

Range Test Module

The Range Test module automates signal strength measurement for deployment validation - letting you map exactly where in your coverage area packets arrive successfully, and at what SNR and RSSI values.

What the Range Test Module Does

Range Test operates as a sender/receiver pair:

- **Sender node** - Broadcasts a test packet at a configurable interval, incrementing a sequence number each time. The test packet carries the sender's position.
- **Receiver node** - Any node with the module enabled can receive and log each packet's SNR, RSSI, and sequence number. The canonical GPS-tagged `rangetest.csv` is written to the flash of an **ESP32-based receiver** (saving is only available on ESP32 boards).
- **Output** - A CSV file (`rangetest.csv`) saved to the ESP32 receiver's internal flash, containing all received packets with position data, signal quality, and sequence numbers. Missing sequence numbers identify packet loss. The Apple/Android apps also offer their own separate position-log exports.

Setting Up a Range Test

Configure the Sender Node

```
meshtastic --set range_test.enabled true
meshtastic --set range_test.sender 60
```

`sender` is the interval in seconds between test packets. 60 seconds works well for driving tests; 30 seconds for walking tests where you move slower.

Configure the Receiver

```
meshtastic --set range_test.enabled true
meshtastic --set range_test.save true
```

With `save true` on an ESP32-based device, received packets are logged to a file called `rangetest.csv` in the device's internal flash memory (no SD card or smartphone required). Retrieve it over WiFi by browsing to `meshtastic.local/rangetest.csv`, which downloads the file automatically. Saving is only available on ESP32 boards; nRF52 boards cannot write the CSV.

Disable Range Test after testing. Leaving `range_test.enabled` on keeps the node broadcasting extra position-bearing test packets - that adds needless airtime and location exposure. Set `range_test.enabled false` when the test is complete.

Conducting a Range Test Drive

1. Place the sender node at your repeater location or test deployment point. Ensure it has GPS lock and is transmitting.
2. Configure an ESP32-based portable node as the receiver (so it can save the CSV to flash).
3. Drive or walk through your intended coverage area.
4. After the test, retrieve the CSV file from the receiver at `meshtastic.local/rangetest.csv`. Each row contains: timestamp, GPS lat/lon, SNR, RSSI, sequence number.
5. Import the CSV into Google Maps (My Maps), QGIS, or any mapping tool to visualize coverage.

Interpreting Results

The bands below are rough rules of thumb, not a Meshtastic-published spec. The actual SNR floor for decoding shifts with the modem preset / spreading factor, so treat the RSSI and SNR columns as approximate and read them alongside the SF note that follows.

RSSI	SNR	Connection Quality
-80 to -100 dBm	>5 dB	Excellent - reliable delivery
-100 to -115 dBm	0 to 5 dB	Good - occasional packet loss
-115 to -125 dBm	-5 to 0 dB	Marginal (preset-dependent)
Below -125 dBm	near the SF floor*	Edge of range - unreliable

* Note: The "edge of range" SNR is **spreading-factor-dependent**, not a flat -10 dB. LoRa can decode packets at negative SNR values - roughly -7.5 dB at SF7 down to approximately -20 dB at SF12. On the default LongFast preset (SF11) the decode floor is about -17 dB, so a -10 dB SNR link on LongFast is still well within usable range, not the edge. This is one of LoRa's most remarkable properties. RSSI alone is not the full picture; low RSSI with high SNR can still be a reliable link. (For consistency, reconcile these thresholds with the [reading-network-statistics](#) and [neighbor-info](#) pages, which should use the same canonical bands.)

Using Range Test for Repeater Placement Decisions

Deploy a temporary repeater at a candidate site, run a range test drive across the intended coverage area, then compare the CSV output against a test from your next-best candidate site. This gives objective, data-driven evidence for repeater placement decisions rather than guessing based on map topology alone.

Revision #3

Created 2026-05-03 05:41:26 UTC by Mesh America Admin

Updated 2026-06-09 12:13:09 UTC by Mesh America Admin