not legal for unlicensed US use)	1dB compression point
	EU868 Max at 1dB compression point	15 dBm	34.5 dBm (2.82W) — exceeds the US FCC Part 15 conducted limit, and is far above EU868 ERP limits (EU/other regions only — not legal at this level in the EU)	1dB compression point

Device / Model	Region / Description	In-App Setting (dBm)	Target Radio Output	Notes
	US915 1W Output	10 dBm	1W	Refer to your local government's requirements
	EU868 1W Output	9 dBm	1W	Refer to your local government's requirements
Ikoka Stick E22-900M30S	1W Model	19 dBm	1W	DO NOT EXCEED (Risk of burn out) data sheet
Ikoka Stick E22-900M33S	2W Model	9 dBm	2W — EXCEEDS US FCC Part 15 limit (not legal for unlicensed US use)	DO NOT EXCEED (Risk of burn out) data sheet Refer to your local government's requirements
Heltec V4	Standard Output	10 dBm	22 dBm (~0.15W)	
	High Output	22 dBm	28 dBm (~0.5W to 0.6W)	

Troubleshooting Guide

My node cannot connect to others

Work through these checks in order.

1. **Verify antenna is connected.** Most common cause. Never transmit without an antenna attached - it can permanently damage the radio's power amplifier (the PA, the transmit chip).
2. **Verify modem preset matches.** Two nodes on different presets are invisible to each other. On Meshtastic, check the [Meshtastic app](#) - Radio Config - LoRa - Modem Preset; on MeshCore, check the radio/preset settings in the MeshCore app. Ask your local community which preset they use.
3. **Verify channel name and PSK match.** Wrong credentials mean your messages are encrypted to a key nobody else has. Verify against local network documentation.
4. **Verify frequency band.** US/Canada = 915 MHz hardware. EU = 868 MHz hardware. These cannot interoperate. Many AliExpress boards ship as 868 MHz by default.
5. **Check if other nodes are in range.** Meshtastic users: visit meshmap.net to see nearby nodes. MeshCore users: use the MeshCore map at map.meshcore.io. In rural areas you may genuinely be out of range.
6. **Verify the radio is transmitting.** Check for LoRa init errors in the serial console. A working node shows increasing packet counts in the app.
7. **Factory reset and reconfigure.** If all else fails, reset and start configuration fresh.

My messages are not getting through

First: identify the failure mode

- Node visible in list but messages do not arrive: routing or range issue
- Node does not appear in list at all: preset, channel, or range issue (see previous page)

Hop limit too low

If your hop limit is set to 1, you can only reach nodes in direct radio range. Increase it (Meshtastic: Radio Config → LoRa → Hop Limit; default 3 is fine for most networks). Note: "hop limit" is the cap you set, while "hop count" is the number of hops a packet has already taken.

Network congestion

If channel utilization exceeds 25% (visible in [Meshtastic app](#)), collisions become more likely. 25% is a soft "green/optimal" ceiling rather than a hard collision cliff — Meshtastic firmware defers transmissions until channel utilization drops back below roughly 25%. Reduce your broadcast intervals for position and telemetry to ease congestion.

ACK failures

Meshtastic uses ACKs to confirm delivery. No ACK means the destination node may be offline or out of range. Check its last-heard timestamp. **Important for life-safety traffic:** a received ACK does not guarantee the intended recipient got the message. On a multi-hop path the ACK you see may be an implicit ACK from a relay rebroadcasting the packet, not a confirmation from the final destination. The mesh also has a hard shared-airtime ceiling with no message prioritization (no

QoS), so urgent traffic competes with everything else and congestion spikes exactly when volume rises. For any critical message, require an explicit human reply rather than trusting the ACK indicator.

Recently joined node

New nodes may not have propagated throughout the mesh yet. Wait 5-10 minutes after joining before expecting full connectivity.

MeshCore specific

MeshCore requires successful route discovery before first contact. If intermediate repeaters are offline, the route cannot be established. Check that all repeaters in the expected path are online.

My battery drains too fast

Step 1: Measure actual power draw

Use a USB inline power meter to measure real current draw. This immediately shows whether you have a software config problem or a hardware issue.

Expected draw by hardware

The figures below are approximate field measurements and vary with firmware, settings, and transmit duty cycle - measure your own device for an accurate number.

Hardware	Typical active draw (approximate)
ESP32 (T-Beam, Heltec), no display, no BT	~40-55 mA
ESP32 with OLED display on	+10-20 mA
nRF52840 (RAK4631, T-Echo, T114)	~8-15 mA

Power drain checklist

- Screen on all the time? Set a short screen timeout (e.g. 10-30 seconds) so the display sleeps. Note: a timeout of 0 usually means never sleep - avoid it on battery.
- Bluetooth enabled? Disable if not needed.
- GPS polling? Disable or set a long interval.
- WiFi enabled (ESP32)? Disable.
- Position broadcast interval too short? Set to 30+ minutes on battery.

Battery sizing

Battery (mAh) divided by draw (mA) = hours of runtime. 1000 mAh at 40 mA = 25 hours; at 10 mA = 100 hours. Switching from ESP32 to nRF52840 hardware typically gives roughly 4x longer life on the same battery (approximate; the exact ratio depends on your settings and usage).

My solar node keeps going offline

Systematic diagnosis in order of likelihood:

1. **Undersized battery or panel.** Goes offline at night or on cloudy days? Review the [Solar System Sizing Guide](#) and resize.
2. **Panel obstruction.** Bird droppings, snow, or vegetation growth. Verify the panel is clean with clear sky access.
3. **Charge controller LVD setting.** If the low-voltage disconnect threshold is too high, the controller cuts power while charge remains. For LiFePO₄, 3.0V/cell (12.0V for a 12V system) is a conservative LVD; the typical BMS cutoff is lower (~2.5V/cell, ~10V). Set the LVD per your specific battery and charge-controller specification.
4. **Water ingress.** Goes offline after rain = water ingress. Inspect all cable glands, connector weatherproofing, and enclosure seals.
5. **Thermal shutdown.** Sealed enclosures in direct sun can exceed safe operating temperatures (often reported in the 70-80 degrees C range internally). Add venting, shading, or a radiation shield.
6. **Firmware hang.** Check for firmware updates. Enable hardware watchdog if supported.
7. **Remote monitoring.** Watch battery voltage trend via Meshtastic telemetry. Declining trend = insufficient harvest. Sudden drop to zero = hardware failure.

Common Beginner Questions

Do I need an amateur radio license?

Short answer: No

Meshtastic and MeshCore operate in the 902–928 MHz ISM band under FCC Part 15 rules (47 CFR § 15.247 and related provisions). No amateur radio license is required for standard operation in the United States — meaning the device as configured by the official apps for your region, with licensed/ham mode left off. Note that unlicensed Part 15 operation carries conditions: your device must not cause harmful interference and must accept any interference received (47 CFR § 15.5).

Why many operators are licensed hams

The amateur radio community has been an early adopter of LoRa mesh. In standard (default, encrypted) configuration the protocols operate under Part 15, not Part 97 (amateur rules), and a license changes nothing about that mode. However, 902–928 MHz is also the amateur **33 cm band**, so licensed hams can optionally operate mesh radios under Part 97 with more power and higher-gain antennas — subject to the restrictions below. You don't need a license to use the mesh; a license adds optional capabilities.

When a license IS relevant

- **Operating above Part 15 power limits:** Under Part 15 the limit on 902–928 MHz is 1 W (30 dBm) conducted output for a digitally-modulated system, with an antenna gain up to 6 dBi (a 36 dBm / 4 W EIRP ceiling); above 6 dBi of gain you must reduce conducted power dB-for-dB, so you cannot simply add gain to raise EIRP. Exceeding these Part 15 limits is not permitted at all under Part 15 — and an amateur license does NOT authorize running a Part 15 device over its certified limits on the ISM band. Licensed amateurs may instead

operate on 902–928 MHz under *Part 97* with more power (up to 10 W PEP for spread spectrum, 47 CFR § 97.313(j)) — **but only with encryption disabled (Part 97 prohibits encrypting messages to obscure their meaning, § 97.113(a)(4)) and with call-sign identification at the end of each communication and at least every 10 minutes during it (§ 97.119)**. Default Meshtastic and MeshCore channels are encrypted, so **you cannot run encrypted mesh channels at higher amateur power, and raising power on a stock configuration is illegal even with a license**.

Meshtastic's "licensed (ham) mode" handles this: it sets your callsign, transmits it periodically, and disables encryption. Standard mesh stays under Part 15 limits.

- **Amateur-band operation generally:** Transmitting LoRa hardware under amateur privileges — on the 33 cm band or any other amateur allocation — requires an amateur license and full Part 97 compliance.
- **APRS gateway:** Feeding Meshtastic traffic into APRS-IS (the internet backbone of the ham APRS network) requires a valid amateur radio callsign — APRS-IS access is restricted to licensed amateurs. Transmitting APRS over RF (144.390 MHz in North America) additionally requires at least a Technician-class license under Part 97.

Canada

Similar licence-exempt rules apply in Canada under ISED **RSS-247** (with RSS-Gen general requirements). No license is needed for standard Meshtastic/MeshCore operation. The license-relevant cases above are stated in US (FCC) terms; the Canadian amateur equivalent is the Amateur Radio Operator Certificate issued by ISED.

How far can my node reach?

Range varies enormously by terrain, antenna height, modem preset, and local conditions.

Realistic ranges

The figures below are approximate field estimates gathered from community experience, not hardware specifications or guaranteed numbers. Actual range depends heavily on antenna height at both ends, modem preset (e.g. LongFast vs. a slower long-range preset), Fresnel-zone clearance, and line of sight. Treat them as rough order-of-magnitude planning aids, and always confirm with a real on-site range test. The longer brackets assume clear line of sight between elevated antennas; the over-water figure in particular is an exceptional best-case shot, not a typical result.

Scenario	Typical Range (approximate, line-of-sight dependent)
Ground level, dense urban	200 m - 1 km
Ground level, suburban	0.5 - 2 km
Rooftop or upper floor (line of sight)	2 - 8 km
Hilltop to hilltop, clear line of sight	10 - 30 km
Mountain repeater to valley (clear line of sight)	15 - 40 km
Over water (lake, bay), unobstructed line of sight	30 - 80+ km (exceptional best case, not typical)

The dominant factor: antenna height

Radio waves travel in straight lines. Elevation beats everything else. Even 3 extra meters of height meaningfully extends the radio horizon. A mediocre antenna on a hilltop outperforms an excellent antenna at ground level.

Best approach: test it

Deploy the node, walk or drive around with a second device, and observe where packets stop arriving. The Meshtastic [Range Test module](#) (Config - Module - Range Test) automates this by logging position and signal data to a CSV file. Because the figures above are only approximate, an on-site range test is the only reliable way to know your actual coverage.

MeshCore or Meshtastic - which should I choose?

