

Meshtastic Range Testing Guide

Overview

Systematic range testing goes beyond "does it connect?" - it quantifies signal quality, identifies path bottlenecks, and produces evidence you can use to justify infrastructure decisions. This guide covers the four primary tools available for characterizing mesh coverage and a recommended workflow for comprehensive coverage analysis.

Understanding Signal Quality Metrics

Metric	Excellent	Good	Marginal	Likely Failure
SNR	> +10 dB	+5 to +10 dB	0 to +5 dB	< -5 dB
RSSI	> -90 dBm	-90 to -110 dBm	-110 to -120 dBm	< -125 dBm

Key point: SNR matters more than RSSI. LoRa can decode signals well below the noise floor - a weak signal in a quiet RF environment (high SNR) will decode reliably even at very low RSSI. Focus on SNR first.

Note: these bands are practical app-display heuristics, not hard thresholds. LoRa can still decode down to roughly -20 dB SNR at high spreading factors (SF11/SF12), so a low SNR is not automatic failure - a -15 dB packet at SF11/12 can still get through.

Tool 1 - Built-In [Range Test Module](#)

The range test module sends periodic test packets and logs which nodes receive them, along with signal metrics. It is the most systematic way to characterize coverage between a fixed location and a node moving through the area.

CLI Setup

On the **sender** (the fixed/infrastructure node), enable the module and set the sender interval:

```
meshtastic --set range_test.enabled true
meshtastic --set range_test.sender 30
```

The second command sets sender interval to 30 seconds. The node will transmit a test packet every 30 seconds. On the **receiver** (the mobile node), enable `range_test.enabled` but leave `range_test.sender` at 0 (off) so it only listens and logs.

App Setup

1. Open the Range Test module settings. On **Apple** this is Settings > Module Configuration > Range Test; on **Android** it is Settings > Range Test.
2. Enable the module
3. Set sender interval
4. Activate **Sender** mode on the fixed node; leave the mobile node in receiver mode (module enabled, sender off)

Procedure

1. Configure the fixed-location node as the **sender** (range_test enabled, sender interval set) - typically a repeater or infrastructure node. It transmits sequential test packets.
2. Configure a second node as the **receiver** (range_test enabled, sender off, GPS on) carried by a person or vehicle. The receiver logs each received packet together with its GPS position.
3. Walk or drive away from the sender. The mobile receiver logs each packet with SNR, RSSI, and GPS coordinates.
4. Export logs to CSV for analysis. (Saving the CSV to onboard flash is ESP32-only; app clients export the log separately, e.g. from the Android Debug Log.) The resulting CSV can be imported into Google My Maps to visualize coverage.

Tool 2 - Trace Route

Trace Route reveals the actual path packets take through the mesh and reports per-hop signal quality. Use it to identify which routers packets are traversing and where bottlenecks are.

CLI

```
meshtastic --traceroute !nodeId
```

App

Long-press a node in the node list → select **Trace Route**.

Output

Trace Route shows each hop in the path, with the **SNR** reported per hop (firmware ≥ 2.5). Per-hop RSSI is *not* carried in the traceroute payload - only SNR per link is returned. Use this to:

- Confirm which repeaters are actually routing your traffic
- Identify weak links in multi-hop paths
- Verify that a new repeater is being used as expected

Tool 3 - Signal Metrics in Node List

The node list provides a quick snapshot of signal quality for all recently heard nodes - useful for baseline assessment without active testing.

CLI

```
meshtastic --nodes
```

Shows

- SNR and RSSI for the last received packet from each node
- Hop count
- Time since last activity
- GPS distance (if both nodes have GPS)
- Battery percentage

Tool 4 - MeshMapper Wardriving (MeshCore)

MeshMapper is a MeshCore wardriving tool (available for Android, iOS, and web), not a Meshtastic module. It combines GPS-tagged signal data into a visual coverage heatmap - ideal for documenting the measured RF coverage of a MeshCore network across a neighborhood, event venue, or service area. If you are mapping a Meshtastic network instead, use the Meshtastic Map view or a community Meshtastic mapper; this tool is included here as the MeshCore equivalent.

Setup

1. Install **MeshMapper** (Android, iOS, or web).
2. Connect a **MeshCore** device over BLE. (MeshCore devices support only one BLE connection at a time, so disconnect the MeshCore companion app first.)
3. Drive, walk, or cycle through the area you want to map.
4. MeshMapper logs GPS coordinates with signal strength for each received packet.
5. Upload to generate a coverage heatmap.

Heatmap Colors

Color	Meaning
Green	Strong signal
Yellow	Marginal signal
Red	Weak signal
Blank	No coverage detected

Recommended Testing Workflow

1. **Baseline:** Check node list (Tool 3) for current signal quality across known nodes.
2. **Path analysis:** Run trace routes (Tool 2) to all infrastructure nodes - confirm expected routing, identify weak hops.
3. **Coverage measurement:** Deploy range test module (Tool 1) for systematic point-to-area coverage data from each infrastructure node.
4. **Area mapping:** Conduct MeshMapper wardriving (Tool 4) for a comprehensive geographic coverage picture of a MeshCore network.

5. **Ongoing monitoring:** Establish MQTT monitoring to continuously log SNR/RSSI from infrastructure nodes - enables detection of degraded links before users report problems.
-

Revision #5

Created 2026-05-03 03:34:41 UTC by Mesh America Admin

Updated 2026-06-09 22:42:37 UTC by Mesh America Admin