

Network Planning

Coverage planning, node placement, hop budgets, and building a reliable community mesh.

- [☐ Start Here — Network Planning Guide](#)
- [Coverage Planning Tools](#)
 - [Coverage Planning Tools Overview](#)
 - [MeshMapper Wardriving Guide](#)
 - [Meshtastic Range Testing Guide](#)
 - [RF Coverage Prediction Tools](#)
 - [Field Testing and Coverage Verification](#)
 - [Mesh Network Capacity and Congestion](#)
- [Advanced Configuration](#)
 - [Path Hash Modes \(MeshCore\)](#)
 - [Regional Scoping with ISO Codes \(MeshCore\)](#)
 - [Frequency Coordination and Channel Planning](#)
- [Telemetry & Monitoring](#)
 - [Environmental Sensors & Telemetry](#)
- [Link Budget & Propagation](#)
 - [Link Budget Calculations](#)
 - [RF Propagation Planning Tools](#)
- [RF Propagation Environments](#)
 - [Urban Propagation](#)
 - [Forest and Vegetation Propagation](#)
 - [Mountain and Complex Terrain](#)
 - [Water and Coastal Propagation](#)

- [Node Placement Strategy](#)
 - [Repeater Placement Principles](#)
 - [Coverage Radius Estimation by Terrain Type](#)
 - [The Repeater Grid Approach for Urban Coverage](#)
 - [Designing for Redundancy](#)
- [Hop Budget and Routing](#)
 - [Understanding Hop Count and Hop Limits](#)
 - [Designing for Multi-Hop Reliability](#)
- [Network Monitoring and Management](#)
 - [Building a Mesh Network Dashboard](#)
 - [Using meshmap.net and Community Maps](#)
 - [Mesh Network Change Management](#)
- [Spectrum Analysis and RF Tools](#)
 - [Using an SDR for 915 MHz Band Analysis](#)
 - [NanoVNA Guide for Mesh Antenna Work](#)
- [Channel and Frequency Planning](#)
 - [Meshtastic Channel Number Selection Guide](#)
 - [Operating Multiple Channels on One Network](#)

☐☐ Start Here — Network Planning Guide

This book is for people designing and managing community mesh networks - coverage planning, RF propagation, node placement strategy, monitoring, and advanced configuration.

☐☐ What's Your Goal?

- **Planning coverage for a new area:** Start with [Coverage Planning Tools Overview](#)
- **Optimizing an existing network:** Start with [Building a Mesh Network Dashboard](#)
- **Placing new repeaters:** Start with [Repeater Placement Principles](#)

☐☐ What's In This Book

Coverage Planning

- [Coverage Planning Tools Overview](#)
- [RF Coverage Prediction Tools](#) - Heywhatsthat, Radio Mobile, etc.
- [MeshMapper Wardriving Guide](#)
- [Field Testing and Coverage Verification](#)

RF Propagation

- [Urban Propagation](#)
- [Forest and Vegetation Propagation](#)
- [Mountain and Complex Terrain](#)
- [Water and Coastal Propagation](#)
- [Link Budget Calculations](#)

- [RF Propagation Planning Tools](#)

Node Placement

- [Repeater Placement Principles](#)
- [Coverage Radius Estimation by Terrain Type](#)
- [The Repeater Grid Approach for Urban Coverage](#)
- [Designing for Redundancy](#)

Channel and Frequency Planning

- [Frequency Coordination and Channel Planning](#)
- [Meshtastic Channel Number Selection Guide](#)
- [Operating Multiple Channels on One Network](#)

Network Monitoring

- [Building a Mesh Network Dashboard](#)
- [Using meshmap.net and Community Maps](#)
- [Mesh Network Capacity and Congestion](#)

RF Tools

- [Using an SDR for 915 MHz Band Analysis](#)
- [NanoVNA Guide for Mesh Antenna Work](#)

➔ Related Books

- [Antennas & RF](#) - Antenna fundamentals and testing
- [MeshCore Repeaters](#) / [Meshtastic Repeaters](#) - Deploying nodes
- [Starting a Community Mesh](#) - Community and governance side

Coverage Planning Tools

Coverage Planning Tools Overview

Coverage Planning Tools Overview

Before deploying hardware, use these tools to estimate coverage, find line-of-sight paths, and identify gaps in your network. Most are free and browser-based.

MeshCore-Specific Tools

Tool	URL	What It Does
NoDakMesh Node Planner	nodakmesh.org/tools/node-planner	Satellite + topographic map with line-of-sight analysis and live MeshCore node visibility. Best starting point for North American planning.
MeshCore Live Map	map.meshcore.io	Live worldwide visualization of MeshCore nodes that have reported their location.

Meshtastic Tools

Tool	URL	What It Does
Meshtastic Site Planner	site.meshtastic.org	Estimates theoretical RF coverage from a location using terrain data. Enter coordinates and antenna height.
Meshtastic World Map	meshmap.net	Real-time map of Meshtastic nodes worldwide that report to the public MQTT broker.

General RF Planning Tools

Tool	URL	What It Does
HeyWhatsThat	heywhatsthat.com	Radio horizon visualization using terrain elevation data. Enter a location and see what is visible from that point.
Radio Mobile Online	radiomobileonline.pe1mew.nl	Advanced RF propagation modeling with configurable transmit power, antenna gain, and receiver sensitivity. Supports point-to-point and coverage area analysis.
SCADACore RF Line-of-Sight	scadacore.com	RF line-of-sight tool for quick path analysis between two coordinates.

Planning Workflow

1. **Check existing nodes:** Start with map.meshcore.io or meshmap.net to see what already exists near your target area.
2. **Identify candidate sites:** Use heywhatsthat.com to find hilltops, buildings, or towers with broad radio horizons.
3. **Model the path:** Use Radio Mobile or the NoDakMesh node planner to estimate signal strength on key links.
4. **Validate with MeshMapper:** After deployment, use MeshMapper wardriving to map actual measured coverage (see [MeshMapper Wardriving Guide](#)).

MeshMapper Wardriving Guide

MeshMapper Wardriving Guide

MeshMapper is a platform for mapping *actual measured* RF coverage of a MeshCore network - not just node locations, but real signal coverage at road level. This is the ground truth that theoretical planners cannot provide. MeshMapper is a MeshCore tool (Android, iOS, and web); for Meshtastic coverage mapping see meshmap.net or the Meshtastic Site Planner instead.

Getting the App

- **Android:** Google Play - search "MeshMapper"
- **iOS:** App Store - search "MeshMapper"
- **Web interface:** wd.meshmapper.net
- The app is **free**.

Connecting Your Radio (BLE)

For wardriving, MeshMapper connects directly to your MeshCore radio over **Bluetooth LE (BLE)** - it does not require an MQTT connection to collect coverage data. Pair your MeshCore device to the app over BLE before starting a session.

Important: MeshCore devices only support one BLE connection at a time. **Disconnect the MeshCore companion app** before launching MeshMapper or the connection will fail.

Observer / Region-Admin Setup (Optional)

Separately from the per-user wardriving flow above, an MQTT observer can be configured to aggregate mesh traffic for region administrators. This is **not** part of normal per-user wardriving. If you are running an observer, connect to one of these brokers:

- `mqtt-us-v1.letsmesh.net:443` (WebSocket TLS)
- `mqtt.meshmapper.cc:443` (WebSocket TLS)

Operating Modes

Mode	Description	Best For
Hybrid (recommended)	Alternates discovery requests and channel messages. 50% fewer transmissions than legacy Active mode.	General wardriving - balances coverage data quality with network impact
Passive	Discovery requests every 30 seconds; no channel messages.	Minimal network impact; good for densely populated mesh areas
Manual Ping	Single on-demand ping.	Spot-checking coverage at a specific location without driving
Active	Legacy channel-message-only mode.	Backward compatibility - Hybrid is superior in all cases
Trace	Focus on a single repeater identified by its hex node ID.	Antenna alignment, diagnosing a specific repeater's coverage, post-installation validation

Coverage Map Color Meanings

Color	Code	Meaning
Green	BIDIR	Two-way confirmed contact - gold standard coverage. Your device and the repeater can hear each other.
Cyan	DISC	Discovery-based two-way confirmation - confirmed via discovery protocol rather than channel message.
Orange	TX	Transmit-only path - your signal reaches the repeater but return path is incomplete (asymmetric link).
Purple	RX	Receive-only - you can hear the repeater but it cannot hear you.
Grey	DEAD	Signal heard but not relayed - marginal contact; unreliable for mesh routing.
Red	DROP	No repeater responded - no coverage at this location.

Wardriving Best Practices

- Mount your device with the antenna as high as practical in the vehicle (dashboard or roof magnet mount).
- Drive at normal road speeds - the app samples frequently enough.
- Cover roads in a grid pattern for systematic area mapping.
- Use **Trace mode** after installing a new repeater to confirm its actual coverage footprint.
- Share your coverage data back to the community map so others benefit.

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.

RF Coverage Prediction Tools

Why Model Before Deploying

Walking a coverage area with a radio after installing a repeater is valuable ground-truth - but it is expensive if the site turns out to be wrong. Free online tools let you model RF line-of-sight and rough coverage before committing to an installation, saving you a wasted site visit and hardware move.

HeyWhatsThat (heywhatsthat.com)

HeyWhatsThat is the best free tool for quickly visualising the radio horizon from a specific point.

- Enter the coordinates (or click a map) and elevation of your proposed repeater site.
- The tool generates a panoramic view and a map showing all terrain that is geometrically visible from that point.
- Accounts for Earth curvature and terrain elevation using SRTM data.

Limitations: HeyWhatsThat's SRTM data is a digital *surface* model (coarse ~30 m resolution), so it partially includes building rooftops and tall canopy but cannot resolve individual buildings or trees. Treat its results as terrain-dominated geometric line-of-sight, not true RF coverage - it does not model actual RF propagation, diffraction, or local clutter.

Use case: Quickly screen potential hilltop sites before visiting. A site with a 360° clear horizon is worth investigating; a site blocked by higher terrain in key directions is a red flag.

Radio Mobile (radiomobile.ca)

Radio Mobile is a more advanced free tool aimed at amateur and community radio network planning.

- **Point-to-point path profiles:** shows the terrain cross-section between two points with Fresnel zone visualisation, so you can see whether the path is truly clear.
- **Coverage maps:** generate area coverage maps from a repeater site given your TX power and antenna height.
- Supports custom frequency, antenna gain, and receiver sensitivity input - LoRa parameters can be modelled directly.

Radio Mobile has a steeper learning curve than HeyWhatsThat but is much more capable for serious network planning.

SPLAT! (Amateur Radio Propagation Tool)

SPLAT! is an open-source Linux/macOS command-line tool for RF propagation analysis.

- Generates coverage maps using the ITM (Irregular Terrain Model) or ITWOM propagation model.
- More accurate than simple line-of-sight tools - accounts for diffraction over ridges.
- Requires downloading SRTM terrain data files for your region.

SPLAT! is best suited to serious network planners who are building a community mesh from scratch and need repeatable, scriptable coverage analysis.

Meshtastic Signal Mapper

Some community tools (including extras around meshmap.net) allow you to import Meshtastic position data and visualise actual observed coverage on a map. Where real-world data already exists, this is more valuable than any theoretical model. Check your regional Meshtastic community's resources for existing coverage maps before starting your own modelling work.

CloudRF / ORCA (Cloud-Based RF Planning)

CloudRF is a commercial service with a free tier that generates RF coverage maps using the ITM/Longley-Rice model.

- Can account for clutter (buildings, forest) in some calculation modes.
- Easy web interface: enter site coordinates, antenna height, frequency, TX power, and antenna gain to get a shaded coverage map.
- The free tier allows a limited number of calculations per month - sufficient for planning a handful of repeater sites.

CloudRF is a good middle ground between the simplicity of HeyWhatsThat and the complexity of SPLAT! for planners who want clutter-aware coverage maps without installing local software.

Field Testing and Coverage Verification

Why Field Test?

Even the best RF prediction tools are only as good as their terrain models. Buildings, vegetation, and local obstructions can significantly degrade predicted coverage. Field testing confirms what the models predict - and reveals the surprises they miss.

Basic Field Test Procedure

1. Deploy the repeater at the proposed site, even temporarily on a tripod.
2. Walk or drive the coverage area with a second device (phone running the [Meshtastic app](#), or a handheld node).
3. Record signal metrics at known locations: SNR (signal-to-noise ratio) and RSSI (received signal strength indicator).
4. Note locations where packets stop getting through - this defines your coverage boundary.
5. Compare results to your predicted coverage map and identify gaps or surprises.

Understanding SNR and RSSI for LoRa

RSSI is received signal power in dBm. For mid-range spreading factors, useful values run roughly -90 to -120 dBm and reliability degrades below about -120 dBm. But -120 dBm is *not* a hard floor: at high spreading factors (SF11/SF12, the Long Fast and Long Slow presets) the SX1262 decodes reliably down to roughly -131 to -137 dBm (BW 125 - 250 kHz). The usable RSSI floor depends on the active modem preset, so don't reject an otherwise-usable high-SF link just because RSSI dipped past -120 dBm.