Both are excellent. Choose based primarily on what your local community already uses. Don't pick a protocol solely on technical merits (routing efficiency, encryption) if it means being alone on the mesh - community presence matters more.

Choose Meshtastic if:

- Your local area already has an active Meshtastic network
- You want the largest app ecosystem and community support
- You want the most beginner-friendly experience with polished apps
- You want the widest hardware compatibility

Choose MeshCore if:

- Your local community uses MeshCore
- You are building dedicated repeater infrastructure
- You prefer MeshCore's selective routing model for direct messages (both MeshCore and current Meshtastic use public-key encryption for direct messages; neither provides Signal-style forward secrecy)
- You are deploying a large network (50+ nodes) where flooding creates congestion

Can they interoperate?

No. Different packet formats and routing protocols. A MeshCore node and a Meshtastic node cannot communicate even on the same frequency. Many operators run both on separate devices for different purposes.

Technical comparison

Feature	Meshtastic	MeshCore
Routing	Flooding	Path-based (path discovery)
Community size	Larger	Smaller, more technical
Hardware support	Very broad	Good (915 MHz boards)
Best for	Personal use, joining existing mesh	Infrastructure, large networks

Hardware FAQ

Which board should I buy as a beginner?

Short answer: Heltec WiFi LoRa 32 V3 or T-Beam Supreme

For most beginners in North America, either of these is an excellent first choice:

Board	Price (approx)	Best for	Notes
Heltec WiFi LoRa 32 V3	\$18-25	Lowest cost, quick start	Small OLED display built-in; ESP32-S3; USB-C charging; no GPS
LILYGO T-Beam Supreme	\$35-45	All-in-one handheld	Integrated GPS; large battery connector; OLED display; good antenna connector
RAK WisBlock Starter Kit	\$40-55	Best battery life	nRF52840-based; excellent for portable use; modular expansion
T-Echo	\$55-65	Best handheld device	E-ink display; GPS; nRF52840; weeks of battery; premium feel

Prices are approximate and change over time; confirm against a current listing before buying. (Note: the "T-Beam Supreme" is a distinct, newer product from the older plain "T-Beam" - check that the listing names the exact model you want.)

What to Avoid as a Beginner

- **868 MHz boards** - Common on AliExpress. Check the listing carefully for "915 MHz" or "US version." 868 MHz hardware will not work on North American mesh networks.

- **No-name ESP32 LoRa clones** - Very cheap boards with no external antenna connector. Beyond poor range from the PCB trace antenna, a board whose LoRa front-end is not properly matched can present a poor SWR, which stresses the radio's power amplifier when transmitting. Choose a board with a real SMA or u.FL antenna port and a properly matched RF front-end - it's worth the extra \$5.
- **LoRaWAN gateways** - These are different from LoRa mesh nodes and will not run Meshtastic or MeshCore. Check for "Meshtastic compatible" in the listing.

Where to Buy Reliably

- **Amazon** - Faster shipping; check seller carefully; returns are easy.
- **Official distributors** - Rokland (US), Heltec official store (AliExpress), LILYGO official store (AliExpress). Ships from source but slower delivery from China. Prices are approximate and change; lookalike "official store" listings are common, so confirm a seller is the genuine official store by following the link from the manufacturer's own website.
- **Local ham radio events and swaps** - Other operators often sell tested hardware at reasonable prices.

Do I Need to Buy an Antenna?

The stock rubber duck antenna included with most boards is functional but not optimal. For a handheld or portable device, the included antenna is fine. For a fixed repeater or rooftop node, upgrading to a fiberglass vertical (5-6 dBi) on a short coax run provides significantly better performance. See the Antennas & RF book for details.

Before you mount anything outdoors: a rooftop node needs proper grounding and a coax lightning arrestor, and roof work carries fall and overhead power-line hazards. Read the antenna-installation safety guidance in the Antennas & RF book before installing.

Why isn't my GPS getting a fix?

GPS fix time expectations

Getting a GPS fix takes time - particularly on a cold start (first power-on or after being stored). Under normal outdoor conditions:

Condition	Expected fix time
Cold start, outdoors, clear sky	30 seconds - 5 minutes
Warm start (powered recently)	5-15 seconds
Indoors, near window	2-10 minutes (often fails entirely)
Indoors, no window	Will not get a fix

Common Causes of No GPS Fix

Indoors

GPS signals are extremely weak (-130 dBm from satellites). They are heavily attenuated by buildings and concrete and effectively blocked by metal. High-sensitivity receivers can sometimes get a fix near a window, but reliable acquisition requires going outdoors with a clear view of the sky.

Antenna not connected or loose

Boards with a U.FL GPS antenna connector (T-Beam, some RAK boards) have a small snap-fit U.FL connector between the GPS module and the antenna. This connector can come loose in shipping. Open the case and verify the connector is fully seated - it should click into place.

GPS module disabled in firmware

If you (or a previous configuration) disabled GPS to save power, the module will not attempt to acquire satellites. For most users, the easiest path is the app: **Radio Config** → **Position** → **GPS Mode** → set to Enabled.

Advanced (requires the Meshtastic command-line tool, a separate install, with the device connected over USB):

```
meshtastic --set position.gps_mode ENABLED
```

Power brownout

If the GPS module is not receiving adequate voltage (often seen when running from a nearly depleted battery), it may fail to initialize. Try with a freshly charged battery or USB power.

AGPS data expired

Some boards support AGPS (Assisted GPS), which provides almanac data to speed up acquisition. If the AGPS data is stale (more than 2 weeks old), fix time increases significantly. Re-upload AGPS data via the app when connected to the internet.

Testing GPS Without Going Outside

Place the device on an outdoor windowsill with the GPS antenna facing up toward the sky. This is the minimum viable indoor setup for GPS - results vary significantly by window direction and whether metal frames or low-E glass are present.

T-Beam GPS Note

GPS modules vary by T-Beam variant: older T-Beam v1.x units use a Quectel L76K or NEO-6M module, while the newer T-Beam Supreme uses a different module and layout - so the specifics below apply mainly to the classic T-Beam, not necessarily the Supreme. The NEO-6M found in older T-Beam versions has been discontinued; Quectel L76K in newer units is generally faster to acquire. The GPS is connected to the ESP32 via UART2 - if you see GPS-related errors in the serial console, check that the GPS power is enabled (some T-Beam versions have a GPS power pin that must be asserted).

Can I use my node inside my house or vehicle?

Short Answer

Yes, with significant range reduction. Interior use is practical for connecting to a nearby outdoor repeater or for testing. It's not suitable as a repeater location.

What Signal Loss to Expect

The figures below are rough, environment-dependent estimates at ~900 MHz, not precise measurements - actual building and vehicle penetration loss varies widely with construction, materials, frequency, and geometry. Published 900 MHz studies (e.g., NTIA Report 94-306 and vendor app notes) report mean building-penetration losses broadly in line with these ranges (for example ~4 dB for a single wood/drywall wall and ~10-12 dB for a typical interior or single concrete wall), but treat any single number as illustrative.

Location	Typical Signal Loss (estimate)	Notes
Near a window, wood frame house	~3-6 dB	Manageable; roughly equivalent to halving your range
Interior room, wood frame	~6-15 dB	Significant; may still reach nearby repeaters
Concrete/brick building	~10-25 dB	Severe; the low end reflects a single wall, the high end multiple walls or whole-building paths; may not reach anything without a nearby repeater
Metal building, basement	~20-40+ dB	Qualitative estimate (Faraday-cage effect); effectively unusable for mesh
Vehicle (windshield path)	~3-8 dB (rough estimate)	Acceptable for personal use; mount near windshield
Vehicle (metal roof path)	~20-30 dB (rough estimate)	Much worse; magnetic mount external antenna required

Improving Indoor Performance

- **Windowsill placement** - Even 6 inches from a window vs deep in a room makes a measurable difference. Place the node as close to a window facing the direction of the nearest repeater as possible.
- **External antenna on a cable** - Many setups run the node indoors with a short coax to a small external antenna mounted outside or near a window. With genuine low-loss cable (LMR-240/400 class), 3-5 meters costs under 1 dB of loss and puts the antenna in a dramatically better RF environment. Note that thin/cheap coax such as RG-58, RG-174, or RG-316 will lose considerably more than 1 dB over 5 meters at 915 MHz - use good cable.
- **Higher floor** - Upper floors have less obstruction from building materials and more line-of-sight above street-level clutter. A third-floor window is significantly better than a ground-floor window.

Vehicle Use

A node placed on the dashboard or near the windshield can typically receive and send to nearby repeaters. For best vehicle performance:

- Mount near the windshield on the upper dash, antenna pointing up. **Safety:** position the device so it does not obstruct your view of the road and is clear of airbag deployment areas, and secure it so it cannot become a projectile in a collision.
- For dedicated vehicle installations, use a magnetic mount external antenna on the roof. NMO or SMA-compatible magnetic mounts are available, but confirm the antenna *element itself* is tuned for 902-928 MHz - many magnetic-mount/NMO antennas are cut for cellular or VHF/UHF and will present a poor SWR at 915 MHz even though the connector fits.
- Power from the 12V accessory port via a USB adapter

Solar and Power FAQ

How big a solar panel do I need?

Short Answer

For most LoRa mesh nodes: a 5W panel for nRF52840-based nodes, 10-20W for ESP32-based nodes. For Raspberry Pi gateways: 20-40W. These are rule-of-thumb guideline ranges, not sourced specifications - actual requirements depend heavily on your node's duty cycle, display use, and local insolation. They assume worst-month sun and a few days of autonomy; cross-check against Meshtastic's solar-powered node documentation and a solar sizing calculator for your specific site.

The Calculation

Solar system sizing is a four-step calculation. The current-draw figures below are approximate and config-dependent (Wi-Fi/BLE duty cycle and display state move them substantially) - measure your own node where you can rather than trusting these numbers:

1. **Measure your node's current draw** - Use a USB inline power meter. Real measurements beat estimates. Typical (approximate) values: nRF52840 repeater: 8-15 mA average; ESP32 repeater with OLED: 40-70 mA; Pi Zero 2W gateway: 100-150 mA. An nRF52840 repeater can average well below 15 mA, and ESP32 draw varies widely with Wi-Fi.
2. **Calculate daily energy consumption** - $\text{Current (mA)} \times 24 \text{ hours} = \text{mAh per day}$.
Example: $12 \text{ mA} \times 24 = 288 \text{ mAh/day} = 0.288 \text{ Ah/day}$.
3. **Find peak sun hours for your location** - This is the key local variable. A panel receives "peak sun hours" as an energy-equivalent of full-rated output. Approximate US annual-average values (source these from NREL PVWatts/NSRDB for your exact location, as they vary by dataset): Miami: 5.5, Denver: 5.3, Seattle: 3.6, Boston: 4.2, Phoenix: 6.1. **Use the worst-month value for sizing, not the annual average.** The worst month is typically December, and December peak sun hours are far lower than the annual figure - e.g. Seattle drops to roughly 1-1.5 PSH in December, not 3.6. Size for that worst month.
4. **Panel size** - A rough starting formula is: $\text{Daily consumption (Ah)} \div \text{worst-month peak sun hours} \times \text{a combined derate of about } 1.5\text{-}2 \times = \text{panel Ah output needed}$, then convert to

Watts at your system voltage. A single 1.25 "efficiency factor" is too optimistic: real-world losses (charge-controller and battery inefficiency, temperature, soiling/dust, panel aging, off-angle mounting, and only 50-70% of rated panel watts actually harvested in winter) commonly total 1.5-2x or more. The bare arithmetic for an ultra-low-power nRF52840 node yields a tiny number (well under 1 Wh/day), but **do not use that as your panel size** - it ignores those losses and multi-day overcast. After accounting for them, **a 5W panel is the practical minimum even for the smallest nodes** in poor-insolation regions like the Pacific Northwest in winter. Size up further for ESP32 and Pi-class nodes per the table below.

