Traceroute and Path Diagnostics

What Is Meshtastic Traceroute?

Traceroute is a diagnostic feature built into Meshtastic firmware that lets you discover the path a packet takes through the mesh to reach a destination node. Unlike a simple ping, traceroute collects the ID of each intermediate node that relayed the packet, along with signal quality data for each hop. Treat the result as one probe's path, not a definitive map: like normal traffic, a traceroute probe's route can differ from where your real messages flow, so it shows how this particular packet was routed rather than a guaranteed picture of all traffic.

How It Works

When you initiate a traceroute, your node sends a special packet addressed to the destination node ID. Each relaying node that knows the channel's encryption key appends its own node ID and the SNR at which it received the packet to the route record (this is not limited to nodes configured with a ROUTER role - any rebroadcasting node that shares the channel key is recorded). From firmware version ≥ 2.5, the route back to the origin is also recorded, with the SNR for each link. When the destination receives the packet, it sends the complete hop list back to you. The result is an ordered list: source → relay 1 → relay 2 → ... → destination.

Traceroute uses Meshtastic's managed flood-routing mechanism (see the mesh-algorithm docs), the same mechanism as normal traffic, but the route-record payload (the RouteDiscovery protobuf) grows with each hop. This means traceroute is more bandwidth-intensive than a regular message and should be used sparingly on busy networks.

Running Traceroute from the App

In the Meshtastic Android or iOS app:

1. Open the **Node List** and tap the target node.

2. Tap the three-dot menu or the node detail view.
3. Select **Traceroute**.
4. Wait for the result - it may take 10 - 30 seconds depending on hop count and channel utilization.

The result displays each hop's node ID and the SNR value observed at that hop.

Running Traceroute from the CLI

```
meshtastic --traceroute '!a1b2c3d4'
```

Replace `!a1b2c3d4` with the destination's node ID (the exclamation-mark prefix is required). **Single-quote the node ID** as shown - in bash the bare `!` triggers history expansion and you will get an "event not found" error otherwise. The CLI prints each hop as it is received:

```
Traceroute to !a1b2c3d4:
 You (!ff00ee11) --> !11223344 (SNR: 8.25) --> !55667788 (SNR: 4.50) --> !a1b2c3d4
(destination)
```

Interpreting the Output

- **Hop count:** fewer hops generally means lower latency and higher reliability.
- **SNR values:** higher is better, but there is no single fixed "weak link" cutoff - the usable SNR floor is **spreading-factor-dependent**. LoRa can demodulate well below 0 dB: roughly -7.5 dB at SF7 down to about -20 dB at SF12. On the default LongFast preset (SF11) the decode floor is around -17 dB, so a hop at -10 dB on LongFast is still usable with margin. A link is weak only when its SNR is within a few dB of the preset's floor (on LongFast, worse than roughly -14 dB is marginal). See a Semtech LoRa SNR demodulation table for the per-SF limits.
- **Missing hops:** if the traceroute times out without a complete result, the destination may be unreachable or the return path failed.

Using Traceroute to Diagnose Coverage Gaps

Run traceroute from multiple points in your network to map which nodes are serving as critical relays. If a single node appears in most traceroute paths, it is a single point of failure - consider adding a redundant node nearby. An unexpectedly long hop count (e.g. 5 hops when you expect 2) suggests a shorter path exists but is not being used, possibly due to SNR thresholds or a misconfigured router role.

Comparing Two Runs to Detect Path Changes

Run traceroute to the same destination before and after a node failure or configuration change. A change in the hop list confirms that traffic is now routing differently. This is especially useful after repositioning a relay node or changing antenna - it lets you verify the improvement objectively rather than relying on signal reports alone. Bear in mind each run reflects one probe's path, so compare several runs rather than reading a single result as definitive.

Revision #4
Created 2026-05-03 06:44:45 UTC by Mesh America Admin
Updated 2026-06-09 12:14:42 UTC by Mesh America Admin