SNR is signal-to-noise ratio in dB. LoRa's key advantage over conventional radios is its ability to decode packets at *negative* SNR values, and how far negative depends on the spreading factor. The theoretical demodulator SNR floor is about -7.5 dB at SF7, about -17.5 dB at SF11, and about -20 dB at SF12. So the SNR at which a link fails is preset-dependent: a Long Slow (SF12) link decodes well below where a fast, low-SF (SF7) link would already be failing.

Rule of thumb (these bands are preset-dependent - treat them as a rough guide and interpret SNR against your active modem preset, not as universal thresholds):

- SNR > 0 dB - strong signal on any preset.
- -5 to -10 dB - marginal on fast/low-SF presets (e.g. ShortFast), but still solidly decodable on Long Fast (SF11) or Long Slow (SF12).
- Below -15 dB - near the noise floor and likely to lose packets on low-SF presets, but long-range high-SF presets routinely decode down toward -20 dB SNR, so -15 dB is still usable on Long Fast / Long Slow.

For planning purposes it can help to adopt a single conservative marginal-SNR floor (around -15 dB) across the book, while remembering the true decode floor is lower at high SF (SF11 ≈ -17.5 dB, SF12 ≈ -20 dB).

Reading Signal Data in Meshtastic

The Meshtastic app shows RSSI and SNR for each received packet in the message details view. This is real link-quality data from your actual deployment - use it as your primary signal source during field testing. For multi-hop paths, the Meshtastic **traceroute** feature reports the **SNR per hop** (not RSSI), which is useful for finding the weakest link in a chain of repeaters.

Recording a Coverage Map

Simple approach: note GPS coordinates alongside SNR and RSSI values in a spreadsheet while driving, then export the data to Google My Maps or another mapping tool.

Automated approach: enable the Meshtastic [Range Test module](#) (Config → Module → Range Test). It automatically logs positions with signal data to a CSV file as you walk or drive the area. In a range test the fixed node acts as the Sender and the mobile node acts as the Receiver - the Receiver is the one that logs the CSV with GPS.

Visualisation: import the CSV into Google My Maps or QGIS to produce a signal-strength overlay map that you can compare directly against your predicted coverage.

What to Look For

- **Expected coverage but no packets:** interference, antenna issue, wrong modem preset, or an obstruction absent from the terrain model.
- **Better-than-expected coverage:** diffraction over a ridge, reflections from buildings or open water.
- **Unexpected dead zones:** metal structures, buildings with metal roofing, or terrain features not captured in SRTM data (SRTM is coarse ~30 m surface-model data, so small obstructions are easily missed).

Iterating on the Design

If field testing reveals coverage gaps, consider: adding a second repeater as a relay node, increasing antenna height, or repositioning the existing repeater. A 10-metre increase in antenna height can eliminate a surprisingly large dead zone - elevation is often more valuable than additional TX power.

Mesh Network Capacity and Congestion

LoRa Channel Capacity

LoRa is a low-data-rate technology. Unlike Wi-Fi, the RF channel is shared by all nodes simultaneously using a CSMA-like approach combined with Meshtastic's **managed flooding** mesh mechanism (every node rebroadcasts packets that still have hop limit remaining; there is no routing table). Understanding channel capacity helps you design a network that doesn't saturate itself.

Airtime Utilisation Metrics

Meshtastic displays **Channel Utilization** and **Air Utilization** percentages in the app. These are your primary indicators of network load. Channel Utilization is measured over a rolling 1-minute window, and the app colour-codes it: green below 25%, orange 25 - 50%, and red above 50%.

- **Channel Utilization < 25% (green):** healthy. Around the 25% mark the firmware begins to self-throttle - it *defers* its own transmissions to avoid colliding on a busy channel, so packets are delayed (queued), not dropped outright.
- **Channel Utilization 25 - 50% (orange):** busy/degrading. Expect added latency as nodes back off and defer.
- **Channel Utilization > 50% (red):** congested. Sustained load at this level risks queue overflow and real packet loss.
- **Air Utilization TX:** the fraction of time *your* node spends transmitting. Keep it low (a few percent is a reasonable target); a high value means your node is contributing a lot of airtime - reduce its broadcast intervals. (There is no documented hard "15%" Air Utilization rule.)

Sources of Traffic in a Mesh

- **Position and NodeInfo broadcasts:** nodes announce themselves periodically. Position and NodeInfo are *separate* broadcast types sent at different default intervals (NodeInfo is typically far less frequent than position). As a rough example, 50 nodes each sending a position packet on a 15-minute interval is one position packet roughly every 18 seconds ($50 \div 900 \text{ s}$) on average - before any user traffic, and before the additional NodeInfo, telemetry, and rebroadcast load. (Verify the exact default intervals against your current firmware/preset, as they vary by role and modem preset.)
- **User messages:** text traffic generated by operators.
- **Telemetry:** device metrics (battery, voltage) and environmental sensor readings.
- **ACKs and routing overhead:** mesh protocol housekeeping packets.

Traffic Reduction Strategies for Dense Networks

- Increase position broadcast interval to 30 - 60 minutes on non-mobile nodes.
- Disable telemetry on nodes where it is not needed.
- Use the **Medium Slow** or **Medium Fast** preset - the higher data rate means each packet occupies the channel for less time, even if physical range is unchanged.
- Be deliberate about infrastructure roles. Both **ROUTER** and **REPEATER** rebroadcast with *elevated* priority (they will rebroadcast even after hearing another node do so), so neither is "less aggressive" than the other. The practical difference is that a REPEATER is headless - it does not run the client app or participate in the node database - not that it retransmits less. To actually cut redundant airtime, reduce the *number* of high-priority infrastructure nodes in range of each other and keep ordinary nodes in the default CLIENT role rather than promoting many of them to ROUTER/REPEATER.
- Limit hop count: most community networks set a maximum of 3 - 5 hops (the firmware default is 3, max 7). Longer hop chains amplify traffic because each hop retransmits every packet.

The Hop Storm Problem

If many nodes retransmit the same packet, a single user message can trigger a burst of dozens of transmissions across the mesh. Meshtastic uses duplicate-packet detection to suppress already-seen packets and prevent routing loops, but in dense networks already near maximum airtime utilisation, a sudden burst of messages can temporarily saturate the channel and cause widespread packet loss. Keeping hop counts low and broadcast intervals long is the primary mitigation.

Monitoring Your Network

Use the Channel Utilization and Air Utilization figures in the [Meshtastic app](#) to assess local network load before and after adding new nodes. If deploying a new repeater in a dense area, monitor whether its addition increases congestion - a new high-visibility, high-priority infrastructure node can increase channel load for every other node in range.

Advanced Configuration

Path Hash Modes (MeshCore)

Path Hash Modes (MeshCore)

Path hash modes control how repeaters identify themselves in routing path headers within MeshCore packets. This feature is available in recent MeshCore firmware versions. Check your firmware release notes for version-specific availability.

Mode Comparison

Mode	Bytes per Hop	Max Hops	Unique IDs	CLI Setting
1-byte	1	64	256	<code>set path.hash.mode 0</code>
2-byte	2	32	65,536	<code>set path.hash.mode 1</code>
3-byte	3	21	16,777,216	<code>set path.hash.mode 2</code>

Mode 0 (1-byte) is the default.

Recommendation

Use **2-byte mode** (`set path.hash.mode 1`) for most North American community meshes (a common community recommendation, e.g. NodakMesh).

- **1-byte mode** provides 256 possible hash IDs. By the birthday paradox the chance of a collision reaches ~50% at only about 19 repeaters in range (well below 256), so with more than ~30 repeaters in range hash collisions become likely, causing routing errors and ambiguous path attribution.
- **2-byte mode** provides 65,536 possible hash IDs, making collisions vanishingly unlikely for any realistic community mesh while keeping per-hop overhead to 2 bytes.
- **3-byte mode** is future-proof but reduces maximum hop count to 21, which may be limiting in networks with long routing chains.

Mixing path hash modes within a single mesh requires all in-path repeaters to be running firmware $\geq$ **1.14**. On networks where some repeaters run older firmware, keep every device on the same (default) mode.

Configuration Steps

1. Connect to the repeater via USB serial (115200 baud) or Bluetooth
2. Set the desired mode:

```
set path.hash.mode 1
```

3. Broadcast updated hash to the network:

```
advert
```

App Access

In recent companion-app versions (around v1.41.0+; check current release notes, as the menu location may change between versions), path hash mode can also be adjusted under:

Settings → Experimental Settings

Note that "Experimental Settings" must be enabled in the app before this option appears.

Network-Wide Consistency

All repeaters and nodes on the same network segment do not need to use the same path hash mode - mode is a per-device setting affecting how that device encodes its own ID in path headers. However, for consistent diagnostics and network analysis, aligning all devices to the same mode simplifies troubleshooting, and mixing modes requires all in-path repeaters to be on firmware $\geq$ 1.14 (see above).

Regional Scoping with ISO Codes (MeshCore)

Regional Scoping (MeshCore Region Filtering)

MeshCore region scoping (region filtering) lets repeater operators tag flood traffic with a **region** so that repeaters only forward scoped packets within their configured region, reducing cross-country network congestion. A region is **not** an ISO 3166-2 code and there is no built-in standard, code table, or official registry inside the firmware. A region is simply an **arbitrary, operator-chosen plain name** that the firmware **SHA-256 hashes into a 2-byte transport code** carried in the packet header; repeaters match on that hash.

Communities often adopt ISO-3166-2-style strings such as `us` or `us-co` purely as a naming *convention*. These work only because they are valid arbitrary names - the firmware never interprets them as ISO codes, validates them against any standard, or enforces a country/state hierarchy. Any agreed string works equally well.

How It Works

- A region name is a string of **lowercase alphanumeric characters and hyphens** (`a-z`, `0-9`, `-`), maximum **~29 bytes** (UTF-8). Within a given mesh, region names must be unique, and they must match **exactly** - any misspelling is treated as a completely different region.
- The firmware hashes the name (SHA-256) down to a **2-byte** identifier. Because the identifier is only 2 bytes, hash collisions between different names are possible (though unlikely for a small set of regions).
- Legacy flood packets that carry *no* scope are still forwarded everywhere by default - region filtering only applies to packets that carry a scope.
- Since v1.10.0 a basic rule applies: a repeater will *not* forward a flood packet that has a scope set when the repeater has no matching region. Repeaters running firmware older than v1.10.0 do not honor scopes, so scoped traffic can still leak across them.

This lets a local community (for example, a Colorado group) agree on a region name and keep its scoped traffic within that region instead of flooding the wider mesh.

Configuration Notes

- The number of regions configurable per repeater depends on firmware version. Refer to MeshCore release notes for current limits.
- Region names: lowercase alphanumeric plus hyphen, maximum ~29 bytes (UTF-8), unique within a mesh, matched exactly.
- There is **no firmware-enforced minimum scope** (no required country + state). Any hierarchy - such as a broad `us` region plus a narrower `us-co` region - is purely a convention agreed among operators, not a rule the firmware enforces.

CLI Configuration Example (Colorado)

The names below (`us`, `us-co`) are arbitrary example strings, not standardized codes. Defining a region is not enough on its own - you also issue `region allowf` (allow flood) for each region to actually permit scoped flood forwarding, then `region save`:

```
region put us
region put us-co us
region allowf us
region allowf us-co
region save
```

This configures the repeater with two arbitrary regions (`us` and `us-co`) and enables flood forwarding for both. The names carry no inherent geographic meaning to the firmware - they match only because every participating operator uses the exact same strings.

Example Region Names (Convention Only)

The strings below are **examples of arbitrary names a community might agree to adopt**. They are not ISO 3166-2 codes recognized by MeshCore and are not validated by the firmware - they work only among networks that all agree to use them. Pick any short, agreed name for your local mesh; everyone in scope must use the identical string.

Example Name	Could Represent
<code>us</code>	United States (broad)
<code>us-co</code>	Colorado

Example Name	Could Represent
<code>us-nd</code>	North Dakota
<code>us-or</code>	Oregon
<code>us-wa</code>	Washington
<code>ca</code>	Canada (broad)
<code>us-dfw</code>	Dallas/Fort Worth metro
<code>us-bay</code>	San Francisco Bay Area metro
<code>us-atl</code>	Greater Atlanta metro

Agreeing on Region Names

There is **no central registry built into MeshCore**. Choosing region names is a coordination exercise: operators discuss in whatever forums they use (or even over the mesh itself) and come to a consensus on how to divide up their geography. Whatever string you settle on, every repeater that should share that region must be configured with the exact same name.

Third-party community sites such as regionmesh.com publish suggested naming conventions to help operators coordinate, but these are **independent community resources, not official MeshCore registries** - the firmware neither knows about nor enforces them.

After Changing Region Configuration

After saving region changes, run `advert` to broadcast the updated configuration to the network immediately rather than waiting for the next scheduled advertisement.

Frequency Coordination and Channel Planning

When multiple independent mesh networks coexist in the same geographic area, frequency and channel coordination prevents interference and allows for intentional interconnection where desired. Note: this is voluntary self-coordination among unlicensed Part 15 operators, not formal frequency coordination. You cannot reserve a frequency, and channel choices are limited to the fixed slots your preset bandwidth defines within 902-928 MHz.