Practical Sizing Recommendations

The current draws and panel sizes below are approximate engineering estimates, not measured specifications. The PNW-winter column assumes you must ride through consecutive overcast days; in the worst sites you may need more than the listed wattage. For solar nodes, disable or duty-cycle the OLED display - leaving it on continuously wastes power for no benefit on an unattended repeater.

Node Type	Average Draw	Panel (temperate US)	Panel (PNW winter)
nRF52840 repeater, no display	10-15 mA	5W	5W (consider 10W for consecutive overcast days)
ESP32 repeater, no display	40-55 mA	10W	20W
ESP32 repeater, OLED on	60-80 mA	15W	30W
Pi Zero 2W + LoRa HAT	120-160 mA	20W	40W (a Pi gateway suits small solar poorly; may need more)

Battery Sizing

Size the battery for 3-5 days of autonomy (no solar input) for ordinary hobby uptime. This covers typical cloudy periods and seasonal weather. **For nodes intended as emergency infrastructure, size for 7-10+ days of autonomy** - the multi-day winter storms, wildfire smoke, and heavy overcast that most demand a working mesh are exactly when solar harvest collapses, and a node sized for only 3-5 days can die mid-incident. Battery (Ah) = Daily consumption (Ah) × autonomy days × a depth-of-discharge multiplier. The right multiplier depends on chemistry: use about 1.25x for LiFePO4 (to stay near 80% usable depth of discharge) and roughly 2x for lead-acid (which wants ~50% depth of discharge for reasonable life). A flat 1.2x is too aggressive for healthy long-term cycling.

Example: nRF52840 at 12 mA $\times$ 24h = 0.29 Ah/day $\times$ 5 days $\times$ 1.25 (LiFePO4) $\approx$ 1.8 Ah minimum.

For permanent outdoor nodes, use a LiFePO4 pack, not a bare LiPo - LiPo must not be charged below 0°C, suffers freeze damage, and is a fire risk in an unattended sealed enclosure (see the battery-chemistry guidance). Note that LiFePO4 cells are $\sim$ 3.2V nominal, not 3.7V like LiPo, so a true "equivalent" differs in voltage and watt-hours; size by capacity (Ah) and chemistry, and pick a pack with comfortable margin (e.g. a few thousand mAh for this example).

Solar & Wiring Safety

For any permanent battery + panel install: **always use a proper charge controller** between the panel and the battery - never wire a solar panel directly to a LiFePO4 (or LiPo) battery, as that risks overcharge, cell damage, and fire. Add an inline fuse at the battery positive terminal sized to your wiring; a shorted lithium pack can dump very high current into thin DIY wiring inside a sealed enclosure. In any climate with freezing winters, use a charge controller or BMS with a low-temperature charge cutoff - lithium chemistries must not be charged below 0°C.

Why does my solar node keep dying at night?

Diagnosing Night Drain

If your solar node runs fine during daylight but goes offline overnight, you have one of three problems: undersized battery, incorrect charge controller settings, or excessive power draw.

Step 1: Verify Actual Battery Capacity

First, measure what you actually have. LiPo and LiFePO₄ batteries are frequently sold at optimistic ratings. A "5000 mAh" LiPo from an unknown manufacturer may test at 3000-3500 mAh in practice.

Test: Fully charge the battery, disconnect solar, and measure how long the node runs. Node runtime (hours) × current draw (mA) = actual battery capacity (mAh).

Step 2: Calculate Required Overnight Capacity

In winter at high latitudes, "night" can mean 16+ hours of darkness. Your battery must cover: hours of darkness × node current draw.

Example: 14 hours dark, 12 mA draw = 168 mAh minimum. On paper even a small 1000 mAh battery covers this. But this bare calculation is optimistic: it ignores depth-of-discharge limits (you should not regularly drain a cell to empty), cold-temperature capacity loss (a 1000 mAh cell delivers substantially less at -10°C), and multi-day low-/no-sun events. For emergency-grade reliability, oversize the battery well beyond the textbook minimum and account for cold-weather

derating and several consecutive cloudy days. If your node dies overnight, either the battery is not fully charging during the day or the current draw is much higher than expected.

Step 3: Verify the Battery is Actually Charging

Check the charge controller status LED or voltage output during the day. Common issues:

- **Panel undersized for the season** - A panel just barely adequate in summer may not fully recharge the battery on short winter days. The battery reaches discharge before it can recover overnight.
- **Panel partially shaded** - Even partial shading (one corner of the panel) can sharply reduce output - often on the order of 50-80%, though the exact loss depends heavily on how the cells are wired in series and whether the panel has bypass diodes. Because series-connected cells are limited by the most-shaded cell, shading even a small area can disproportionately cut output. Inspect during the time of day the sun is lowest (winter morning/afternoon).
- **LVD set too high** - If the charge controller's Low Voltage Disconnect threshold is set above the battery's actual minimum, it cuts power while charge still remains. A conservative LVD that protects pack longevity is 3.0V/cell for LiFePO4 (12.0V for 4S) and 3.0-3.2V/cell for LiPo; the absolute safe-discharge floor is lower (~2.5V/cell for LiFePO4, ~2.5-3.0V/cell for LiPo). Always defer to your specific cell/pack datasheet for the recommended cutoff.
- **Charge controller in wrong battery mode** - Lithium chemistries need a matching lithium charge profile. A controller configured for lead acid uses different absorption/float voltages and cutoffs, which can overcharge a LiPo or leave a LiFePO4 pack undercharged. Verify the battery type setting matches your battery chemistry, and check the actual charge voltages against your cell datasheet.

Step 4: Reduce Power Draw

If the battery and panel are correctly sized but the node still dies, attack the power draw. The savings below are approximate and vary with board, configuration, and firmware version; measure your own node where possible:

- Disable OLED display (saves roughly 15-20 mA on a typical small OLED, depending on brightness and how many pixels are lit)
- Disable Bluetooth (saves roughly 5-15 mA on ESP32, depending on advertising interval)
- Switch from ESP32 to nRF52840 board (saves roughly 30-50 mA average, depending heavily on configuration and sleep behavior)

- Reduce TX power to minimum needed. Note: lowering TX power reduces transmit-state energy, but not in a clean "halve the current per 3 dB" relationship - a 3 dB cut halves the radiated RF power, but PA efficiency is non-linear and the supply current does not scale 1:1. For a typical low-duty-cycle mesh node, transmit is only a small fraction of average draw, so the overnight benefit of backing off TX power is usually modest compared with display, Bluetooth, and idle draw.

What battery chemistry should I use outdoors?

Short Answer: LiFePO4 for outdoor deployments

Lithium Iron Phosphate (LiFePO₄) is the recommended battery chemistry for any permanent outdoor LoRa mesh installation. It is safer, more durable, and handles temperature extremes better than standard LiPo batteries. "Better at temperature extremes" mainly means greater high-temperature thermal stability and good cold-temperature *discharge* tolerance - it is not unlimited, and LiFePO₄ still must not be charged below 0°C (see below).

Why Not LiPo?

LiPo (Lithium Polymer) batteries are the default on most development boards because they're inexpensive and compact. For an outdoor deployment, they have serious limitations:

- **Temperature sensitivity** - LiPo loses significant capacity below 0°C and should never be charged below 0°C (causes internal lithium plating and eventual failure). In any climate with freezing winters, an outdoor LiPo battery will degrade rapidly or fail entirely within 1-2 seasons.
- **Thermal runaway risk** - LiPo batteries can catch fire if punctured, overcharged, or deeply discharged and then recharged. Not ideal in unattended outdoor enclosures.
- **Short cycle life** - 300-500 full charge cycles. A solar node cycling daily would exhaust a LiPo in 1-1.5 years.

LiFePO4 Advantages

Property	LiPo	LiFePO ₄
----------	------	---------------------

Operating temperature	0°C to 45°C	-20°C to 60°C (discharge)*
Cycle life	300-500 cycles	2,000-4,000 cycles
Thermal runaway	Yes (fire risk)	Much more resistant; thermal runaway is far less likely but not impossible under severe abuse/very high temperature
Nominal voltage	3.7V/cell	3.2V/cell
Energy density	~150-200 Wh/kg	~90-130 Wh/kg
Cost	Lower	Higher (but lower cost per cycle)

***Operating (discharge) range is -20°C to 60°C, but LiFePO4 must NOT be charged below 0°C without a low-temperature charge cutoff in the BMS** - sub-freezing charging causes lithium plating, which permanently damages the cell.

LiFePO4 Products for LoRa Deployments

Product lines and model numbers below are examples and change over time; verify chemistry, exact model number, and current availability against the vendor's catalog before buying (as of June 2026). Make sure any pack you select is genuinely LiFePO4 - some "lithium" jump-start and powersports packs are standard Li-ion, not LiFePO4.

- **Small cells (3.2V)** - EVE LF50K, EVE LF100 18650-format cells; use with a LiFePO4-compatible BMS
- **Integrated packs** - Bioenno 3.2V to 12.8V packs with built-in BMS; Dakota Lithium packs
- **12V packs** - For larger systems with 12V charge controllers; Battle Born and Dakota Lithium offer 12V LiFePO4 packs in roughly 10-100Ah sizes (confirm the exact current model number with the vendor)

Charge Controller Compatibility

LiFePO4 requires a charge controller set to LiFePO4 chemistry. Lead acid charge profiles will undercharge LiFePO4 (not a safety issue, but reduces usable capacity). A LiPo/Li-ion charge profile targets about 4.2V per cell; applied to a LiFePO4 cell (maximum ~3.65V) it overcharges the cell, which is a genuine safety concern. A quality LiFePO4 BMS should disconnect on overvoltage as a backstop, but you should not rely on that - verify your charge controller supports LiFePO4 mode before purchasing.

Networking and Range FAQ

How many hops can a message travel?

Meshtastic Hop Limits

In Meshtastic, every packet is born with a "hop limit" - a countdown that decrements each time the packet is relayed by a node. When the hop limit reaches zero, the packet is dropped and not forwarded further.

