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 is 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 exisitng 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}`:

```
┌──────────────────┬──────────────────────────┐
│ path.hash.mode │ Advert path hash size │
```

```
├──────────────┼──────────────────────────┤
| 0 | 1 byte (default) |
├──────────────┼──────────────────────────┤
| 1 | 2 bytes |
├──────────────┼──────────────────────────┤
| 2 | 3 bytes |
└──────────────┴──────────────────────────┘
```

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.

Revision #4
Created 2026-05-03 05:05:08 UTC by Mesh America Admin
Updated 2026-06-08 19:46:49 UTC by Mesh America Admin