The 902-928 MHz Band Structure

In the United States, the 902-928 MHz ISM band is 26 MHz wide. LoRa channel bandwidth in Meshtastic ranges from 125 kHz (SF12 Long-Slow) to 500 kHz (ShortTurbo); the common LongFast/MediumFast presets use 250 kHz. Because each transmission occupies only part of the band, many slots can coexist. Note: FCC 15.247(a)(2) digital modulation requires a minimum 6 dB bandwidth of 500 kHz, so compliance for the narrower LoRa bandwidths depends on the device's specific FCC type acceptance. Dense networks in the same area can still cause packet collisions if they share the same frequency slot.

Meshtastic Channel Numbers

Meshtastic derives its center frequency from a frequency-slot number and the preset bandwidth. Slot spacing equals the preset bandwidth — 250 kHz for LongFast (104 US slots), and 125 kHz only for the 125 kHz presets — so 250 kHz is the default spacing, not 125 kHz. Slots span the whole regional band (0-103 for US LongFast); when the slot is left at default it is chosen by hashing the channel name, not by a simple channel index. Note that these frequency slots are distinct from the up-to-8 logical channels (name/PSK), which all share the same frequency.

```
# US LongFast (250 kHz BW) default frequency:
# 906.875 MHz, which is slot 20 – selected by hashing
# the default channel name "LongFast" (not "channel 0").
#
# Raw formula for an explicit slot n (1-indexed):
```

```
# freq = 902.0 + 0.125 + (n-1)*0.25 MHz
```

```
# Set the frequency slot explicitly via CLI:
```

```
meshtastic --set lora.channel_num 3
```

MeshCore Frequency Selection

MeshCore uses fixed frequency presets. One commonly used US/Canada preset is 910.525 MHz (as of 2026-06-08; verify against current MeshCore release/docs, as there are multiple US presets at different SF/BW and frequencies — there is no single fixed USA/Canada frequency). Networks in the same area using MeshCore should coordinate to avoid using the same frequency if they don't want to interoperate.

Avoiding Interference with Other Users

The 902-928 MHz band is shared with ISM devices, FHSS systems, baby monitors, cordless phones, and other LoRa networks. If you observe high packet error rates that don't correspond to weak signal strength, interference from another source may be the cause. Use an SDR (Software Defined Radio) to scan the band and identify active signals.

Multi-Network Coordination

When two community networks exist in overlapping geographic areas, coordinate:

1. **Different channel keys with same frequency** - Networks stay logically separate even if they occupy the same frequency. Because they share the same frequency slot, they share airtime and can still collide; acceptable for low-traffic networks.
2. **Different frequencies + different keys** - Minimizes collisions between your two networks. Note: 902-928 MHz is unlicensed shared ISM spectrum (FCC Part 15); no operator owns a frequency and you have no legal protection from interference by other ISM/FHSS devices. Different center frequencies reduce, but do not guarantee elimination of, collisions. Also note that nodes on different frequency slots cannot hear each other.
3. **Intentional bridging** - A dual-radio or dual-channel node that bridges the two networks. Both networks benefit from the other's coverage.

Documentation for Multi-Network Areas

Maintain a local frequency coordination document shared between network operators:

Network	Frequency	Preset	Coverage Area	Contact
Portland Mesh	906.875 MHz	LongFast	Metro PDX	ops@pdxmesh.net
Columbia Gorge Mesh	907.125 MHz	LongFast	Hood River/The Dalles	k7xyz@arrl.net

Telemetry & Monitoring

Environmental Sensors & Telemetry

MeshCore nodes can be equipped with environmental sensors to report weather data, air quality, and precise positioning across the mesh. This turns repeater nodes into distributed sensor stations. Note that MeshCore's sensor/telemetry support is more limited and more firmware/build-dependent than Meshtastic's telemetry module - check the MeshCore repository for the current list of supported sensors before planning a deployment.

Supported sensor types

Sensor	Measurements	Interface	Cost	Notes
BME280	Temperature, humidity, barometric pressure	I ² C or SPI	\$3 - 8	Most common; accurate; fast response
BME680	Temperature, humidity, pressure, VOC/air quality index	I ² C or SPI	\$8 - 15	Adds air quality score; best for environmental monitoring
SHT31	Temperature, humidity	I ² C	\$5 - 10	Higher humidity accuracy than BME280 (Sensirion SHT31 $\approx \pm 2\%$ RH vs Bosch BME280 $\approx \pm 3\%$ RH); no pressure
GPS modules	Latitude, longitude, altitude, speed, heading	UART	\$5 - 25	See GPS module comparison below
INA219 / INA260	Voltage, current, power draw	I ² C	\$2 - 5	Battery and solar monitoring; useful for remote health checks

GPS module comparison

Current-draw and time-to-first-fix figures below are approximate; confirm against each module's datasheet (u-blox M8/M10, Quectel L76K, Unicore UC6580) for your specific board and configuration. Which GPS chip a given board ships with varies by revision.

Module	Constellations	Current draw	Cold fix time	Notes
u-blox M8N	GPS, GLONASS, BeiDou	~25 mA	~30 s	Common in T-Beam; good urban performance (note: T-Echo commonly ships the Quectel L76K, not the M8N)
u-blox M10	GPS, GLONASS, BeiDou, Galileo	~15 mA	~25 s	Newer generation; lower power, better accuracy. The MAX-M10S below is a specific M10-family module
u-blox MAX-M10S	GPS, GLONASS, BeiDou, Galileo	~5 mA	~25 s	Ultra-low power; T-Deck Plus; best for battery nodes
Quectel L76K	GPS, GLONASS, BeiDou	~20 mA	~35 s	Budget option; lower sensitivity than u-blox
UC6580	GPS, GLONASS, BeiDou, Galileo, QZSS, NavIC (L1+L5)	~30 mA	~20 s	Dual-band L1+L5; high precision. Verify the exact Heltec board name it ships on against current Heltec listings

Note: GPS draws significant power (~15 - 30 mA active). For repeater deployments where position is static, configure the node to fix position at startup and then disable GPS polling to save power.

Connecting a BME280 to common boards

Most MeshCore-compatible boards expose I²C headers. The pin assignments below are indicative only - **always check your specific board's official pinout** before wiring. I²C pins are remappable in firmware on many boards, the firmware default may differ from the values shown, and you must confirm each **VCC pin is 3.3 V** (not a 5 V pin) to avoid damaging the sensor:

BME280 pin	RAK4631 / WisBlock	Heltec V4 / V3	T-Echo
------------	--------------------	----------------	--------

VCC	3.3V (verify pin)	3.3V (verify pin)	3.3V (verify pin)
GND	GND	GND (verify pin)	GND
SDA	SDA (check board pinout)	SDA (check board pinout)	SDA (check board pinout)
SCL	SCL (check board pinout)	SCL (check board pinout)	SCL (check board pinout)
SDO	GND (I ² C addr 0x76)	GND	GND
CSB	3.3V (I ² C mode)	3.3V	3.3V

Enabling telemetry in MeshCore firmware

Sensor support is compiled into the firmware for supported boards. For custom sensor connections, build flags similar to the following are used - but **verify the exact build-flag names against the current MeshCore source (platformio.ini and the sensor driver code)**, as the names below are illustrative and may differ between firmware versions:

```
# In platformio.ini build flags, add a sensor definition
# (verify exact flag names in the MeshCore source):
build_flags =
  -DHAS_BME280=1
  -DBME280_I2C_ADDR=0x76

# Then rebuild and flash (see the PlatformIO CLI docs):
pio run -e your_target --target upload
```

As of 2026-06-08, pre-built firmware with BME280 support may be offered for some boards such as the RAK4631 WisBlock (which has a dedicated sensor slot) and select Heltec builds. Availability of pre-built sensor variants changes over time - check the MeshCore firmware releases/flasher page for the current list rather than assuming a given board has a pre-built image.

Telemetry data in the mesh

Sensor data is included in periodic telemetry packets broadcast by the node. Other nodes and apps that receive these packets display:

- Temperature and humidity at the repeater's physical location
- Barometric pressure trend (useful for weather monitoring)

- Air quality index (BME680 only)
- Battery voltage and solar input (if INA219 connected)

As an illustrative example, a local mesh community might operate several telemetry-equipped repeaters on hilltops and tower sites, providing real-time weather data for the surrounding area via the mesh network. (This is a hypothetical use case, not a documented specific deployment.)

Network monitoring with MQTT

For operators managing multiple nodes, MeshCore can be bridged to MQTT for centralized monitoring:

- Run a MeshCore Room Server (running on dedicated nRF52840 or ESP32 hardware)
- Traffic the Room Server is configured to bridge (such as messages, telemetry, and node advertisements it receives) is forwarded to MQTT topics - confirm exactly what the bridge forwards against the MeshCore repository, as not all mesh traffic is necessarily relayed
- Visualize with Grafana + InfluxDB for historical trending
- Alert on node disappearance (node stops advertising = possible power failure)

See the *Room Servers & Gateways* section for MQTT bridge setup details.

Link Budget & Propagation

Link Budget Calculations

A link budget calculation estimates whether a radio path between two nodes will work reliably before you deploy hardware. It's the single most useful tool for avoiding wasted installation trips and surprised failures.

The link budget equation

$$\text{Received Power (dBm)} = \text{TX Power (dBm)}$$

- + TX Antenna Gain (dBi)
- TX Cable Loss (dB)
- Free Space Path Loss (dB)
- Obstruction Loss (dB)
- + RX Antenna Gain (dBi)
- RX Cable Loss (dB)

$$\text{Link Margin (dB)} = \text{Received Power (dBm)} - \text{Receiver Sensitivity (dBm)}$$

A positive link margin means the link should work. As a common rule of thumb, a margin of 10 dB or more is treated as reliable (longer or mission-critical links often target a 15 - 20 dB fade margin). The author's recommended minimum is about 3 dB - below that a link is borderline and not recommended for permanent infrastructure. These are engineering conventions, not hard standards.

Key values for LoRa at 915 MHz

Receiver sensitivity by MeshCore preset

Figures below are SX1262 datasheet values with Rx Boosted gain (standard gain is roughly 3 - 4 dB worse). They assume the bandwidth listed for each preset.

Preset equivalent (SF / BW)	Receiver sensitivity
USA/Canada (SF7 / 62.5 kHz)	~-125 dBm

Preset equivalent (SF / BW)	Receiver sensitivity
Long Fast (SF11 / 250 kHz)	~-131 dBm
Long Slow (SF12 / 125 kHz)	~-137 dBm
Medium Slow (SF10 / 250 kHz)	~-129 dBm

Lower sensitivity number = can receive weaker signals = more range potential. Long Slow gives the best sensitivity but at the cost of extremely low data rate. (Sensitivity figures near -141/-148 dBm only occur at very narrow bandwidths around 10.4 kHz, not at the 125 - 250 kHz bandwidths these presets use.)

Free Space Path Loss at 915 MHz

Practical form (distance in km, frequency in GHz): $FSPL\ (dB) = 20 \times \log_{10}(d_km) + 20 \times \log_{10}(f_GHz) + 92.45$. The 92.45 constant already folds in the $4\pi/c$ term for kilometres and gigahertz.

In practical terms for 915 MHz:

Distance	Free Space Path Loss
1 km (0.62 mi)	91.6 dB
5 km (3.1 mi)	105.6 dB
10 km (6.2 mi)	111.6 dB
20 km (12.4 mi)	117.6 dB
50 km (31 mi)	125.6 dB

Note: Free space path loss assumes clear line of sight with no obstructions. Real-world losses are always higher.

Worked example: Rooftop repeater to ground-level node

Scenario: 5 km path, rooftop repeater at 30m height, portable node at 2m height.

Parameter	Value
TX Power (repeater)	27 dBm (requires external PA - see note)
TX Antenna Gain	+5 dBi
TX Cable Loss (1m LMR-200)	-0.4 dB

Parameter	Value
Free Space Path Loss (5 km, 915 MHz)	−105.6 dB
Obstruction/Fresnel loss estimate	−10 dB (mixed urban)
RX Antenna Gain (portable node, 2 dBi)	+2 dBi
RX Cable Loss (none for portable)	0 dB
Received Power	27 + 5 − 0.4 − 105.6 − 10 + 2 = −82.0 dBm
Receiver Sensitivity (USA/Canada SF7)	−125 dBm
Link Margin	−82.0 − (−125) = +43.0 dB

Power note: common LoRa modules built on the SX1262 top out at +22 dBm conducted, so a 27 dBm example implies an external power amplifier - state it explicitly when planning. FCC Part 15.247 (902 - 928 MHz) caps conducted power at 1 W (30 dBm) and derives a 36 dBm EIRP ceiling, and requires a 1 dB reduction in conducted power for every dB of antenna gain above 6 dBi. At the +5 dBi gain used here no reduction is required, but higher-gain antennas would force the conducted power down.