- **Default hop limit: 3** - The hop limit counts relay hops, so a default of 3 lets a message pass through up to 3 relay nodes between sender and receiver (sender → relay → relay → relay → receiver).
- **Maximum configurable hop limit: 7** - Meshtastic enforces a hard cap of 7 to prevent runaway flooding
- **Recommended range: 3-5** - 3 for most local networks; 4-5 for large geographic deployments; 7 only in special circumstances

Configure in app: **Radio Config → LoRa → Hop Limit** or via CLI: `meshtastic --set lora.hop_limit`

3

MeshCore Hop Behavior

MeshCore uses path-based routing, where hop limits work differently. A route discovery (path discovery packet) flood uses its own hop limit; once a route is established, data packets are forwarded along the specific path. The practical maximum network diameter depends on how many repeaters are in the discovered path, bounded by the path discovery packet hop limit setting (default sufficient for most deployments).

Why Not Set Hop Limit to Maximum?

Higher hop limits mean more retransmissions, more airtime consumption, and greater risk of broadcast storms (loops where packets bounce indefinitely). On a small, sparse network, a higher hop limit lets messages travel farther. On a dense network with many relays, hop limit 7 can cause severe congestion. Start with the default (3) and only increase if users at the network edges report messages not arriving.

Practical Distance Per Hop

Each hop adds roughly 2-30 km of distance, depending heavily on terrain and antenna height. The figures below are rough field estimates, not specifications - actual per-hop and end-to-end distances vary widely with siting, and the high end of each range requires near-ideal, line-of-sight conditions. They are also best-case in another sense: every added hop lowers the end-to-end delivery probability (compounding packet loss and airtime), so the longer multi-hop ranges are achievable but not routine or reliable. Treat them as illustrative ceilings, not planning numbers:

Hop Count	Typical Coverage (flat terrain with rooftop repeaters)
1 hop	2-8 km from sender to first repeater (approximate, antenna/preset dependent)
3 hops	10-40 km end-to-end (illustrative, environment-dependent)
5 hops	20-80 km end-to-end (best-case, assumes strong links each hop)
7 hops	40-150+ km end-to-end (exceptional best-case, requires mountain repeaters and ideal line-of-sight - not typical)

Why do I see duplicate messages?

Why Duplicates Happen

Duplicate messages in Meshtastic are normal and expected - they are a feature of flood routing, not a bug. When a node receives a message, it rebroadcasts it. If you're within radio range of multiple nodes that each received and retransmitted the same message, you may receive that message 2-4 times.

Meshtastic uses a packet ID deduplication window to suppress most duplicates at the firmware level - your phone typically doesn't show them all. When duplicates do appear in the app, it's usually because the deduplication window has been exceeded (the second copy arrived much later) or there's a routing issue.

Common Causes of Excessive Duplicates

- **Multiple nearby Router/Repeater nodes** - If 4 different routers can all hear the sender, all 4 will retransmit, and you receive 4 copies. This is expected flooding behavior.
- **Very high hop limit** - Higher hop limits mean more retransmissions; more retransmissions from more paths means more copies arrive.
- **Routing loop** - Rare but possible: Node A relays to B, B relays to C, C relays back to A. This can cause a packet to circulate until the hop counter reaches zero. Firmware prevents infinite loops via the hop counter, but you may see many copies during the loop's lifetime.
- **Firmware bug** - Older firmware versions had deduplication issues. Ensure all nodes are on current firmware.

What To Do

A small number of duplicates (1-2 extra copies of busy-traffic messages) is normal in a well-functioning mesh. If you see 5+ copies consistently:

- Check that no nodes are accidentally configured with very high hop limits (7)
- Verify all nodes are on current firmware (deduplication has been improved over time)
- Look for nodes sharing the same node ID. Node IDs are derived from a device's identity, so collisions are unusual, but they can occur if a node's NodeDB or keys were cloned onto a second device. Two devices presenting the same ID can confuse deduplication and routing.
- Reduce hop limit slightly if the network is small and dense

What is channel utilization and why does it matter?

What Channel Utilization Means

Channel utilization is the percentage of time the LoRa radio channel is occupied by transmissions. It's displayed in the [Meshtastic app](#) as a percentage (visible in the channel info or device telemetry).

Think of it like a single-lane road. If utilization is 10%, there's plenty of room for everyone. At 25%, it's getting busy. Around 40-50%, there's constant congestion and collisions. At 80%+, the channel is saturated and very few packets get through.

Why the 25% Threshold Matters

Meshtastic's flood routing uses random backoff timing to reduce collisions - nodes wait a random short period before retransmitting a packet. At low utilization, these collisions are rare. The ~25% figure is a community rule of thumb (not a hard physical constant): around 25% utilization, rising collision probability starts to degrade throughput noticeably, and once you reach roughly 40-50% the network becomes largely unusable.

What Drives High Channel Utilization

In order of typical impact:

1. **Position broadcast interval too short** - The default fixed position broadcast interval is 15 minutes, but users can lower it, and with smart broadcast enabled a moving node can transmit as often as every 30 seconds (the smart-broadcast minimum). On a 100-node

network with everyone broadcasting every 30 seconds, position traffic alone saturates the channel. Fix: set position broadcast to 5-30 minutes for fixed nodes, 1-5 minutes for mobile nodes. `meshtastic --set position.position_broadcast_secs 1800`

2. **Telemetry too frequent** - If many nodes broadcast device/environment telemetry at 5-minute intervals, the aggregate airtime adds up. Fix: set telemetry intervals to 15-30 minutes.
3. **Long-range preset with many nodes** - The Long Slow preset uses a high spreading factor, so each packet takes on the order of several seconds of airtime (the exact figure depends on payload size and bandwidth). With 50+ nodes, even moderate message rates saturate the channel. Fix: migrate to Medium Slow or Medium Fast, which cut airtime per packet substantially; note this trades some range for much lower airtime.
4. **High hop limits creating more retransmissions** - Total airtime for a packet scales with the number of nodes that actually rebroadcast it, so lowering the hop limit (for example from 5 to 3) reduces airtime by an amount that depends on how many relays would otherwise have re-transmitted, not a fixed percentage.
5. **Too many Router role nodes** - Every node set to ROUTER retransmits every packet it hears. In a dense area, CLIENT nodes should outnumber ROUTER nodes significantly. Fix: set personal handheld devices to CLIENT role, not ROUTER.

Target Values for a Healthy Network

Channel Utilization	Status	Action
0-10%	Healthy	None needed
10-25%	Moderate	Monitor; consider reducing position intervals
25-40%	Congested	Reduce intervals; consider faster preset
40%+	Saturated	Immediate action needed: reduce broadcasts, change preset

Firmware and Software FAQ

How do I update my Meshtastic firmware?

The Easy Way: Web Flasher

The Meshtastic web flasher at flasher.meshtastic.org handles everything automatically. It works in Chrome and Edge (Firefox does not support WebSerial).

1. Connect your device to your computer via USB
2. Open flasher.meshtastic.org in Chrome or Edge
3. Click "Connect" and select your device's serial port
4. The flasher auto-detects your board type and shows the latest stable release
5. Click "Flash" and wait 2-3 minutes for the process to complete
6. The device reboots automatically when done

Your configuration is preserved during a normal update. Settings stored in flash persist through firmware updates in most cases. However, feature releases (e.g., 2.2.x to 2.3.x) can occasionally reset settings - always export your config before any update.

Exporting Config Before Updating

```
meshtastic --export-config > my-node-config.json
```

This saves all your settings to a JSON file. If the update resets your config, restore with:

```
meshtastic --import-config my-node-config.json
```

Over-the-Air (OTA) Updates

ESP32-based boards (such as Heltec and most T-Beam variants) are updated over USB with the Web Flasher (or the CLI/esptool) - they are not flashed over Bluetooth. nRF52840 boards such as the RAK4631 and T-Echo do support Bluetooth OTA via the Nordic nRF DFU process (in addition to USB UF2 drag-and-drop flashing) - note that DFU OTA carries some risk of a failed update, so keep USB recovery available. A few nRF52840 boards (for example the Seeed XIAO nRF52840) are UF2/USB-only and cannot be updated over Bluetooth. When an in-app "Firmware Update" option is available for your board in the Android app, it is convenient for nodes that are accessible but not near a computer.

How Often Should I Update?

For personal devices: update when you want new features or bug fixes. There's no urgency unless a security issue is announced. For community infrastructure nodes: test new firmware on a non-critical node first, then coordinate with the community before updating. Avoid updating repeaters mid-event or during high-use periods.

When NOT to Update

- During an active emergency operation - never update infrastructure during an event
- If your network is working well and other nodes are on older firmware - major version jumps can cause compatibility issues until all nodes update
- Right before a planned use - allow 24 hours to discover and fix any update-related issues

What Meshtastic firmware version should I run?

Always Run Stable Releases on Infrastructure

Meshtastic releases three types of firmware builds:

Version numbers below are illustrative examples only. Firmware versions change frequently - always check the [official Meshtastic releases page](#) for the current stable and beta versions (examples shown reflect the 2.x release line as of 2026).

Build Type	Stability	Use for
Stable (e.g., 2.7.x as of 2026 - check for the current release)	High	All production nodes and community repeaters
Beta (current release line, beta channel - check for the current label)	Medium	Personal testing nodes only
Alpha / Nightly	Low	Developers only; may have breaking bugs

Version Compatibility

Meshtastic maintains backward compatibility within a major version (2.x). All 2.x nodes can communicate with each other, though newer firmware may use packet formats that older firmware ignores. Avoid running 1.x firmware anywhere on an active 2.x network - the 1.x and 2.x lines use different protocol and encryption defaults and are effectively incompatible on the same mesh.

Public Key Cryptography (PKC/PKI) Direct Messaging was introduced in firmware **2.5.0** and requires both nodes to run 2.5.0 or newer for public-key-encrypted DMs. Channel messages still work across 2.x versions regardless; only direct messages between nodes that are both on 2.5.0+ (and have

exchanged keys) receive PKI encryption. DMs to or from devices on 2.4.3 or older firmware are not protected by PKI.

Checking Your Current Version

In the [Meshtastic app](#): Settings → About, or check the device info screen. Via CLI:

```
meshtastic --info | grep Firmware
```

Upgrading a Community Network

For coordinated community upgrades across multiple repeaters:

1. Announce the planned update in your community communication channel
2. Update one non-critical repeater first and test for 48-72 hours
3. If no issues, update remaining repeaters in off-peak hours (late night)
4. Document firmware versions for each node in your network inventory

How do I factory reset my node?

When to Factory Reset

Factory reset clears all configuration and returns the node to out-of-box defaults. Do this when:

- You've changed so many settings that the node misbehaves and you can't identify the cause
- You're repurposing a node from one network to another and want a clean start
- Selling or giving away a node (removes your channel keys and personal info). A factory reset clears your channel keys and configuration, but verify there is no message history or personal data on an inserted SD card, and note that other nodes on the mesh may retain a record of your node ID until it ages out. Remove any SD card before transferring a device.
- As a last resort after troubleshooting fails

Via the App

In the Meshtastic Android/iOS app with the node connected: Settings → Device → Reset Node DB (resets node database only) or Factory Reset (clears all configuration). This is the recommended method for non-technical users.

Via CLI (advanced/optional)

The command below is advanced/optional - it requires the separate Meshtastic command-line tool installed on a computer and a USB connection to the node. If you are not comfortable with the command line, use the in-app Factory Reset described above instead.

```
meshtastic --factory-reset
```

This clears all radio configuration, channels, and user settings. The node reboots and appears with default settings. Note: this does NOT clear the firmware - you will still be on the same firmware version after a factory reset.

Via Hardware (Button)

Most boards support a hardware factory reset by holding the user button during power-on, or pressing it a specific number of times. Check the Meshtastic documentation for your specific board model. The hardware method is useful when you can't connect to the device via Bluetooth or USB.

What Factory Reset Does NOT Do

- Does not change the firmware version
- Does not change the hardware node ID (derived from the hardware MAC address, not erasable). Note, however, that a factory reset does regenerate the node's encryption keys, which can break direct messaging with nodes that cached your old key until they re-exchange keys with you.
- Does not affect other nodes on the network - their node databases will still have a record of your node until it ages out

After a Factory Reset

You'll need to reconfigure everything: region, channel, role, position (if fixed), and any module settings. If you exported your config before resetting, restore it with:

```
meshtastic --import-config my-node-config.json
```

If you're joining a community network, obtain the channel QR code or URL from the network coordinator and scan/tap it to configure your channels automatically.

Security and Privacy FAQ

Is Meshtastic encrypted?

Can anyone read my messages?

Short Answer

Meshtastic messages are encrypted, but the level of protection depends on which channel you're using. The default channel (LongFast) uses a known public key and provides essentially no privacy. Custom channels with randomly generated keys provide strong message confidentiality.

The Default Channel: Not Private

The default Meshtastic channel uses a Pre-Shared Key (PSK) of `AQ==`, which is base64 for a single byte of value `0x01` - a publicly known and documented default key. Any Meshtastic user in radio range can read messages on the default channel, including the [Meshtastic app](#) developers and anyone who has read the public documentation.

The default channel is suitable for public community communication where privacy isn't a concern. Do not send anything private on the default channel.

Custom Channels: Strong Privacy

When you create a channel with a randomly generated PSK (using the app's random key generator), the app can produce a 256-bit AES key that cannot be recovered by brute force with current technology. Note that Meshtastic uses AES-128 or AES-256 in CTR mode depending on the key length (16 vs 32 bytes) - the default and many shared channels are AES-128, while the app's random-key generator can produce a 256-bit key. Messages on this channel are readable only by nodes that have the same PSK.

Encryption security: AES-256 is a strong, government-approved cipher and the cryptography itself is sound. However, security depends on key management (how you distribute the PSK to your community), and Meshtastic exposes metadata and relies on manual key distribution - so do not treat a custom channel as government-grade secure for highly sensitive traffic.

Direct Messages: Even Better (Firmware 2.5+)

Direct messages use X25519 (Curve25519) ECDH public-key (PKI) encryption. PKI direct messaging was introduced in firmware 2.3 and became the robust default in 2.5; run firmware 2.5 or later on all nodes for current PKI DM security. It provides:

- End-to-end encryption between just sender and recipient
- No shared secret to distribute - keys are derived automatically from each node's public/private key pair

Note: Meshtastic does **not** provide forward secrecy. Because the keys are long-lived, traffic captured today can be decrypted later if a key is compromised - the official documentation notes that Meshtastic is vulnerable to "harvest now, decrypt later" attacks. Do not assume that seizing a key in the future cannot expose past messages.

What an Eavesdropper Can See

Even with properly configured channel encryption, a radio observer can see:

- That LoRa transmissions are occurring on the frequency
- The approximate timing and frequency of transmissions
- Some packet header fields that are not encrypted (node IDs, a channel hash, hop count, and routing/metadata)

Message content, sender names, and channel names are inside the encrypted portion of the packet, so they are hidden if a custom PSK is in use. However, the unencrypted header still exposes node IDs, a channel-hash byte, and routing/metadata - so an eavesdropper can still learn who is transmitting and when.

Practical Recommendations

- For community chat where privacy isn't critical: default channel is fine

- For any sensitive coordination: create a custom channel with a random PSK
- For private one-on-one messages: use DMs with firmware 2.5+ on both ends
- For highly sensitive communications: LoRa mesh is supplemental - use Signal or other end-to-end encrypted messaging for truly sensitive content

Who can see my location on the mesh?

How Position Data Spreads

When position reporting is enabled on a Meshtastic node, GPS coordinates are broadcast as position packets on the mesh. These packets travel through the network like any other message and are visible to all nodes that receive them.

Who Can See Your Position

- **Anyone on your channel** - If you're on the default channel, anyone with Meshtastic within a few hops can see your position in their node list and on their map
- **meshmap.net** - If you have MQTT uplink enabled to the public Meshtastic broker, your position appears on the global map
- **MQTT operators** - Anyone subscribed to the same MQTT topic receives your position packets

Controlling Position Privacy

Option 1: Disable Position Reporting

Turn off GPS or disable position broadcasts entirely:

```
meshtastic --set position.gps_mode DISABLED  
meshtastic --set position.position_broadcast_secs 0
```

Your node will still relay messages and appear in node lists (by node ID), but without location data. Note that disabling position only hides your GPS coordinates: the node still transmits its node ID and (by default) its name, and the transmitter itself can be located by radio direction-finding. For

OPSEC-sensitive field or shelter operations, also anonymize the node name, minimize transmissions, and remember that the mere presence of a radio signal is detectable.

Option 2: Use Position Precision Reduction

Broadcast a less precise location rather than exact GPS coordinates:

```
meshtastic --set position.position_precision 10
```

The `position_precision` value sets how many bits of latitude/longitude are shared, which maps to a much coarser ground radius than the raw number suggests. Approximate values (per the Meshtastic `precision_bits` reference): 32 = full precision; 19 $\approx \pm 46$ m; 16 $\approx \pm 365$ m; 13 $\approx \pm 2.9$ km; 11 $\approx \pm 11.7$ km; 10 $\approx \pm 23$ km; 0 = position never sent. Note these are coarser than they appear - for example, precision 16 still exposes a roughly 365 m radius, not 1 km, so choose conservatively. With precision 10 (± 23 km) you appear somewhere in a wide area around your city rather than at your exact street address.

Option 3: Private Channel with No MQTT

Create a private channel and disable MQTT uplink. A private channel with MQTT disabled keeps your position content out of the encrypted payload that reaches the public maps, so your coordinates don't appear there. It is not total invisibility, though: position packets still travel over RF, and unencrypted header fields (node IDs, hop data) remain observable to any radio listener. Privacy also depends on every node on the channel keeping MQTT/uplink disabled - if any participant later enables MQTT or bridges to the internet, position can leak.

Fixed Position vs GPS

For fixed infrastructure nodes (repeaters), configure a static position manually rather than using GPS. This gives you explicit control over what coordinates you publish, and you can choose a slightly offset position if you prefer not to reveal your exact rooftop address.

Can I trust MeshCore encryption for sensitive communications?

MeshCore Encryption Summary

MeshCore provides two layers of encryption:

- **Channel encryption** - AES-128 ECB + HMAC-SHA256 with a key derived from the channel configuration. All nodes on the channel share this key. Note that public (no-PSK) channels and open hashtag rooms are readable by any passive listener.
- **Direct message encryption** - ECDH key exchange (Curve25519 elliptic curve) with AES session keys. Only sender and recipient can read DMs.

Cryptographic Strength

Both AES-128 and Curve25519 ECDH are modern, vetted cryptographic primitives used in TLS 1.3, Signal Protocol, and other high-security applications. Note, however, that those protocols use authenticated (AEAD) modes, whereas MeshCore's mode of use - AES-128 in ECB mode with a short 2-byte authentication tag and static ECDH keys - is weaker than the AEAD constructions in TLS 1.3 and Signal. MeshCore direct messages use sound primitives (X25519 ECDH key agreement + AES-128) but in a much simpler construction than Signal. Specifically, it uses AES-128 in ECB mode, a short 2-byte authentication tag, and static (long-term) keys - so it does NOT provide forward secrecy or the ratcheting that Signal uses. Treat it as basic confidentiality against casual interception, not as equivalent to Signal for high-sensitivity traffic.

What Encryption Protects Against

- **Passive eavesdroppers with LoRa hardware** - Cannot read message content with correct key management. (Public, no-PSK channels and open hashtag rooms remain readable by passive listeners.)
- **Casual interception of DM content** - Without the key, an eavesdropper cannot read DM plaintext. Note, however, that because DMs use AES-128 in ECB mode, identical plaintext blocks produce identical ciphertext, so an observer recording traffic long-term may detect repeated or structured content patterns (for example, templated emergency-comms messages) even without the key. Avoid sending fixed-format sensitive messages.
- **Replay/recorded-traffic decryption** - MeshCore's DM ECDH uses static long-term identity keys, not ephemeral session keys, so it does NOT provide forward secrecy. If a device's private key is later recovered, previously recorded DMs to or from that device can be decrypted. Replay resistance, where present, comes from per-message timestamps, not from the key exchange.

What Encryption Does NOT Protect Against

- **Physical access to your device** - Private keys are stored on the device. Seizure of the device potentially allows recovery of stored messages.
- **Compromised network operator** - A room server operator can see metadata (who is communicating with whom, when) and the content of room/channel posts that pass through the server; only person-to-person DMs are end-to-end encrypted and not readable by the operator.
- **Traffic analysis** - Observers can see that radio transmissions are occurring, timing patterns, and frequency of communication even if they cannot read content
- **Social engineering** - If a recipient shares a DM, encryption provides no protection
- **Channel key compromise** - If the shared channel key is obtained by an adversary, all past channel traffic (if recorded) can be decrypted
- **Loss of forward secrecy** - Because DM keys are static, recovering a private key compromises past as well as future messages; there is no per-message ratcheting like Signal's Double Ratchet.

Appropriate Use Cases

MeshCore's encryption is appropriate for:

- Community coordination that you'd prefer not to broadcast publicly
- Emergency operations where commercial networks are unavailable - but only as a supplemental, best-effort text channel. Mesh has no delivery guarantee, low throughput, and degrades under congestion; maintain a primary emergency-comms method and

never rely on mesh alone for life-safety traffic. The encryption above provides basic confidentiality only and is not suitable for life-safety or classified-equivalent secrecy.

- Private group communications within a trusted community

MeshCore encryption is *not* appropriate as the sole protection for:

- Communications subject to legal privilege (attorney-client, medical)
- Operational security against nation-state adversaries
- Highly sensitive personal information

For these use cases, use end-to-end encrypted applications (Signal, ProtonMail) with LoRa mesh serving only as a transport layer to reach an internet gateway.