A 43 dB margin is very comfortable - this link will work reliably even with additional obstruction losses not captured in the estimate. (Because this example uses the SF7 sensitivity of −125 dBm, it is unaffected by the higher-SF sensitivity corrections above; a Long Slow / SF12 link at −137 dBm would have an even larger margin.)

Fresnel zone clearance

The Fresnel zone is an elliptical (football-shaped) region around the straight-line path between two antennas. Radio energy travels through this whole zone, not just the visual line, so obstacles near - not just directly on - the line still degrade the signal.

Even in "clear" line-of-sight paths, the first Fresnel zone must be about 60% clear of obstructions for reliable communication. The first Fresnel zone radius at the midpoint of a path:

$$r = 8.66 \times \sqrt{(d_{\text{km}} / f_{\text{GHz}})} \text{ meters}$$

Where d = path length in km, f = frequency in GHz

(This is the same as the form $r = 17.3 \times \sqrt{(d / (4 \cdot f))}$ used elsewhere:

$$17.3 / \sqrt{4} = 8.66.)$$

The radius scales with link length. For 915 MHz:

1 km path: $r \approx 9$ meters

10 km path: $r \approx 28.6$ meters

So an obstruction within ~28.6 m of the direct path midpoint will partially block the signal on a 10 km link, but on a 1 km link the relevant zone is only ~9 m. Use the formula for your actual path length rather than a fixed radius.

This is why hilltop-to-hilltop links work so well: the terrain clears the Fresnel zone naturally. For rooftop-to-rooftop links in cities, trees and building facades at path midpoints can add 10 - 20 dB of loss even when the antennas themselves have direct line of sight.

When to use a link budget

- Before installing a repeater at a new site, calculate whether it can reach your intended coverage area
- When planning a point-to-point relay link between two specific nodes
- When a deployed link is underperforming - work backwards from measured RSSI to identify where the losses are
- When comparing two candidate repeater sites - small differences in height can produce large differences in link budget

RF Propagation Planning Tools

Several free tools can help you model coverage and plan repeater placement before deploying hardware. Using these tools can save wasted trips and help you choose between candidate sites.

HeyWhatsThat (heywhatsthat.com)

The fastest tool for estimating radio horizon from a specific point.

- Enter a location (address, coordinates, or click on map)
- Set the antenna height
- Get a visualization of the radio horizon: which areas have line-of-sight from that point

How to use for repeater site selection:

1. Go to heywhatsthat.com
2. Click on your candidate repeater site on the map
3. Set height to your intended antenna mounting height
4. Click "Submit" and examine the color-coded visibility map
5. Compare multiple candidate sites by opening each in a new tab

Limitations: HeyWhatsThat uses SRTM elevation data, which is a Digital *Surface* Model, not bare earth - it partially captures building rooftops and tall vegetation (HeyWhatsThat's own documentation notes you may see "shadowy bumps and gaps" in cities because the elevation data includes rooftops). The real caveat is that SRTM is too coarse (~30 m) to model individual buildings or trees reliably, so treat the visibility output as terrain-dominated: actual coverage will still be lower than predicted in areas with tall buildings or dense forest that the coarse data cannot resolve.

Radio Mobile Online (radiomobileonline.pe1mew.nl)

More sophisticated link analysis tool with full path profile and link budget integration. The online version is hosted at radiomobileonline.pe1mew.nl.

- Enter transmitter and receiver coordinates, heights, antenna gain, TX power, and frequency
- Generates a path profile showing terrain elevation along the path
- Calculates predicted received signal level
- Shows Fresnel zone clearance along the path

Radio Mobile is best for detailed analysis of specific point-to-point paths, not area coverage visualization.

CloudRF (cloudrf.com)

Professional-grade coverage prediction with a free tier. Features:

- SRTM + LIDAR terrain data (more accurate than basic tools)
- Can include clutter data (buildings, vegetation) for urban environments
- Overlay predicted coverage on Google Maps or OpenStreetMap
- Free tier: limited calculations per month; paid plans for heavy use

Best for: precise coverage maps for presentations, permitting, or professional deployments.
Overkill for casual site selection.

Splat! (free, offline)

Open-source RF propagation tool that runs locally on Linux/Mac. Uses SRTM terrain data. **This is an advanced tool - it is not a copy-paste one-liner.** Before you can run it you must (1) create `.qth` site-location files describing each transmitter/receiver (location, height, name), (2) download SRTM elevation data and convert it into SPLAT's own `.sdf` format using the `srtm2sdf` utility (SPLAT! does not read raw SRTM directly), and (3) set the operating frequency and other path parameters in an `.lrp` file - SPLAT! has no `-f` frequency command-line option (in SPLAT! the `-f` flag relates to Fresnel-zone clearance, not center frequency).

```
# Install on Debian/Ubuntu (package availability varies by release;
# you may need to build from source per the SPLAT! project site)
sudo apt install splat

# Download SRTM data for your region from dwtkns.com/srtm or usgs.gov,
# then convert it to SPLAT's .sdf format:
srtm2sdf N40W106.hgt

# Create your tx.qth / rx.qth site files and a tx.lrp parameter file
# (frequency, ERP, etc. go in the .lrp file, NOT on the command line).
# Then generate a point-to-point path analysis:
splat -t tx.qth -r rx.qth -d /path/to/sdf/data
```

Splat! is overkill for most community mesh deployments but valuable when building professional-grade coverage documentation or integrating with GIS workflows. Note that the 0.5 W ERP figures used in such examples are not buildable on stock LoRa radios - the SX1262 chip maxes out at +22 dBm and an external PA would be required.

Practical planning workflow

For most community mesh network planning, this workflow is sufficient:

1. **Identify candidate sites**

Use topographic maps (USGS topo viewer, CalTopo) to identify hilltops, ridgelines, water towers, and tall buildings in your coverage area.

2. **Quick horizon check**

For each candidate site, run HeyWhatsThat at the proposed antenna height. Immediately discard sites with poor visibility to your target coverage area.

3. **Link budget for remaining candidates**

For the 2 - 3 best candidates, calculate link budgets to your target coverage area edges (farthest points). Compare margin values.

4. **On-site test before permanent install**

Set up a temporary antenna and node at the candidate site. Walk/drive your coverage area while monitoring RSSI/SNR. Real-world testing always beats prediction tools.

5. **Document the result**

Record the actual coverage performance after deployment. This data helps plan future expansion nodes.

Interpreting RSSI and SNR from field tests

The bands below are approximate field heuristics, not hard cutoffs - the actual decode limit depends on the modem preset (spreading factor). LoRa decodes below the noise floor: the relevant SNR threshold is the per-spreading-factor demodulator floor from the SX1262 datasheet - roughly -7.5 dB at SF7, -12.5 dB at SF9, -17.5 dB at SF11 (LongFast), and -20 dB at SF12 (Long Slow). A link that looks "weak" at SF7 may still decode reliably at SF11/SF12.

RSSI	SNR	Link quality assessment
> -90 dBm	$> +10$ dB	Excellent - reliable at all data rates
-90 to -105 dBm	$+5$ to $+10$ dB	Good - reliable for all normal use
-105 to -115 dBm	0 to $+5$ dB	Marginal - may see occasional packet loss
-115 to -120 dBm	-5 to 0 dB	Weak - intermittent; not suitable for infrastructure
< -120 dBm	< -5 dB	Very weak - expect frequent failure (though SF11/SF12 can still decode down to roughly $-17.5/-20$ dB SNR)

Key insight: In the noise-limited regime, what determines whether a packet decodes is whether the SNR exceeds LoRa's per-spreading-factor decode floor (e.g. -17.5 dB at SF11) - RSSI alone does not. Both metrics matter, though: when the signal is well above the noise floor, absolute received power (RSSI) becomes the limiting factor. A signal at -125 dBm with $+5$ dB SNR is more reliable than one at -100 dBm with -5 dB SNR (which is being swamped by noise). When diagnosing a noise-limited marginal link, prioritize improving SNR; when the link is power-limited, work on RSSI.

RF Propagation Environments

How 915 MHz RF behaves across urban, forested, mountainous, and coastal environments.

Urban Propagation

Urban Propagation at 915 MHz

Dense urban environments present some of the most complex RF propagation conditions for mesh networks. Understanding these effects helps planners place nodes effectively and set realistic range expectations.

Street Canyon Effect

Buildings create RF corridors along streets and dead zones perpendicular to them. A node at street level may have excellent range in one direction and very poor range 90 degrees away. This directional bias is caused by reflections off building facades channeling energy down the street while absorbing or blocking energy crossing between streets.

When planning urban coverage, orient your mental model around street grids — as a rough illustration, a single node may cover several blocks along a street axis but only one or two blocks across it. These block counts are illustrative of the canyon anisotropy, not measured figures; actual coverage depends on building height, street width, and node placement.

Building Penetration Losses at 915 MHz

Material	Typical Loss
Concrete / brick exterior walls	10 - 20 dB
Interior walls (drywall)	3 - 5 dB
Windows (plain glass)	2 - 5 dB

These figures are consistent with commonly cited material-attenuation tables (e.g. ITU-R P.2040 and vendor RF references). Note an important exception: **low-emissivity (low-E) or metal-coated glass**, common in modern energy-efficient buildings, attenuates far more than plain glass — often **10-30+ dB** — because the metallic coating reflects RF. Don't assume the 2-5 dB plain-glass figure for coated windows.

Elevator shafts and stairwells can sometimes act as opportunistic waveguides, occasionally propagating signal across multiple floors or between building sections — though this is anecdotal

and not reliable. In practice, metal-lined elevator shafts are more often RF-blocking than waveguiding. Where it does occur, it can be exploited (placing a node near a stairwell to reach upper floors) or can cause unexpected interference between nodes, but do not count on it in a link budget.

Rooftop Advantage

Nodes above the building line communicate freely across the urban environment. The **roofline is the critical breakpoint** — it acts as a diffraction edge, so crossing from just below to just above the parapet wall can change a link dramatically. The figure of "1 meter below vs 1 meter above the parapet" is illustrative of how sensitive this diffraction edge is, not a precise threshold.

Getting above the building line converts an urban node from a neighborhood-scale device into a metro-scale relay. A single well-placed rooftop node can serve an entire neighborhood that would otherwise require dozens of street-level nodes.

Reflections and Multipath

Urban environments create multipath propagation — signals arrive at the receiver via multiple reflected paths with different time delays and phase offsets. This causes constructive and destructive interference that varies by location, creating "dead spots" (fade nulls) spaced roughly every half wavelength — about **16 cm (~6 inches) at 915 MHz**, not a few feet. Because the nulls recur on this fine spatial scale, moving a node even a few inches can move it into or out of a null, and this can degrade narrow-band radio performance.

LoRa's chirp spread spectrum handles multipath well compared to conventional narrowband radios. The chirp encoding is largely immune to moderate multipath delay spread, making LoRa a good choice for urban mesh deployment where multipath is unavoidable.

Underground Infrastructure

Subway stations, underground parking garages, and utility tunnels are essentially RF-opaque at 915 MHz. Plan for coverage gaps in underground infrastructure — nodes above ground do not penetrate reliably. Underground coverage requires dedicated nodes installed within the underground space itself.

Practical Urban Planning Guidelines

1. **Establish a rooftop backbone first.** Prioritize 3 - 5 high-rise or rooftop nodes to establish a "backbone" visible across the metro area. These nodes handle the long-haul mesh connectivity.

2. **Fill in with lower nodes as the network grows.** Street-level and mid-rise nodes fill gaps in the backbone coverage for pedestrian and in-building use.
3. **One well-placed rooftop node can serve an entire neighborhood.** Resist the urge to densely deploy at street level before establishing rooftop coverage - the rooftop node will outperform 10 street nodes.
4. **Account for building penetration in link budgets.** If a link must pass through walls, add the appropriate dB loss to your budget before assuming the link will work — and use the higher coated-glass figure if the building has low-E windows.

Forest and Vegetation Propagation

Forest and Vegetation Propagation at 915 MHz

Vegetation is one of the most significant impairments to 915 MHz propagation. Forested terrain requires a fundamentally different planning approach from open or urban environments.

Foliage Attenuation

915 MHz is significantly absorbed by vegetation. Specific attenuation through dense in-leaf woodland near 900 MHz is roughly **0.2 - 0.5 dB per meter** (ITU-R P.833 / Weissberger), i.e. about **20 - 50 dB per 100 m** of traversal. Importantly, the loss does *not* grow linearly without bound - it saturates beyond roughly 14 m of foliage depth, so very deep canopy adds less than a simple per-meter multiplication would predict. Typical losses:

Vegetation Type	Loss per 100 m of Traversal
Dense deciduous forest (summer, full leaf)	~20 - 50 dB
Coniferous forest (pine, fir)	Comparable - needle vs. broadleaf differences at 915 MHz are not well-characterized and depend on density and moisture; do not assume conifers are reliably lower (year-round dense, high-moisture canopy can be similar or higher)

Even a few hundred meters of dense forest can consume the entire link margin of a typical LoRa deployment. As a result, ground-level range in dense forest is often only 200 - 500 meters even with the longest LoRa spreading factors - this figure is derived from the loss budget and consistent with field reports, not a fixed measured constant, and actual range varies with antenna height, modem preset, and forest density.