Troubleshooting Advanced Issues

My node is flooding the channel with repeated messages

My node is flooding the channel with repeated messages

If you notice unusually high channel utilization, see the same message appear many times in quick succession, or other mesh users report your node is hammering the airwaves, your node may be contributing to a flooding condition. Here is how to understand it, diagnose it, and fix it. (Note: the role and menu details below are Meshtastic-specific. If you run a MeshCore repeater that is over-flooding, MeshCore repeaters use selective/path-based forwarding rather than managed flooding - reduce duty cycle and how often the node broadcasts its position/adverts to cut congestion.)

Is flooding always a problem?

Some re-broadcasting is normal and intentional in a mesh network. When a node receives a packet it has not seen before and its hop limit allows, it re-broadcasts so the message can reach nodes that did not hear the original. Meshtastic uses a *managed flood routing* approach where every ROUTER node re-transmits. What becomes a problem is when the same packet circulates far more times than needed, driving channel utilization above the recommended 25% level. Meshtastic treats roughly 25% as the green/optimal soft ceiling for channel utilization: above it the firmware's contention window scales up and TX is deferred, collisions increase, and overall throughput degrades for every node on the channel.

Common causes

- **Two ROUTER nodes in direct line-of-sight of each other:** When both nodes hear each other perfectly and both re-broadcast the same packet, they can trigger a

forwarding loop if the hop count has not decremented to zero. Each re-broadcast is heard by the other, which may queue another retransmit depending on firmware version.

- **Bridge nodes on the same channel:** Nodes configured as bridges can inadvertently re-introduce packets to a segment that already received them if the bridge logic is not configured correctly, creating a loop.
- **Hop limit set too high:** The default hop_limit is 3, meaning a packet can traverse up to 3 intermediate nodes before being dropped (the maximum is 7). If it has been set to 7 or higher, packets live long enough to circle back through the network.
- **Multiple nodes with the same node key or ID:** Duplicate node identities confuse deduplication logic, causing packets that should be discarded to be treated as new.

How to diagnose

1. Open the [Meshtastic app](#) and check your node's channel utilization in its device metrics. If utilization is above 25% and rising when no one is actively sending, something is re-broadcasting excessively.
2. Check the node list for any duplicate node names or IDs that might indicate a misconfigured clone.
3. Temporarily power down nodes one at a time and watch whether utilization drops - this isolates the culprit node.

Fixes

- **Reduce hop_limit to 3** (Radio Config → LoRa → Hop Limit). For most deployments, 3 hops is sufficient to cross a well-designed mesh.
- **Change ROUTER nodes to CLIENT role** for nodes that do not need to relay traffic. CLIENT nodes do not re-broadcast, which dramatically reduces airtime.
- **Review bridge node configuration:** Ensure bridge nodes have channel isolation configured correctly so they do not echo packets back onto the originating channel.
- **Enforce firmware-level deduplication:** Keep all nodes on current stable firmware. Older firmware versions had less aggressive deduplication windows.
- **Stagger node placement:** If two ROUTER nodes are in perfect line-of-sight, consider whether both need to be ROUTERS, or whether one can be demoted to CLIENT.

Keep in mind that each relaying node adds another full transmission of the packet, so total airtime scales with the number of relays (roughly one transmission per relay plus the original), not just with hop count.

My Meshtastic node and MeshCore node cannot communicate

My Meshtastic node and MeshCore node cannot communicate

This is one of the most common points of confusion for new mesh users: both Meshtastic and MeshCore run on similar LoRa hardware, operate in the same 915 MHz band, and can even share the same physical channel frequency - yet they cannot exchange messages with each other. Here is why, and what your options are.

Why they are incompatible

Meshtastic and MeshCore are completely independent software stacks. While both modulate data over LoRa radio, they differ in every layer above the physical radio signal:

- **Packet format:** Meshtastic uses Protocol Buffers (protobuf) serialized payloads inside its own envelope structure. MeshCore uses a different binary packet format with its own header fields, addressing, and payload encoding. A MeshCore node receiving a Meshtastic packet sees garbage bytes it cannot interpret, and vice versa.
- **Encryption:** Meshtastic encrypts channel traffic with AES (AES-128 or AES-256 depending on the pre-shared channel key length; the default "LongFast" channel uses a short, publicly known key that provides no real privacy). MeshCore uses a different scheme - AES-128 for channel traffic with Curve25519 public keys for direct messages. The two use incompatible key material and algorithms, so even if a node could parse the outer envelope, it cannot decrypt the other's payload.
- **Routing logic:** Meshtastic uses managed flood routing with hop decrement. MeshCore uses a different routing model oriented around repeater nodes and room servers. The routing metadata fields in each protocol are incompatible.

- **Node identity:** Meshtastic identifies nodes by a 32-bit node number derived from hardware MAC. MeshCore uses a public-key-based identity system. There is no translation layer between them.

There is no bridge - yet

As of the current date, no production bridge firmware or software exists that can transparently relay messages between a Meshtastic mesh and a MeshCore mesh in real time. Research and community projects have explored the concept, but none have reached a state suitable for general use.

Workarounds for cross-protocol communication

If your community includes both Meshtastic and MeshCore nodes and you need them to share information, the following approaches can help:

- **Room server as intermediary:** A MeshCore room server is normally a mesh contact reached over RF (or Bluetooth) by selecting it from your contact list - it is not an internet service by default. Some operators bridge a room server to the internet as a separate, non-default and uncommon setup; this is an advanced configuration, not something that works out of the box. Even where such a bridge exists, it only connects MeshCore clients to that room server - it does not let a Meshtastic node join.
- **Manual human or gateway relay:** The only reliable cross-protocol path today is a human (or a custom script) reading messages off one network and re-posting them on the other. Note that **Winlink is not a bridge between Meshtastic and MeshCore:** Winlink is an amateur-radio email system that requires a valid amateur radio license and separate HF/VHF gear or an internet connection, and neither Meshtastic nor MeshCore connects to it natively. Do not plan an interop path around Winlink.
- **Dual hardware stacks:** Deploy two physically separate nodes at the same location - one running Meshtastic and one running MeshCore. A human operator or automated script on a co-located computer can relay messages between the two networks over their respective serial or Bluetooth APIs.

Recommendation

For a new community deployment, choose one protocol and standardize. Mixed-protocol communities incur significant coordination overhead. If you are integrating with an existing regional mesh, match whatever protocol that mesh uses.

RF interference is affecting my node — how do I diagnose it

RF Interference Is Affecting My Node - How to Diagnose It

LoRa spread-spectrum modulation gives it excellent resistance to narrowband interference, but it is not immune. If your node is experiencing unexplained packet loss, poor RSSI from nearby nodes, or erratic behavior that does not correlate with distance or obstacles, RF interference may be the culprit.

Symptoms of interference

- **High SNR variation:** For instance, a link that normally shows around -5 dB SNR may suddenly drop to -15 dB or worse for no physical reason. (Baseline SNR varies from link to link; the diagnostic signal is the sudden, unexplained change, not the absolute number.) If SNR fluctuates wildly over minutes without any change in node position, an external RF source is likely corrupting packets.
- **Unexpectedly poor RSSI from nearby nodes:** RSSI substantially worse than your link budget predicts can indicate that wideband noise is raising the noise floor. Link-budget estimates carry large uncertainty, so treat a clear, persistent gap between predicted and observed RSSI - not a precise dB figure - as the warning sign.
- **Time-of-day failure patterns:** If failures cluster during business hours, when appliances run, or at regular intervals (e.g., every 30 minutes), a duty-cycling device such as a smart meter network is a likely source.
- **One direction fails more than others:** If nodes in one geographic direction consistently perform worse, a directional interference source may be oriented toward your antenna.

Diagnosis tools

The most effective way to see what is happening in the 902 - 928 MHz band is with a Software Defined Radio (SDR):

- **Hardware:** An RTL-SDR dongle (~\$25 USD) with a 915 MHz antenna is sufficient. Higher-end options like HackRF or Airspy offer better sensitivity.
- **Software:** SDR# (Windows) or GQRX (Linux/macOS) - both are free. Open the spectrum analyzer view and zoom to 902 - 928 MHz.
- **What to look for:** Constant carriers (a vertical spike that never moves) indicate a narrowband source. Sweeping signals suggest frequency-hopping devices such as cordless phones. Periodic bursts at regular intervals suggest smart meter networks (ITRON, Sensus, etc.) which use this band extensively in North America.

Common interference sources in the 902 - 928 MHz band

- Baby monitors (some older analog models use 900 MHz, though most modern units operate at 2.4 GHz or on DECT at ~1.9 GHz)
- 900 MHz analog/digital cordless phones (older models - note these are not DECT, which is a separate ~1.9 GHz standard)
- Smart meter networks (AMI/AMR systems from utilities)
- Industrial wireless sensors and SCADA equipment
- Some older Wi-Fi extenders and video senders

Mitigation strategies

- **Change frequency slot (channel) in Meshtastic:** Meshtastic lets you select a different frequency slot within the ISM band while keeping the same modem preset. If a specific frequency is congested, moving to a different frequency slot shifts your center frequency away from the interference - this is the correct way to dodge a narrowband interferer. **Important:** the modem preset (LongFast vs. LongSlow vs. MedFast) sets the spreading factor and bandwidth (speed vs. range), *not* the center frequency. Changing the preset does not move you off an interfering frequency, and every other node in your mesh must use the identical region and preset to communicate - if you change your preset you will lose contact with all nodes still on the old preset. Change the frequency slot, not the preset, and keep your mesh on a common preset.
- **Use a directional antenna:** A Yagi or patch antenna pointed toward your intended mesh nodes, and away from the interference source, provides front-to-back rejection typically on the order of 10 - 20 dB, depending on the antenna design (a small patch may give less; a larger multi-element Yagi can give more).

- **Relocate the antenna:** Moving the antenna even a few meters can place a building or terrain feature between your node and the interference source, providing significant shielding.
- **Reduce antenna height:** Counter-intuitively, lowering an antenna can reduce pickup of distant interference while maintaining adequate coverage of nearby nodes.

MeshCore-Specific FAQ

What is the difference between a Repeater and Room Client in MeshCore?

What Is the Difference Between a Repeater and Room Client in MeshCore?

MeshCore ships several firmware variants. This page compares the **Repeater** firmware with the **Companion (room) client** firmware - the variant a user-carried node runs to connect to a room server. (Note: across the rest of this wiki this client is referred to as the **Companion** variant; "Room Client" here means the same Companion firmware used to interact with a room server. The room server itself - the node that actually stores and serves message history - runs separate Room Server firmware and is not covered in detail here.) Choosing the wrong one is a common source of confusion. This page explains both in detail.

REPEATER_FIRMWARE

A node flashed with REPEATER_FIRMWARE acts as pure RF infrastructure. Its job is to forward MeshCore packets toward their destination to extend the reach of the mesh. Unlike flood-based mesh systems, a MeshCore repeater forwards **selectively** (route/path-based) - it does not rebroadcast every packet it receives. Key characteristics:

- **No screen required:** Repeater firmware is designed to run on minimal hardware - a bare ESP32 or nRF52 board with a LoRa module. It has no UI and does not need a display.
- **Minimal power consumption:** Because it does not maintain Bluetooth, Wi-Fi, or app connections, a repeater node consumes significantly less power than a room client. This makes it ideal for solar-powered or battery-backed infrastructure deployments.

- **On-mesh identity:** A repeater has its own on-mesh identity (a keypair and a name) so it can be discovered and administered, but it does not originate or receive user messages; it only forwards packets at the radio layer.
- **No room server interaction:** Repeater firmware does not connect to a room server. It has no concept of message storage or retrieval.

Best for: Hilltop relays, tower-mounted infrastructure nodes, solar repeaters in remote locations - any deployment where the goal is coverage extension rather than message origination.

ROOM_CLIENT_FIRMWARE

A node running the Companion ("room client") firmware has a full client identity and can connect to a MeshCore **room server** to retrieve stored messages. Note that the message storage and retrieval capability lives in the *room server* firmware; the room client is simply a Companion node that logs in to that server to fetch history it missed. Key characteristics:

- **Client identity:** The node generates a public/private key pair on first boot. This keypair is its identity on the mesh and with room servers. Messages can be addressed to it specifically.
- **Room server connectivity:** A room client connects to a room server primarily over RF through the mesh (or via BLE to a paired room server) to send messages and download stored messages it missed while offline. The room server, not the client, holds the stored message history.
- **Higher power use:** Maintaining a client identity and processing addressed messages consumes more CPU cycles and radio time than a pure repeater.
- **Does not relay:** A Companion/room client does *not* rebroadcast or relay mesh traffic the way a repeater does. Only Repeater firmware forwards packets for other nodes. If you need range extension, deploy a repeater node.

Best for: User-carried nodes, [base station nodes](#) that interact with a room server, nodes that need to send and receive addressed messages.

Can a node be both?

No. You must choose one firmware at flash time. A single physical node cannot simultaneously run the Repeater firmware and the Companion (room client) firmware. If you need both roles at one location - for example, a hilltop site that also needs a room-server-connected client - you need two separate physical nodes: a repeater for range extension and a separate client node.

Summary comparison

Feature	REPEATER_FIRMWARE	ROOM_CLIENT_FIRMWARE
---------	-------------------	----------------------

Relays/forwards packets for others	Yes (selectively, route-based)	No
Has on-mesh identity	Yes (infrastructure)	Yes (user)
Connects to room server	No	Yes
Sends/receives user messages	No	Yes
Power use	Low	Higher
Screen needed	No	Optional

How do I connect to a MeshCore room server from the app?

How Do I Connect to a MeshCore Room Server From the App?

A MeshCore room server stores messages for offline nodes and enables larger-group conversations that persist beyond the RF range of any single transmission. Importantly, a room server is reached **over the LoRa mesh, not over the internet**. There is no server IP address, hostname, or TCP port involved, and no firewall configuration is needed. To join one you need to be within RF range of the mesh (directly or via relays) and know the room server's password, which is set by the server operator.

Step-by-step connection

1. Open the **MeshCore app** on your phone and ensure your companion node is connected via Bluetooth (or USB serial).
2. Wait for your companion node to **discover the room server as a contact on the mesh**. Room servers advertise themselves over LoRa, so they appear in your contact list once your node hears them (directly or relayed through other nodes).
3. Select the **room server** from your contact list.
4. When prompted, enter the **room server password**. This is the shared secret set by the server operator. It must match exactly, including capitalization.
5. Once the password is accepted, you join the room and can send and receive messages. Messages are stored by the room server and delivered to members as they come into range.

Administration of the room server itself is done locally over Bluetooth or USB serial on the room server device, not over a network connection.

Troubleshooting connection failures

If you cannot join the room server, work through these checks:

Room server does not appear as a contact

- The room server node may not be powered on or may be out of RF range. Verify the room server has power and that your node can hear it on the mesh (directly or through relays).
- Move closer to the room server or to a node that relays it, then wait for your node to rediscover it on the mesh.
- Confirm your companion node is on the same region/frequency settings as the room server so they can hear each other on LoRa.

Password rejected or wrong password error

- The password you entered does not match the room server's configured password. Passwords are case-sensitive. Try re-entering it manually rather than copy-pasting to rule out invisible characters.
- The operator may have changed the password. Ask the server operator for the current password.

Messages are not arriving

- You may be out of RF range of the room server and any relaying nodes. Because delivery happens over the mesh, you only receive stored messages when your node is within reach of the server (directly or relayed).
- Confirm your companion node is still connected to your phone over Bluetooth or USB and is participating in the mesh.

Can I run MeshCore and Meshtastic simultaneously on the same hardware?

Can I Run MeshCore and Meshtastic Simultaneously on the Same Hardware?

The short answer is: **no**. You can only run one firmware at a time on a given LoRa node. However, there are practical workarounds if you need coverage of both protocols at one location.

Why only one at a time

Both MeshCore and Meshtastic are compiled firmware images flashed directly to the microcontroller (ESP32, nRF52840, RP2040, etc.) on your LoRa board. When you flash a firmware image, it replaces whatever was there before. There is no multi-boot capability, no virtual machine layer, and no way to timeshare the LoRa radio hardware between two independent firmware stacks. The LoRa transceiver (SX1276, SX1262, LR1121, etc.) is a single physical peripheral that can only be driven by one firmware at a time.

Additionally, even if you could somehow run both, they would need to use the same LoRa radio simultaneously - which is physically impossible without two separate radio modules. Each transmission requires exclusive use of the transceiver.

Switching between firmwares

You can re-flash a node from MeshCore to Meshtastic (or vice versa) at any time using the appropriate web flasher or CLI tool. The process takes a few minutes and requires a USB connection. However, you lose all configuration from the previous firmware when you do so, making it impractical to switch frequently.

Running both protocols at one location

If your community or deployment site genuinely needs to participate in both a MeshCore mesh and a Meshtastic mesh, the practical solution is two separate physical nodes:

- **Node A:** Running MeshCore (REPEATER_FIRMWARE or ROOM_CLIENT_FIRMWARE as appropriate)
- **Node B:** Running Meshtastic firmware

Both nodes can be co-located at the same site and connected to the same power supply, but each should have its **own separate antenna**. Do **not** share one antenna between two transmitting nodes with a passive splitter: a splitter provides no port-to-port isolation, so each radio couples transmit power directly into the other's receiver front-end (risking damage to the LNA/PA), and it also adds roughly 3+ dB of insertion loss in each direction. Co-located 915 MHz transmitters instead need adequate physical antenna separation (ideally several feet of vertical or horizontal spacing) to limit desense and protect the receivers; sharing a single antenna between two transmitters requires a proper RF combiner or cavity duplexer, not a splitter. Many community infrastructure operators run exactly this dual-node, separate-antenna configuration at hilltop repeater sites to serve both ecosystems.

The future: software-defined gateways

Community developers have discussed building a software gateway that runs on a host computer (Raspberry Pi, etc.) and uses two LoRa radio modules - one for each protocol - to bridge messages between the two networks at the application layer. As of today, no such gateway exists in a production-ready state. Any such project would also need to handle the fundamental differences in addressing, encryption, and routing described in the cross-protocol FAQ page.

Recommendation

For most communities, the right answer is to **choose one protocol and standardize**. Mixed-protocol communities face ongoing friction in coordination, troubleshooting, and user experience. If your regional mesh has already standardized on one platform, matching that choice eliminates the need for dual-protocol coverage entirely.

If you are starting a new community from scratch and expect to attract users from both ecosystems, deploying two nodes at each infrastructure site is a legitimate and manageable

approach - the hardware cost of an extra node (~\$30 - 60 USD) is usually worth the dual-network coverage it provides (note: the two networks still cannot exchange messages with each other - each node simply lets you participate in its own network).

Community and Social FAQ

How do I find other mesh users in my area?

Finding local mesh users is one of the most common early questions - and one of the most important, since a single node is useful but a community is transformative.

Start With Online Resources

- **Community maps** - For Meshtastic, use the official map linked from meshtastic.org (community maps such as meshmap.net also exist); for MeshCore, use map.meshcore.io. Zoom to your area and see nodes that have reported positions. Note: these maps show only opt-in nodes that have position reporting enabled, so an empty or sparse map does not mean there are no users near you - many nodes disable position reporting for privacy.
- **r/meshtastic on Reddit** - Active community with a strong culture of welcoming new users. Post "[State/City] - Anyone here?" and you'll typically get responses within hours.
- **Official Meshtastic Discord** - discord.gg/ktMAKGBnBs - Has regional channels organized by continent, then country, then US state. The most active community hub.
- **MeshCore Discord** - Separate community from Meshtastic; search for regional channels or post in #general asking about your area.
- **r/meshcore** - Smaller but growing subreddit for MeshCore users.

Local Amateur Radio Clubs

Ham radio clubs are increasingly interested in LoRa mesh. Find clubs at:

- **arrrl.org/find-a-club** - ARRL club finder by zip code
- **QRZ.com club section** - Many clubs post there

Contact the club's net manager or digital committee. Even if they don't have mesh nodes yet, they may be interested in starting - and ham clubs are excellent vehicles for building community mesh networks.

Actually Going on the Mesh

The most reliable way to find local users is to simply get on the air with the default LongFast preset and listen. If there are active users in your area, you'll see their nodes in your node list within minutes. Send a message on the primary channel introducing yourself. Keep in mind the default channel is effectively public (it uses a well-known shared key): use it for discovery and general chat, but move any sensitive or incident-coordination traffic to a private channel.

If you hear nothing after 30 minutes in an area where you'd expect users, try:

- Moving to a higher elevation (rooftop, parking structure, hilltop)
- Checking a community map for nearby nodes and messaging them on Discord to arrange an on-air test

Note: don't switch to a slower preset (such as LONG_SLOW or VERY_LONG_SLOW) just to "find" users. All radios on a mesh must share the same modem preset to hear each other, so changing your preset away from the local default makes you invisible to everyone still on the default - the opposite of what you want when hunting for unknown local users. Only change presets if the whole local mesh has agreed to use the same one.

How do I start a mesh network where there are none?

Starting from zero is actually common - most community networks were started by one person who got tired of being alone on the mesh and decided to fix that. Here's how to bootstrap effectively.

This advice applies to both Meshtastic and MeshCore networks. Pick one protocol for your local mesh first, since the two do not interoperate (see the protocol comparison page) - everyone you recruit should run the same one.

The Minimal Viable Network

You need at least 2 nodes to have a network. Your first goal: find one other person willing to put up a node. Just one. A two-node network proves the concept and gives you something to demo to the next recruit.