Seasonal Variation

Deciduous forests have dramatically different propagation in summer (full leaf) versus winter (bare branches). A link that works reliably in December may fail completely in July when the leaves are out.

Always plan coverage for worst-case summer leafed-out conditions. Links that are marginal in winter will likely fail in summer. If your network must work year-round, design for July.

Elevation Above Canopy

The single most effective technique for improving range in forested terrain is getting the antenna above the tree canopy.

A node mounted at canopy level or just above it has near-line-of-sight to distant nodes that are also above the trees. Even clearing the canopy by roughly 5 - 10 meters dramatically improves range in forested areas (an approximate guideline - the benefit depends on canopy height and link geometry).

Node Position	Typical Range (derived / site-dependent)
Ground level in dense forest	~200 - 500 m (typical, derived from loss budget)
At or above canopy (~20 m elevation)	~5 - 10 km to other elevated nodes (typical example; clear-LOS LoRa links can far exceed this, dense conditions fall short)

This is a dramatic difference - illustrative of the order-of-magnitude benefit, the same hardware can perform roughly 10 - 20× better simply by being above the canopy. The exact multiplier is an illustration, not a precise measured figure.

Trail Corridor Effect

Trails create linear openings in the forest canopy. Range along a trail is significantly better than off-trail in the same forest. The open sky corridor above the trail allows near-LOS propagation along the trail axis.

This is useful for planning hiking or trail mesh coverage - nodes near trail intersections or high points along trails will have better coverage than nodes placed arbitrarily in the forest interior.

Mixed Terrain Path Budgets

When a link crosses both open and forested terrain, **plan for the worst-case segment.** A 10 km link that crosses 3 km of dense forest needs to be designed for the forest loss, not the open segments.

Use a simple approach: calculate the total forest path length in your link, apply a conservative **~40 - 50 dB/100m** for dense summer deciduous canopy (the high end of the 20 - 50 dB/100m range is the conservative worst-case planning value; note the loss saturates beyond ~14 m depth), and verify the total loss fits within your link budget. If it doesn't, raise antenna height or use a higher-gain antenna. (Transmit power is already at the radio's ~22 dBm chip maximum on standard mesh hardware; FCC Part 15.247 caps conducted power at 30 dBm and EIRP at 36 dBm, and antennas above 6 dBi require a dB-for-dB power reduction - so there is little legal headroom to simply "turn up the power.")

Summary for Forest Deployments

- Mount antennas as high as practical - at or above canopy height
- Test and plan for summer worst-case conditions
- Use trail corridors for coverage where possible
- Account for every meter of forest in your path budget
- Consider tower or tall-tree mounting for backbone nodes

Mountain and Complex Terrain

Mountain and Complex Terrain Propagation

Mountain and highly variable terrain introduces propagation challenges - and opportunities - that differ fundamentally from flat-land or urban planning. Terrain masking is the dominant factor, but ridge placement can turn a liability into an asset.

Terrain Masking

Terrain masking is the most significant propagation factor in mountains. When terrain lies between the transmitter and receiver, path loss increases dramatically - **diffraction over ridges adds 10 - 30+ dB** compared to free-space loss at the same distance.

Before planning a mountain link, verify line-of-sight using a terrain profile tool: HeyWhatsThat or Radio Mobile (web-based, easy to use) - or SPLAT! for advanced users comfortable with the Linux/macOS command line and converting SRTM terrain data to its own file formats. If the path crosses terrain, budget for the additional diffraction loss.

Knife-Edge Diffraction

When a signal diffracts over a sharp ridge, it bends into the shadow zone on the far side. This diffraction is calculable using **Fresnel zone analysis** - the geometry of the ridge height relative to the first Fresnel zone determines how much loss (or occasionally gain) results.

Tools like Radio Mobile (based on the Longley-Rice ITM model) estimate knife-edge diffraction loss, though accuracy degrades in mountainous terrain with multiple obstacles - ITM uses a single equivalent rounded-obstacle approximation and is known to underestimate field strength in long-distance, multi-obstacle fringe cases. A sharp, isolated ridge causes less diffraction loss than a broad rounded hill, which blocks a larger portion of the Fresnel zone.

Ridge-Mounted Repeaters

A repeater placed on a ridgeline can reach **both** the illuminated side and the shadow side of the ridge. Coverage is not symmetric, however: the illuminated side is served by direct line-of-sight, while the shadow side is reached only via diffraction, with significantly reduced range and reliability that worsens as the shadowing depth increases.

This is the key insight for mountain mesh design: **place repeaters on ridges, not in valleys**. A valley node can communicate well within the valley, but a ridgeline node covers its own valley and can reach into adjacent valleys where line-of-sight exists, with diffraction-limited reach where terrain intervenes - still multiplying usable coverage substantially per node compared to a valley-floor site.

Valley Isolation

Nodes in valleys can communicate well within the valley but are effectively isolated from nodes in adjacent valleys without a ridge or mountain repeater bridging them. This creates natural "valley clusters" in mountain mesh networks - each valley segment is connected internally but disconnected from neighbors unless ridge nodes exist.

Planning a mountain mesh means identifying which valleys need coverage, then finding the ridgelines that can serve multiple valleys with a single node.

Elevation vs. Range (Rule of Thumb)

For an antenna at height h meters above local terrain, the refraction-corrected (4/3-earth) radio horizon is:

“ **Radio horizon distance $\approx 4.12 \times \sqrt{h}$ km**

Height Above Valley Floor	Radio Horizon
10 m	~13 km
50 m	~29 km
200 m	~58 km
500 m	~92 km

A repeater at 200 m above the valley floor sees a radio horizon of approximately 58 km. **But radio horizon is the geometric ceiling, not guaranteed coverage.** Valleys, terrain masking, and the

link budget to low, handheld nodes in the valleys will reduce real coverage well below 58 km. Use this figure only to identify candidate sites, then confirm with terrain analysis and field testing before assuming a single node blankets a region.

Emergency Drone / Hilltop Mesh Extension

Some communities deploy temporary mesh nodes on hilltops via drones during emergencies to bridge isolated valleys. The same capability works for large outdoor events in terrain-constrained areas. A drone-carried LoRa node at 100 - 200 m AGL can bridge two otherwise-isolated valleys for the duration of its battery, providing emergency communications coverage without requiring permanent infrastructure.

Portable battery-powered repeater kits carried to hilltops on foot are another practical approach for planned events or disaster response.

Snow Effects on Antennas

Snow has low RF absorption at 915 MHz and does not significantly affect propagation. However, **heavy wet snow buildup on antennas can detune them:**

- Use vertically-polarized antennas with a drip point design to shed snow
- Avoid horizontal radial elements in heavy-snow environments
- Wet snow accumulation on a horizontal radial can shift the resonant frequency noticeably under heavy wet-snow loading, reducing antenna efficiency and degrading SWR

For high-altitude winter deployments, sleeved dipoles and verticals with a tapered/drip-capable profile outperform flat-panel or horizontal-element designs.

Water and Coastal Propagation

Water and Coastal Propagation

Water surfaces create some of the most favorable RF propagation conditions at 915 MHz. Coastal and over-water deployments can achieve ranges that far exceed typical terrestrial links.

Over-Water Propagation

At low grazing angles, calm water reflects RF efficiently, behaving somewhat like a "mirror." But the long ranges seen over water are not driven by reflection alone: low-loss line-of-sight, specular reflection at low grazing angles, and - frequently the dominant factor - **evaporation ducting** (a refraction mechanism in which a near-surface humidity gradient bends signals beyond the geometric horizon) all contribute. A higher sea state roughens the surface, scatters the reflected ray, and degrades the "mirror" behavior. Compared with land (rough, absorptive, and cluttered), open water still provides a much better reflective ground plane and an effective radio horizon that extends further than over land.

The Two-Ray Model Over Water

Over a flat, reflective surface like open water, two signal paths exist between transmitter and receiver:

1. The **direct ray** traveling line-of-sight
2. The **ground-reflected ray** bouncing off the water surface

These two rays interfere constructively or destructively depending on antenna heights and distance. The transition point is the two-ray breakpoint distance, $d \approx 4 \cdot h_1 \cdot h_2 / \lambda$. For two nodes at 5 m height over water at 915 MHz ($\lambda \approx 0.328$ m), the breakpoint is roughly **300 m** - not 1 - 2 km. *Below* the breakpoint the received field oscillates through a pattern of nulls and peaks; *beyond* the breakpoint, received power falls off steeply and monotonically (roughly d^{-4} , about 40 dB/decade), with no further oscillation. The long over-water ranges operators report are sustained mainly by clear line-of-sight and evaporation ducting, not by the two-ray interference pattern.

Raising antenna height pushes the two-ray breakpoint to greater distances and generally improves over-water performance.

Documented Long-Range Links Over Water

Some LoRa operators report **50 - 80 km links** across large lakes or bays with elevated antennas, but these are best-case results - they depend on antenna height, calm water, clear line-of-sight, and atmospheric ducting, and are not a routine or guaranteed outcome. Do not plan a network assuming them; verify any over-water link by field test. Reported and documented figures include:

- Favorable over-water conditions have produced 50 - 80 km links with 5 - 8 dBi antennas at 10 - 20 m height, but results vary widely and depend on sea state, ducting, and clear LOS
- As of 2024, the LoRa/LoRaWAN distance record is **1,336 km** (ground-based trackers on a boat/buoys near Sesimbra, Portugal reaching a Canary Islands gateway - not balloon-assisted), surpassing the earlier 832 km balloon record. Treat all such records as exceptional best-case LOS, not representative range
- Over-water links of **30 - 50 km** with standard hardware and good antenna heights (10+ m) are achievable in favorable over-water conditions with clear line-of-sight, but results are not guaranteed

These distances are not achievable over land with equivalent hardware and height - the over-water propagation advantage is real and significant, but the figures above are best-case ceilings, not dependable planning numbers.

Coastal Network Planning

Islands, peninsulas, and coastal communities benefit greatly from over-water propagation. A node on a bluff or sea cliff can cover:

- Coastal marine traffic (boats, kayaks, vessels with LoRa-equipped trackers)
- Island communities at ranges exceeding typical land deployments
- Adjacent coastal nodes along the shoreline

For coastal networks, prioritize elevated nodes on headlands, bluffs, and sea cliffs. Even modest elevation (10 - 20 m above sea level on a coastal promontory) provides excellent coverage over water. As a horizon reference, a 10 - 20 m height gives roughly a 30 km line-of-sight radio horizon over water - present longer ranges as best-case, not typical.

Marine Environment Hardware

Considerations

Coastal humidity and salt air accelerate corrosion of connectors, coax, and metal mounting hardware. Coastal deployments require additional weatherproofing measures compared to inland installations:

- Use **marine-grade stainless steel** hardware (316 SS or better) for all mounting
- Apply **NO-OX-ID A-Special** or equivalent anti-oxidant compound to all coax connectors
- Inspect weatherproofing tape, heat shrink, and connector boots **more frequently** than inland sites - annually at minimum, semi-annually in high-exposure locations
- Use sealed junction boxes (IP67 or better) for any exposed connections
- Consider conformal coating for PCBs in equipment enclosures near the waterline

Troposcatter

At longer distances over water, tropospheric scatter occasionally enables beyond-horizon propagation. Troposcatter occurs when RF energy scatters off irregularities in the troposphere and some portion reaches a beyond-horizon receiver.

Troposcatter is rare, unpredictable, and unsuitable as a network planning basis - you cannot count on it being available when needed. However, it explains occasional unexpectedly long contact distances reported by LoRa operators over open ocean or large lakes. If you observe an anomalously long contact, troposcatter (or evaporation ducting under temperature inversion layers) is the likely explanation.

Node Placement Strategy

Principles and methods for selecting repeater sites that maximise coverage and network reliability.

Repeater Placement Principles

The Three Rules of Repeater Placement

Every successful LoRa mesh deployment rests on three placement rules. Violate any one of them and the network will under-perform regardless of hardware quality, antenna gain, or software tuning.

Rule 1 - Height Above Terrain

LoRa operates in UHF spectrum where propagation is quasi-optical. The band depends on region: 902 - 928 MHz in the United States, 863 - 870 MHz in the EU, and 433 MHz in some regions. In the US, use 902 - 928 MHz — 868 MHz is the EU ISM band and is not available for Part 15 LoRa here, and 433 MHz US operation is tightly restricted under Part 15. (Note: 433/868/915 MHz are all UHF; SHF begins at 3 GHz.) The radio horizon - not the visual horizon - sets the hard ceiling on range. Antenna elevation is the single most powerful lever available to a network planner.

Doubling antenna height does *not* double range; it multiplies it by roughly $\sqrt{2}$ ($\sim 1.41\times$). Using the standard refraction-corrected (4/3-earth) radio horizon formula **$d(\text{km}) \approx 4.12 \times \sqrt{h \text{ in metres}}$** , a node at 2 m AGL (above ground level) has a radio horizon of about 5.8 km on flat earth. Raise it to 20 m and the horizon extends to ~ 18 km. Raise it to 50 m and you reach ~ 29 km. Every metre of height is worth more at low elevations than at high ones - the return diminishes as you climb. (The purely geometric/optical horizon uses the smaller coefficient $3.57 \times \sqrt{h}$; the 4.12 figure includes normal atmospheric refraction.)

Practical target: For a community repeater, aim for a minimum of 10 - 15 m AGL. A rooftop mount, water tower, or hilltop installation should be the first option explored.

Rule 2 - Line-of-Sight to the Coverage Area

A repeater serves areas it can "see" far better than areas it cannot. Because LoRa uses relatively low frequencies compared to Wi-Fi, it can diffract around moderate terrain features - but non-line-of-sight coverage via diffraction is weak and unreliable, and penetration through significant ridgelines or dense urban canyons is limited. The Fresnel Zone (a football-shaped region around the direct path) must be at least 60% clear for reliable propagation; 100% clearance of the first Fresnel zone is the gold standard.

At 915 MHz the first Fresnel zone radius at the midpoint of a 10 km path is approximately:

$$\begin{aligned} r_1 &= 17.32 \times \sqrt{d / (4f)} \quad [\text{equivalently } 8.66 \times \sqrt{d / f}] \\ &= 17.32 \times \sqrt{10 / (4 \times 0.915)} \\ &\approx 17.32 \times \sqrt{2.73} \\ &\approx 28.6 \text{ m} \\ &(\text{d in km, f in GHz}) \end{aligned}$$

By the 60% rule, an obstacle begins to impair the link once it intrudes more than ~40% into the first Fresnel zone - i.e., once it rises into roughly the central ~17 m of the ~29 m radius at midpoint (about 11 - 12 m of intrusion). Full clearance of all ~29 m is the gold standard. Use terrain analysis tools (HeyWhatsThat, SPLAT!, CloudRF, or the terrain profile view in Google Earth) to verify clearance *before* climbing a tower.

Rule 3 - Overlapping Coverage with Adjacent Repeaters

A mesh node at the edge of one repeater's coverage zone must be within range of at least one other repeater. This overlap provides two things: redundancy if one repeater fails, and smooth routing as a mobile node moves between coverage zones. As a planning rule of thumb, target roughly 20 - 30% geographic overlap between adjacent repeater footprints. These are illustrative heuristics rather than measured thresholds: too little overlap (very roughly under ~10%) tends to leave dead corridors, while very heavy overlap (over ~50%) adds little extra reliability for the extra sites.

The Minimum Viable Repeater Principle

One well-placed hilltop or rooftop repeater at 30 m AGL can cover 300 - 800 km² of flat to rolling terrain. Ten ground-level nodes scattered across the same area will collectively cover far less, generate more channel congestion, and create a fragile, multi-hop routing mess.

This principle is counterintuitive for operators coming from Wi-Fi, where more access points always help. In LoRa mesh, channel time is shared by all nodes. Each additional ground-level node adds noise and contention while contributing minimal coverage.

“ **Rule of thumb:** Before deploying any new ground-level node, ask whether a single elevated repeater could solve the coverage problem instead. In almost every case, it can.

Dead Zone Identification Methods

Dead zones are coverage gaps where a mesh client cannot hear any repeater. Identifying them systematically before deployment prevents costly re-work later.

Method 1 - Terrain Masking Analysis

Load a digital elevation model (DEM) of your target area into a tool such as SPLAT!, Radio Mobile, or CloudRF. Run a viewshed analysis from each proposed repeater site. Areas outside all viewsheds are your primary dead zones. This takes minutes and costs nothing.

Method 2 - Wardrive / Walk Survey

Deploy a mobile node running MeshMapper or a GPS-tagged Meshtastic client. Drive or walk the target area while logging signal reports (RSSI and SNR) from known repeaters. Overlay the log on a map. Holes in the RSSI heat-map are dead zones. This method catches dead zones that terrain analysis misses (e.g., urban canyons, underground parking, indoor spaces).

Method 3 - Predictive Modelling with Terrain Obstruction Factor

After estimating coverage radius by terrain type (see the Coverage Radius Estimation page), draw coverage circles around each repeater. Grid cells covered by zero circles are dead zones. Adjust for known terrain features: a ridge intruding the line-of-sight path adds diffraction loss that depends on the ITU-R P.526 diffraction parameter (the ridge's height relative to the direct line, its position along the path, and frequency), so model it per-path rather than applying a fixed percentage. A ridge near one endpoint, at the midpoint, or barely grazing the Fresnel zone all produce very different losses.

Method 4 - Community Spot Reporting

For an existing network, a simple form where users report "no connectivity" locations aggregated on a shared map is surprisingly effective. Each report is a data point; a cluster of reports identifies a dead zone. The disadvantage is latency - it only works after deployment.

Using Terrain Analysis to Find Optimal Sites Before Going Outside

The cheapest field test is the one you never have to make. Modern terrain analysis tools allow a planner to shortlist candidate sites from a desk and eliminate poor candidates before anyone puts boots on the ground.

1. **Download a DEM.** USGS 1/3 arc-second DEMs (10 m resolution) are freely available for the contiguous USA via The National Map. Import into QGIS, SPLAT!, or similar.
2. **Identify candidate high points.** Use the DEM hillshade layer to find hilltops, ridgelines, and tall structures within your target area. Water towers and tall buildings are often not in the DEM - add them manually.
3. **Run viewsheds.** For each candidate, run a viewshed at your estimated antenna height (e.g., 15 m AGL). Note the percentage of target area covered.
4. **Rank candidates.** Sort by coverage area. A single site covering >80% of target area is a strong anchor candidate.
5. **Check access and permission.** The best RF site is worthless if you cannot install equipment there. Check ownership, zoning, and access before committing to a site.
6. **Field-verify top candidates.** Visit only the top 2 - 3 sites. Confirm elevation, access, power availability, and backhaul options (if applicable).

This workflow can substantially reduce the number of field visits required compared to ad-hoc site hunting, and produces better placements because decisions are based on quantitative terrain data rather than intuition.

Coverage Radius Estimation by Terrain Type

The Radio Horizon Formula

The theoretical radio horizon for a single antenna at height h metres above a smooth spherical earth, accounting for standard atmospheric refraction (the 4/3-earth model), is:

$$d \text{ (km)} = 4.12 \times \sqrt{h_m} \quad (\text{radio horizon, 4/3-earth refraction})$$

For a link between two antennas at heights h_1 and h_2 the total radio horizon is:

$$d_{total} \text{ (km)} = 4.12 \times (\sqrt{h_1} + \sqrt{h_2})$$

Note: the purely geometric (optical) line-of-sight horizon uses the coefficient $3.57 \times \sqrt{h}$. The 4.12 coefficient adds the standard 4/3-earth refraction correction and is the value used for *radio* horizon throughout this book (consistent with the mountain-and-complex-terrain and repeater-placement pages). There is no latitude term in this formula.

This is the *maximum possible* range on a flat, unobstructed earth - a theoretical ceiling, not a usable-coverage promise. Real terrain, vegetation, and buildings reduce this significantly. The formula provides the ceiling; terrain obstruction factors bring it down to a realistic estimate.

Terrain Obstruction Factors

Apply these multipliers to the radio-horizon distance to get a realistic coverage radius. The factor bands below are *illustrative planning heuristics*, not measured constants - no Tier-1 standard assigns these exact multipliers to named terrain classes. The underlying propagation behaviour is described in ITU-R P.526 (diffraction), P.833 (vegetation), and P.1411 (urban/suburban short-range); use those for rigorous modelling.

Environment Type	Obstruction Factor	Effective Coverage (%)	Notes
------------------	--------------------	------------------------	-------

Flat open (farmland, desert, water)	0.85 - 1.00	85 - 100%	Near-theoretical range; ground reflections can help at low angles
Gentle rolling terrain	0.65 - 0.80	65 - 80%	Moderate ridge shadowing; elevated repeaters mitigate well
Suburban (low-rise, gardens)	0.50 - 0.65	50 - 65%	Houses and trees add 5 - 15 dB of excess loss (ITU-R P.1411 range)
Dense forest / jungle	0.35 - 0.55	35 - 55%	Vegetation loss ~0.2 - 0.5 dB/m through canopy at 915 MHz (ITU-R P.833 / Weissberger); saturates beyond ~14 m depth
Urban (mid-rise, 3 - 8 floors)	0.30 - 0.50	30 - 50%	Building diffraction dominates; rooftop-to-rooftop paths much better
Dense urban (high-rise, canyons)	0.15 - 0.30	15 - 30%	Multipath + shadowing severe; per-block coverage planning needed
Mountainous / complex terrain	0.20 - 0.60	20 - 60%	Highly variable; valleys may have near-zero coverage from a single site

Worked Examples

The coverage *areas* below are theoretical maxima derived from the radio-horizon ceiling times an obstruction factor. They are **not** the usable coverage you should plan around. In practice a single well-placed repeater at ~30 m AGL delivers on the order of **300 - 800 km²** of reliable coverage in flat/rolling terrain, and far less in forest or urban settings - see the repeater-placement-principles and repeater-grid pages for realistic per-terrain planning densities. Treat the figures here as "maximum, not expected."

Example 1 - 30 m Tower on Flat Land

A community group installs a repeater at the top of a 30 m self-supporting tower in flat agricultural land. The mobile clients they serve have antennas at 1.5 m AGL.

Radio horizon (tower): $4.12 \times \sqrt{30} = 4.12 \times 5.48 = 22.6 \text{ km}$
Radio horizon (client): $4.12 \times \sqrt{1.5} = 4.12 \times 1.22 = 5.0 \text{ km}$
Total radio horizon: $22.6 + 5.0 = 27.6 \text{ km}$

Obstruction factor (flat open): 0.90

Theoretical coverage radius: $27.6 \times 0.90 \approx 24.8$ km

Theoretical coverage area (ceiling): $\pi \times 24.8^2 \approx 1,930$ km²

That $\sim 1,930$ km² is the radio-horizon ceiling, not a usable-coverage guarantee. Realistic usable coverage from a single well-sited rural tower is far lower - typically on the order of 300 - 800 km² once real-world fading, foliage, and link-margin requirements are accounted for. Plan around the usable figure, not the ceiling.

Example 2 - 10 m Mast in Suburbs

A volunteer mounts a repeater on a 10 m mast attached to their house in a typical American suburb (one-storey houses, trees). Mobile clients at 1.5 m.

Radio horizon (mast): $4.12 \times \sqrt{10} = 4.12 \times 3.16 = 13.0$ km

Radio horizon (client): $4.12 \times \sqrt{1.5} = 4.12 \times 1.22 = 5.0$ km

Total radio horizon: $13.0 + 5.0 = 18.0$ km

Obstruction factor (suburban): 0.55

Theoretical coverage radius: $18.0 \times 0.55 \approx 9.9$ km

Theoretical coverage area (ceiling): $\pi \times 9.9^2 \approx 308$ km²

This is a solid community repeater covering roughly the footprint of a small city, though the 308 km² figure is again a ceiling - expect meaningfully less usable coverage. Pockets of shadow behind larger buildings will exist. A wardriving survey is recommended after installation to confirm actual coverage.

Example 3 - Rooftop in Dense Urban

A repeater is placed on the flat roof of a 7-storey (21 m) apartment building in a dense city. Surrounding buildings average 6 storeys (18 m). Mobile clients at street level (1.5 m).

Radio horizon (roof): $4.12 \times \sqrt{21} = 4.12 \times 4.58 = 18.9$ km

Radio horizon (client): $4.12 \times \sqrt{1.5} = 4.12 \times 1.22 = 5.0$ km

Total radio horizon: $18.9 + 5.0 = 23.9$ km

Obstruction factor (dense urban): 0.25

Theoretical coverage radius: $23.9 \times 0.25 \approx 6.0$ km

Theoretical coverage area (ceiling): $\pi \times 6.0^2 \approx 113$ km²

Only 25% of the theoretical range is realised because the repeater barely clears the surrounding roofline. Raising the antenna by 3 additional storeys (to 30 m) lifts the horizon to ~ 27.6 km and, at an improved ~ 0.35 obstruction factor, the ceiling rises to a radius of ~ 9.7 km - roughly $\pi \times 9.7^2 \approx 295$ km² (still a theoretical maximum, with usable coverage well below that). In dense urban networks, rooftop height relative to surroundings matters enormously.

Coverage Overlap: The 20 - 30% Rule

When planning adjacent repeaters, their coverage footprints should overlap by 20 - 30% of the coverage radius. This is a rule of thumb borrowed from cellular and WLAN cell-planning practice (which commonly calls for 15 - 30% overlap). It ensures:

- No gap corridor between repeaters where nodes lose connectivity
- Sufficient signal margin at the cell edge for reliable forwarding (not just barely detectable signals)
- Redundancy: a node in the overlap zone can reach two repeaters

If repeater A has a coverage radius of 8 km and repeater B also has 8 km, place the two sites about 12 - 13 km apart to keep ~ 20 - 30% overlap (a single shared lens-shaped zone roughly 3 - 4 km wide in the middle - not "on each side"). At exactly 16 km the two cells are tangent: they just touch, with zero overlap margin and zero gap. Only beyond ~ 16 - 18 km does a real gap corridor open up where nodes may lose both repeaters.