Finding your second node:

- A family member or close friend in a different part of town
- A neighbor on higher ground than you
- Someone from the local ham radio club who agreed to try it out
- A coworker or friend interested in emergency preparedness

Once you have two nodes that can communicate, you have a live demo you can show to anyone.

Growing Past Two Nodes: 5 Tactics That Work

Common tactics that community organizers report working:

1. **Build a demo kit** - A pre-configured node on a tripod with an external antenna that you can bring to any meetup and have running in 5 minutes.
2. **Present at a local ham club meeting** - Request 10 minutes on the agenda. Demo is everything: bring a second node on the far end of the room.
3. **Post in local neighborhood apps** - You can frame it around preparedness, but be honest about what the mesh is: a supplemental, experimental comms tool, not a guaranteed emergency system. LoRa mesh is best-effort and unmonitored, so don't market it as a reliable emergency service - overselling its reliability can lead neighbors to depend on it when it may not deliver. For example: "I'm setting up a local mesh network for experimenting with off-grid text messaging. Anyone interested in participating?"
4. **Attend CERT training** - CERT graduates are motivated, community-oriented, and already thinking about emergency communications.
5. **Offer a "hardware night"** - Host a 2-hour session where you help 2-3 people set up their first node. Hands-on beats slides every time.

What Not to Do

- **Don't wait for critical mass** - Deploy your first node now. The mesh doesn't need to be large to be useful; it needs to exist to grow.
- **Don't over-engineer the first version** - An entry-level Heltec WiFi LoRa 32 V3 (around \$20-30) with its stock antenna works. Optimize hardware after you have community, not before.
- **Don't gatekeep** - The easier it is to join, the faster you grow. Simplify your onboarding documentation and be patient with beginners.

Can I use Meshtastic or MeshCore for commercial purposes?

The short answer: yes, with some limitations. Both platforms are open source with licenses that permit commercial use, and the ISM band spectrum they operate on allows commercial activity. Here's what you need to know.

Software License Considerations

Meshtastic firmware is licensed under the GNU General Public License v3 (GPLv3). Commercial use is permitted, but:

- If you modify the firmware and distribute it (to customers, as a product), you must release your modifications as GPLv3 as well
- You cannot make the firmware proprietary or closed-source while distributing it
- Using Meshtastic firmware as-is without modification carries minimal GPL obligations

MeshCore firmware is licensed under the permissive **MIT License**, which freely allows commercial use and proprietary derivatives - so it is *less* restrictive than Meshtastic's GPLv3, not more. (Some optional premium client features are sold separately.) As always, verify the current license in the MeshCore GitHub repository before deployment.

FCC Part 15 and Commercial Use

FCC Part 15 (the ISM band rules) explicitly permits commercial use. There is no requirement to be a non-profit or individual user. Commercial enterprises can operate LoRa mesh networks on 902-928 MHz under Part 15 rules, subject to the same power limits and emission standards as individual users.

Important caveat: Part 15 operation is unlicensed and unprotected (47 CFR 15.5). Your device must accept any harmful interference it receives from licensed services, and it must stop transmitting if it causes harmful interference to them. A commercial operation that needs guaranteed, interference-protected communications should not rely on a Part 15 mesh as its sole channel.

Common Commercial Applications

- **Event production companies** - Staff communication at festivals, sporting events, productions
- **Construction/site management** - Job site coordination in areas without reliable cell service
- **Agriculture/ranching** - Remote monitoring and worker communication
- **Mining and oil/gas** - Remote operations in areas without infrastructure
- **Telecommunications consulting** - Network design and deployment services

What You Cannot Do

- Exceed FCC Part 15 power limits: 1 W (30 dBm) maximum conducted output power. With the standard 6 dBi antenna allowance this yields roughly 4 W EIRP; higher-gain directional antennas require conducted power to be reduced dB-for-dB per 47 CFR 15.247(b)(4) and (c), so you cannot simply add a high-gain antenna to a 1 W radio and stay legal
- Operate outside the 902-928 MHz band in the US (or the equivalent ISM band in your country)
- Cause harmful interference to licensed services and fail to accept interference from licensed services
- Redistribute GPLv3-licensed firmware modifications without releasing source code

Consult with an RF engineer and attorney familiar with FCC regulations before major commercial deployments.

Antenna and RF FAQ

Do I need an external antenna?

The stock antenna that comes with most LoRa boards is a rubber duck (flexible whip) antenna, typically 1-3 dBi gain (often a quarter-wave stubby around 2 dBi). For many use cases, this is adequate - but upgrading to an external antenna is one of the most cost-effective improvements you can make.

When the Stock Antenna is Fine

- Indoor portable use (office, home) within 200-500m of your nearest mesh node
- Temporary deployments where you're moving frequently
- Testing and development before a permanent installation
- Dense urban areas with many nearby nodes (short hop distances)

When You Should Upgrade

- **Fixed outdoor installation** - Any permanent outdoor node should use an external antenna rated for outdoor use. Stock rubber ducks are not weatherproof.
- **Coverage issues** - If you can't reach nodes you'd expect to reach, a better antenna is the first thing to try.
- **Backbone repeater** - Repeaters covering a neighborhood or city need the best possible antenna. A 5-8 dBi fiberglass omni provides several dB more gain than a stock whip, which can substantially extend range (roughly doubling it under line-of-sight conditions; the gain in cluttered terrain is smaller).
- **Point-to-point link** - If you're trying to bridge two specific locations, a directional yagi (commonly 6-10 dBi for compact 915 MHz models) extends range significantly. **FCC compliance:** on US 915 MHz, FCC 15.247(b)(4)(i) requires you to reduce conducted transmit power 1 dB for every dB of antenna gain above 6 dBi - and, unlike the 2.4 GHz band, the 902-928 MHz band has *no* point-to-point exception that relaxes this. For example, with a 12 dBi yagi you must drop conducted power roughly 6 dB below 1 W to stay within the EIRP limit. Most firmware lets you set TX power accordingly; do not run full power behind a high-gain antenna.

What External Antenna to Buy

For most fixed outdoor deployments, a 915 MHz fiberglass omnidirectional antenna is the right choice:

- **Taoglas TI.92.2113 (3 dBi)** - \$15-20, compact, good for moderate ranges
- **Proxicast 5 dBi (ANT-DB5-5)** - \$25-35, good all-around outdoor omni
- **Taoglas FXP73 (5 dBi, mag base)** - \$25-40, great for vehicle or temporary mounts
- **L-com HG908U-PRO (8 dBi)** - \$45-60, excellent for high-gain omni backbone nodes

Connector Adapters

Most LoRa boards use a standard SMA connector (male pin on the antenna/pigtail, female body on the board) or a u.FL connector. External antennas typically use an N-connector or SMA. **Watch out:** SMA and RP-SMA (reverse-polarity SMA) look almost identical but do not mate - RP-SMA swaps the center pin and socket, so an SMA antenna will not connect to an RP-SMA board (and vice versa). Check which gender and polarity your specific board revision uses before ordering a pigtail or antenna. Match your connectors:

- **Heltec V3, T-Beam:** SMA female on board - use SMA male on pigtail or antenna (verify your revision, as some units ship RP-SMA)
- **RAK4631:** u.FL (IPEX) connector - needs u.FL to SMA pigtail (~\$5) to connect to any standard antenna
- **T-Deck, T-Echo:** SMA female - use SMA male pigtail or direct-connect SMA antenna

What is the difference between dBi and dBd antenna gain?

Antenna gain specifications use two different reference points - dBi and dBd - and confusing them leads to incorrect [link budget calculations](#). Here's what each means and how to convert between them.

The Reference Antennas

- **dBi (decibels relative to isotropic)** - Compares gain to a theoretically perfect isotropic radiator (a point that radiates equally in all directions - a perfect sphere). This is a theoretical reference that doesn't exist in practice.
- **dBd (decibels relative to dipole)** - Compares gain to a half-wave dipole antenna, which is the most common practical antenna type and a natural reference for antenna engineers.

The Conversion

$$\text{dBi} = \text{dBd} + 2.15$$

Examples:

$$0 \text{ dBd (dipole reference)} = 2.15 \text{ dBi}$$

$$3 \text{ dBd} = 5.15 \text{ dBi (approximately 5 dBi)}$$

$$5.85 \text{ dBd} = 8 \text{ dBi}$$

$$9 \text{ dBd} = 11.15 \text{ dBi (approximately 11 dBi)}$$

Which is Used in Practice?

Most commercial antenna manufacturers use dBi because the numbers look higher (marketing benefit). For the 902-928 MHz ISM band that matters here, FCC Part 15 expresses its EIRP and antenna-gain limits using the isotropic (dBi) reference - so convert any dBd spec to dBi (add 2.15) before checking it against the 4 W (36 dBm) EIRP ceiling or the 6 dBi antenna-gain threshold. Most link budget calculators accept either unit, as long as you're consistent.

Rule of thumb: When comparing antennas, make sure you're comparing the same units. A "5 dBd" antenna and a "5 dBi" antenna are NOT equivalent - the dBd antenna is 2.15 dB better. This difference can mean the difference between a reliable link and a marginal one.

Practical Antenna Gain Reference

Antenna Type	Typical Gain (dBi)	Typical Gain (dBd)
Stock rubber duck	~0 to 2 dBi	~-2 to 0 dBd
Quarter-wave with ground plane	~5 dBi (ideal ground plane; less in practice)	~2.85 dBd
Half-wave dipole	2.15 dBi	0 dBd
5/8 wave vertical	4-5 dBi	2-3 dBd
3-element yagi	7-8 dBi	5-6 dBd
5-element yagi	10-11 dBi	8-9 dBd
Commercial 5 dBi fiberglass	5 dBi	2.85 dBd
Commercial 8 dBi fiberglass	8 dBi	5.85 dBd

Note: a quarter-wave monopole over an *ideal* (infinite, perfectly conducting) ground plane radiates into a half-space and so has roughly 3 dB more gain than a dipole - about 5 dBi. Real, finite ground planes deliver less than this, but it is not equal to a plain dipole. Use this table as the single canonical reference for stock-antenna gain figures across the wiki.

What Gain Actually Buys You

Every 3 dB of additional gain (all else equal) doubles the effective radiated power. Because free-space range scales with the square root of the power ratio ($\text{range} \propto \sqrt{\text{EIRP}}$), gain translates to range as:

- 3 dB gain improvement $\approx$ 41% range increase in free space ($\sqrt{2} = 1.41x$)

- 6 dB gain improvement $\approx$ 100% range increase / double in free space ($\sqrt{4} = 2x$)
- 10 dB gain improvement $\approx$ 216% range increase in free space ($\sqrt{10} = 3.16x$)

These are free-space figures. In practice real-world gains are lower due to terrain and building losses, and higher-gain antennas are also constrained by the 4 W (36 dBm) EIRP limit - you often cannot legally or usefully realize the full theoretical range gain. Still, the relative improvement from a better antenna (within the legal limit and with good siting) is significant.