Link Budget Margin: Target >10 dB

Coverage radius estimates are not hard boundaries - they define the distance at which the link margin drops to zero. In practice, a link operating at exactly the sensitivity floor is unreliable. Fading, multipath, vegetation sway, and atmospheric changes will cause it to fail intermittently.

Target at least 10 dB of margin at the cell edge for reliable operation. This means planning for a coverage radius at which the received signal is 10 dB above the receiver sensitivity floor. The 10 dB figure is a standard engineering convention (see any link-budget reference); the margin you

actually need scales with the environment - budget more in heavy-fading or foliage-dense paths.

For Meshtastic on SF12 / BW125 (the Long Slow preset), receiver sensitivity is approximately -137 dBm with the SX1262's *Rx Boosted* gain enabled (standard gain is $\sim 3 - 4$ dB worse). Note that LongFast (SF11 / BW250) has a sensitivity closer to -131 dBm, not -137 . A link budget target of -127 dBm at the coverage boundary gives 10 dB of margin relative to the SF12/BW125 floor. Links measured below -127 dBm at normal operating distances should be treated as marginal and either reinforced with a relay node or deprioritised until a better repeater site is available.

Quick field check: If a node reports $RSSI < -125$ dBm or $SNR < -10$ dB when communicating with its nearest repeater, that link is at or below the 10 dB margin boundary. Plan to add a relay or move the repeater closer.

The Repeater Grid Approach for Urban Coverage

Why a Grid Approach?

Ad-hoc repeater placement - putting nodes wherever a willing host can be found - produces uneven coverage with clusters of overlapping repeaters in some areas and dead zones in others. A systematic grid approach starts from coverage requirements and works backward to site requirements, ensuring uniform coverage and efficient use of channel capacity.

Target Repeater Density by Area Type

The figures below are planning heuristics, not guarantees: actual spacing and density depend heavily on antenna height and local building/terrain density, so treat them as rough starting points and confirm with a real coverage survey.

Area Type	Target Repeater Spacing	Approx. Density	Rationale
Urban core (high-rise, dense)	1.0 - 2.0 km	1 per 1 - 3 km ²	Heavy building obstruction limits each repeater to a small footprint
Suburban (low-rise, residential)	3.0 - 5.0 km	1 per 7 - 20 km ²	Moderate obstruction; rooftop repeaters can reach 3 - 8 km reliably
Exurban / light industrial	5.0 - 8.0 km	1 per 20 - 50 km ²	Low building density; tall structures (grain elevators, water towers) excellent
Rural (farmland, grassland)	8.0 - 15.0 km	1 per 50 - 175 km ²	Near line-of-sight; hilltop or tower sites dominate

Area Type	Target Repeater Spacing	Approx. Density	Rationale
Wilderness / remote	15.0 - 30.0 km+	1 per 175 - 700 km²	Solar-powered mountain-top repeaters; realistic in open/semi-open terrain only

These spacings are heuristics that assume repeaters are elevated (10 - 30 m AGL for urban, 30+ m for rural) and achieve 10 dB of link margin at the target spacing (cross-reference your own link-budget margin calculations). For low ground-mounted installations, spacing scales roughly with the $\sqrt{h}$ radio-horizon relationship rather than linearly; as a rough rule of thumb, reduce spacing by 30 - 40% (or recompute from the horizon scaling for your specific heights).

Step 1 - Identify Anchor Sites

Anchor sites are high, prominent structures that provide disproportionately large coverage footprints. Finding and securing anchor sites is the most important work in urban coverage planning. Common anchor site types:

- **Water towers:** Heights vary widely (some standpipes are under 20 m, some elevated tanks exceed 50 m; a common range is roughly 25 - 50 m AGL — consult AWWA tank standards if you need precise figures). Often publicly owned, with existing antenna mounts. Utility companies are sometimes receptive to emergency communications partnerships. Ideal anchor nodes.
- **Tall commercial buildings (10+ storeys):** Roof access is harder to obtain but a single mid-rise rooftop repeater in a dense urban core can match 5 - 10 lower installations. Target telecommunications companies, building management firms, or property owners sympathetic to community projects.
- **Hilltops and ridgelines:** In cities built on rolling terrain (Pittsburgh, San Francisco, Seattle), hilltops within or adjacent to urban areas are the highest-value sites. Even a 5 - 10 m height advantage over surrounding terrain dramatically extends range.
- **Radio / TV transmission towers:** Licensed broadcasters and tower companies sometimes offer shared mounting space. The collocation fee may be justified by the coverage gained.
- **Church steeples and clock towers:** Often the tallest structures in older residential neighborhoods. Many faith communities are receptive to community emergency communications projects.

Map all potential anchor sites in a GIS tool. Run viewshed analyses from each. Rank by coverage area. Secure the top 3 - 5 sites before planning secondary fill repeaters.

Step 2 - Fill Planning

Once anchor sites are installed and their actual coverage verified (wardriving survey or community signal reports), overlay the confirmed coverage zones on your planning map. Grid cells with no coverage, or with RSSI below -120 dBm, are fill targets.

Fill repeaters do not need to be as elevated as anchor sites - they only need to bridge a specific gap. A 5 m residential rooftop install may be sufficient to cover a neighborhood dead zone if it is positioned on the line of sight between two anchor sites. Prioritise fill sites that:

1. Cover the largest dead zone area with the smallest number of new nodes
 2. Can hear at least two anchor-tier repeaters (for redundancy)
 3. Are accessible for maintenance
-

Step 3 - Documenting Coverage Gaps

Maintain a living coverage map updated after every wardriving run or community signal report. A shared GIS layer (e.g., a Google MyMaps or an QGIS project shared via cloud storage) accessible to all network administrators is the best tool for this. For each coverage gap, record:

- Geographic boundary of the gap (polygon on the map)
- Estimated area (km^2) and population affected
- Identified candidate fill site(s), if any
- Status: *open* / *candidate identified* / *site secured* / *fill node deployed*
- Last confirmed date of the gap (wardriving date or community report date)

A structured gap log prevents the common failure mode of deploying fill nodes in areas that are already well-covered while neglecting persistent dead zones that lack a vocal advocate.

Worked Example: Planning a Mid-Sized City Grid

Target: A city of 80,000 people covering approximately 120 km^2 , mostly suburban with a 2 km^2 dense downtown core.

1. **Downtown core (2 km²):** Target 1 - 2 km spacing → need 1 - 2 anchor repeaters. Identified a 12-storey bank building and a water tower 1.3 km apart. Both anchor sites secured. Downtown coverage achieved with 2 nodes (2 nodes / 2 km² = 1 per 1 km², within the 1 - 3 km² urban-core target).
2. **Suburban ring (118 km²):** Target 4 km spacing. Hexagonal packing at 4 km site spacing (each site covering a hex of roughly 2 km radius, $\approx 2.6 \times 2^2 \approx 10$ km² per site) gives 118 / 10 $\approx$ **11 - 12 sites**, not 7 - 8 — a triangular/hex grid covers less area per site than a naive square estimate. Identified 9 water towers and 3 church steeples evenly distributed across the suburban ring. All 12 sites secured. Average spacing: ~ 3.9 km (12 nodes / 118 km² = 1 per ~ 9.8 km², within the 7 - 20 km² suburban target).
3. **Total anchor infrastructure: 14 nodes** for 120 km². Report density per zone rather than as a blended average: downtown 1 per 1 km² and suburban 1 per ~ 9.8 km², both meeting their zone targets. (A single blended "1 per ~ 8.6 km²" figure would conceal that each zone is independently compliant.)
4. **Post-wardriving fill:** Survey revealed 3 dead zones in valley neighborhoods. 3 fill repeaters added on rooftops in those valleys. Total: 17 nodes for complete city coverage.

Compare this to an ad-hoc approach: a typical volunteer-driven deployment in a city this size might have 40 - 80 ground-level nodes with significant overlap in connected areas and persistent dead zones in underserved neighborhoods. The grid approach delivers better coverage with far fewer nodes and less channel congestion.

Designing for Redundancy

Why Single-Path Mesh Is Fragile

A tree-topology mesh - where each node has exactly one path back to the network core - is the natural shape that forms when coverage is barely adequate. In a tree topology, the failure of any interior node partitions the network. Nodes "below" the failed repeater become an isolated island: they can hear each other but cannot reach the wider network.

For a community mesh serving emergency communications, this is unacceptable. The very events that trigger heavy mesh use (storms, earthquakes, infrastructure failures) are also the events most likely to take repeaters offline through power loss, physical damage, or access loss.

Per-Zone (+1) Redundancy

Note on terminology: in classic systems engineering, **N+1 redundancy** means N components are needed to carry the load plus *one* shared spare for the whole system. A mesh network needs something stronger than that — a single shared spare for the entire network does not help if the spare cannot reach the area that lost its repeater. What we want for a mesh is **per-coverage-zone redundancy**: every coverage area should have a backup path, not just the network as a whole. In practice, this translates to:

“ **Every node in the network should be able to reach the network core via at least two independent paths through different physical repeaters.** ”

A "network core" is the set of nodes with Internet gateway access or the central coordination point. In a city mesh, the core might be 3 - 5 well-connected anchor repeaters.

To verify per-zone redundancy for a given node:

1. Identify all repeaters that node can directly reach ($RSSI > -120$ dBm).
2. For each of those repeaters, confirm it has at least one other path back to core.
3. If the node can only reach a single repeater, it has no redundancy. If that repeater fails, the node is isolated.

Ring Topology vs. Tree Topology

Tree Topology

Nodes connect to the nearest repeater, which connects to the nearest anchor, which connects to core. This forms a tree. Advantages: simple to plan and understand. Disadvantages: any broken branch isolates all nodes below it. Single points of failure are everywhere.

```
Core
├─ Anchor A
│   └─ Repeater A1
│       └─ Client nodes (isolated if A1 fails)
│   └─ Repeater A2
├─ Anchor B
│   └─ Repeater B1
│       └─ Client nodes (isolated if B1 or Anchor B fails)
```

Ring Topology

Anchor repeaters are interconnected in a ring so that each anchor has two paths back to core. Fill repeaters connect to two anchors where physically possible. This creates a lattice rather than a tree.

```
Core — Anchor A — Anchor B — Anchor C — Core
\   |   /
Repeater Fill Repeater
(hears A (hears (hears B
and B) B and C) and C)
```

Ring topology requires more careful planning and more anchor sites (each anchor must be within radio range of two others), but it eliminates the single points of failure that make tree topologies fragile.

Recommendation: Design anchor-tier repeaters in a ring or lattice. Fill-tier repeaters can remain in a simplified tree to anchor, but each fill node should reach at least two anchors where terrain permits.

Identifying Single Points of Failure with Path Analysis

A single point of failure (SPOF) is any node whose failure disconnects part of the network. Identify SPOFs through path analysis:

1. **Draw the network graph.** Each repeater is a node. Each radio link between repeaters is an edge. Include only reliable links — for example, those with about 15 dB of margin above the receiver's sensitivity. With a typical LongFast/LongSlow sensitivity near -130 dBm, that means including only links measured at roughly -115 dBm or stronger. (Adjust the cutoff to your preset's actual sensitivity floor.)
 2. **Find the bridge nodes (cut vertices).** A cut vertex is any node whose removal disconnects the graph. You can find these visually: any node that is the *only* link between two clusters is a single point of failure. If you have your topology in software, a graph tool (such as Gephi, or Python's `networkx.articulation_points()`) will list them automatically — but for most community meshes, eyeballing the map for any node that is the sole bridge between two groups is enough.
 3. **Prioritise SPOF mitigation.** For each SPOF identified, either add a redundant link (find a fill repeater position that bypasses the SPOF) or ensure the SPOF node has UPS backup power, weatherproof housing, and remote monitoring.
-

Testing Redundancy by Taking a Node Offline

Theoretical redundancy analysis should be validated with live tests. The procedure is simple:

1. **Notify operators.** Announce a planned maintenance window (e.g., "Node X will be taken offline for 30 minutes on Saturday 14:00 UTC for redundancy testing").
2. **Take the target node offline** by powering it down or disconnecting its antenna.
3. **Measure impact.** Using a network map (MeshMapper, Meshtastic node list, or MeshCore admin panel), observe which nodes lose connectivity. Nodes that disappear from the map are isolated - this is your actual failure impact, which may differ from the theoretical prediction.
4. **Document the partition.** Record which nodes were isolated and for how long they would be unreachable in a real failure event.
5. **Restore the node** and plan remediation for any isolated segments found.

Perform redundancy tests at least once per year, and after any significant change to the network topology (adding or removing anchor repeaters, significant coverage expansion).

Practical Redundancy Checklist

- Every anchor repeater can reach at least two other anchors directly.
- Every fill repeater can reach at least two anchors directly.
- No anchor repeater is a single point of failure for more than one fill repeater.
- All anchor repeaters have UPS or generator backup covering at least 72 hours.
- Network graph has been drawn and cut vertices identified.
- Redundancy live-test performed in the last 12 months.
- Failure impact documented: "If node X fails, Y nodes lose connectivity."

Hop Budget and Routing

How hop limits, routing protocols, and multi-hop link budgets determine end-to-end message delivery.

Understanding Hop Count and Hop Limits

What Is a Hop?

A **hop** is a single radio transmission between two adjacent nodes. When a message originates at Node A and is received by Node B, that is one hop. If Node B re-transmits the message and Node C receives it, that is a second hop. Each re-transmission consumes airtime and contributes to channel congestion.

In a well-designed mesh, most messages should arrive in 1 - 3 hops. A message requiring 6 - 7 hops to reach its destination is a symptom of poor repeater placement (insufficient elevated infrastructure) or an oversized network diameter.

Hop Limits in Meshtastic

Meshtastic uses a **hop limit** field in the packet header to prevent infinite re-transmission loops. As of Meshtastic firmware 2.x (per meshtastic.org docs, verified 2026-06-08), the default is **3 hops** and the maximum configurable value is **7 hops**. Defaults can change between firmware versions, so verify against your firmware.

Each time a node re-transmits a packet, it decrements the hop limit by 1. When the hop limit reaches 0, the packet is not forwarded further. This is analogous to the TTL (Time To Live) field in IP networking.

Hop limit | Maximum relay chain

-----+-----

- 1 | Source → 1 relay → Destination
- 2 | Source → 2 relays → Destination
- 3 | Source → 3 relays → Destination (default)
- 5 | Source → 5 relays → Destination
- 7 | Source → 7 relays → Destination (maximum)

Note: the hop limit counts the number of re-transmissions (relays), not the total number of nodes in the chain. A packet with hop limit h undergoes up to h transmissions. The chain shown above is source + h relays + destination, where the destination is reached on the h -th (final) transmission and is itself the last node in the chain (so it does not re-transmit). A packet with hop limit 3 therefore reaches a destination up to 3 hops away.

Why Unlimited Hops Would Cause Broadcast Storms

Meshtastic uses flood routing: every relay node re-broadcasts every packet it receives. Without a hop limit, a packet could circulate indefinitely, with every node in the network repeatedly forwarding it. This creates a **broadcast storm** - a positive feedback loop where each re-broadcast triggers more re-broadcasts, rapidly consuming all available channel time and rendering the network unusable.

Packet deduplication (comparing packet IDs to avoid re-forwarding a packet already seen) and managed-flooding suppression (a node listens briefly before rebroadcasting and stays silent if it hears another node already relay the same packet), together with CSMA/CA contention windows, are the primary mechanisms preventing broadcast storms within a hop radius. The hop limit additionally bounds the maximum path length so a packet cannot circulate indefinitely.

Calculating Maximum Network Diameter from Hop Limit

With a hop limit of h , a packet undergoes up to h transmissions and traverses up to $h + 1$ nodes including the source, with the destination reached on the h -th hop. This means the maximum network "diameter" (the longest path between any two nodes) is h hops.

The figures below are **illustrative only**. Per-hop distance is highly terrain-dependent and is not a Meshtastic-published value: real LoRa links can be sub-1 km in dense urban areas or 100+ km in clear line-of-sight, depending on antenna height, spreading factor, transmit power, and obstructions. The table assumes a nominal ~ 5 km/hop (suburban) and ~ 15 km/hop (rural) purely to show how diameter scales with hop limit - do not treat these as guaranteed ranges.

Hop Limit	Illustrative path length (assumes ~5 km/hop, terrain-dependent)	Illustrative path length (assumes ~15 km/hop, rural, terrain-dependent)
1	5 km	15 km
3 (default)	15 km	45 km
5	25 km	75 km
7 (max)	35 km	105 km

For a well-designed urban network with elevated repeaters, a hop limit of 3 should be sufficient. If your network spans a wide rural area with widely spaced repeaters, you may need hop limit 4 or 5. Increasing beyond 5 is rarely justified and noticeably increases channel congestion.

Meshtastic Flood Routing vs. MeshCore Path-Based Routing

Meshtastic: Flood Routing

In Meshtastic, every node (by default, in CLIENT role) re-broadcasts packets it receives with a non-zero hop limit - flooding is not restricted to the router role. ROUTER/REPEATER roles have higher rebroadcast priority but are not required for flooding. The packet is sent once by each relay, regardless of whether a better path exists - though under managed flooding a relay suppresses its own rebroadcast if it first hears another node relay the same packet. This is simple and robust but inefficient: a 3-hop path in a dense network might result in dozens of redundant re-broadcasts from nodes along parallel paths.

Hop count behaviour: the hop limit is decremented at each relay. A packet leaving the source with hop limit 3 arrives at the destination (if 3 hops away) with hop limit 0. Packets consumed mid-network by deduplication still consumed their transmission slot on the channel.

MeshCore: Path-Based Routing

In MeshCore, the sender stores a learned path in its contact list. The path is embedded into the outgoing packet (source routing); each repeater forwards the packet only if its own address matches the next entry in the embedded path, otherwise it stays silent. Repeaters do *not* maintain a per-node routing table that they consult to pick a next hop. This is a form of source routing, where the sender specifies the path the packet should follow - not hop-by-hop layer-3 IP forwarding-table routing.

Hop count behaviour in MeshCore: MeshCore tracks hop and path information differently from Meshtastic's decrementing HopLimit TTL (the firmware has an internal maximum of 64 hops). This provides useful diagnostic information (you can see how many relays a received message used). MeshCore prevents routing loops through duplicate (packet) detection plus a hop limit: a packet cannot revisit a node (duplicate detection) and eventually expires regardless of network topology.

For network planners: in a Meshtastic network, channel load scales directly with hop limit and network density - every node hears and potentially re-transmits every packet. In MeshCore, channel load is more predictable because only the nodes on the actual path forward a given packet.

When to Increase Hop Limit - and the Tradeoffs

Consider increasing the hop limit above the default of 3 when:

- Your network covers a geographically large area (50+ km diameter) with widely spaced repeaters where 3 hops cannot span the network
- A specific critical path (e.g., a remote outpost to the network core) is consistently failing to deliver messages
- Path analysis shows that known routes require 4 - 5 hops due to terrain

Do not increase hop limit as a first response to poor coverage. Poor coverage is a placement problem, not a hop limit problem. Adding hops to compensate for missing repeaters increases channel congestion for all users without solving the root cause.

The tradeoffs of increasing hop limit:

Hop Limit	Network Reach	Channel Congestion	Typical Use Case
1	Very limited	Minimal	Dense urban, single repeater covers all clients
3 (default)	Moderate	Low - moderate	Well-planned city mesh with elevated anchor repeaters
5	Large	Moderate - high	Regional mesh spanning multiple cities or large rural area
7 (max)	Very large	High	Sparse wilderness mesh; use only when topology analysis confirms need

Designing for Multi-Hop Reliability

Link Budget Through Multiple Hops

In a multi-hop chain, each individual link (hop) must have a positive link margin. Unlike a wired network where signal is regenerated cleanly at each switch, a LoRa repeater decodes the incoming RF signal and re-transmits at full power, so the RF leaving each repeater is clean. But a packet that is only marginally decoded may contain bit errors or fail its CRC and be dropped — so each hop's received signal quality still determines whether the packet survives that hop. The re-transmitted signal is clean, but a weak receive at a repeater can still break the chain.

This is why **the weakest link in the chain determines end-to-end reliability**. If three hops each have reliabilities of 99%, 95%, and 90%, the end-to-end delivery probability is:

$$P(\text{delivery}) = 0.99 \times 0.95 \times 0.90 = 0.846 = 84.6\%$$

This is the key insight of multi-hop network design: each marginal link has an outsized negative effect on overall reliability because the probabilities multiply. A single 85% link with two near-perfect (~99%) links yields about $0.85 \times 0.99 \times 0.99 \approx 83\%$ end-to-end. If the other two links are also only moderate (~90% each), end-to-end delivery drops to roughly 70%.

Planning Hop Paths for Your Network

Before finalising repeater placement, map the expected relay chains for your most critical communication paths. The process:

1. **Draw the relay chain.** On your coverage map, trace the sequence of repeaters a message from a specific source node would traverse to reach the destination. In Meshtastic (flood routing), there may be multiple parallel paths; identify the primary path

(shortest hop count with best margins) and the fallback path.

2. **Estimate link margin at each hop.** For each hop in the chain:
 - Calculate FSPL (Free Space Path Loss): $\text{FSPL(dB)} = 20\text{-log}_{10}(d) + 20\text{-log}_{10}(f) + 92.45$ (d in km, f in GHz)
 - Add terrain/vegetation loss (see Coverage Radius Estimation page for obstruction factors)
 - Add receiver antenna gain; subtract any receiver feedline/connector loss
 - Link Margin = TX Power + TX Antenna Gain – Path Loss + RX Antenna Gain – RX Feedline Loss – RX Sensitivity (all in dB, with TX power and sensitivity in dBm)
 3. **Identify where margin is thin.** A common planning rule of thumb is to flag any hop with less than 10 dB of margin as at-risk. (This is a heuristic, not a fixed standard — planners use anywhere from ~6 dB in benign environments to ~20 dB where high availability or heavy fading is expected.) Options: move the repeater to a better site, add a fill node on that hop, increase antenna gain on that link, or accept reduced reliability on that segment.
 4. **Document the analysis.** Record the estimated margin at each hop. Update after each wardriving survey that provides measured RSSI/SNR data on that link.
-

Worked Example: Three-Hop Path Analysis

Scenario: A remote ranch (Client A) communicates with the county EOC (Destination) via three repeaters (R1, R2, R3). Meshtastic on 915 MHz, SF10/BW125. TX power: 22 dBm (~160 mW) — the SX1262 chip maximum used by stock Meshtastic/MeshCore radios. (FCC Part 15.247 permits up to 30 dBm / 1 W conducted in 902 - 928 MHz, but reaching that requires an external power amplifier, and antenna gain above 6 dBi requires a 1-for-1 dB power reduction to stay within the 36 dBm EIRP limit. The figures below assume the buildable 22 dBm.) Antenna gain: 3 dBi omnidirectional on all nodes. Receiver sensitivity (SF10/BW125): –132 dBm.

Hop 1: Client A → Repeater R1 (hilltop, 4.2 km, rural open)

$$\begin{aligned}\text{FSPL} &= 20\text{-log}_{10}(4.2) + 20\text{-log}_{10}(0.915) + 92.45 \\ &= 12.46 + (-0.77) + 92.45 \\ &= 104.1 \text{ dB}\end{aligned}$$

Terrain loss (rural open): 0 dB additional (clear LOS)

$$\begin{aligned}\text{Link Margin} &= 22 \text{ (TX)} + 3 \text{ (TX ant)} - 104.1 \text{ (FSPL)} + 3 \text{ (RX ant)} - (-132) \text{ (sensitivity)} \\ &= 22 + 3 - 104.1 + 3 + 132 \\ &= 55.9 \text{ dB}\end{aligned}$$

Result: **55.9 dB margin** - excellent. This link is rock-solid.

Hop 2: Repeater R1 → Repeater R2 (rooftop, 9.8 km, suburban)

$$\begin{aligned}\text{FSPL} &= 20\text{-log}_{10}(9.8) + 20\text{-log}_{10}(0.915) + 92.45 \\ &= 19.82 + (-0.77) + 92.45 \\ &= 111.5 \text{ dB}\end{aligned}$$

Suburban obstruction add: +12 dB (unsourced rule-of-thumb clutter allowance; suburban clutter at 915 MHz commonly runs roughly +5 to +15 dB. For a rigorous estimate use a named model such as ITU-R P.1546 or COST-231 Hata.)

$$\begin{aligned}\text{Link Margin} &= 22 + 3 - (111.5 + 12) + 3 + 132 \\ &= 22 + 3 - 123.5 + 3 + 132 \\ &= 36.5 \text{ dB}\end{aligned}$$

Result: **36.5 dB margin** - good, well above the 10 dB minimum.

Hop 3: Repeater R2 → Destination EOC (18.5 km, suburban, EOC is in a building)

$$\begin{aligned}\text{FSPL} &= 20\text{-log}_{10}(18.5) + 20\text{-log}_{10}(0.915) + 92.45 \\ &= 25.34 + (-0.77) + 92.45 \\ &= 117.0 \text{ dB}\end{aligned}$$

Suburban obstruction add: +12 dB

Building penetration loss (EOC indoor): +10 dB (construction-dependent; building penetration at 915 MHz ranges roughly 5 to 25+ dB. See ITU-R P.2109.)

$$\begin{aligned}\text{Link Margin} &= 22 + 3 - (117.0 + 12 + 10) + 3 + 132 \\ &= 22 + 3 - 139.0 + 3 + 132\end{aligned}$$

= 21.0 dB

Result: **21.0 dB margin** - acceptable (above 10 dB), but the EOC indoor penalty is significant. If the EOC uses a rooftop-mounted external antenna instead of an indoor unit, the 10 dB building penalty disappears and margin rises to about 31 dB. Strongly recommend external antenna at the EOC.

Chain Summary

Hop	Distance	Margin (dB)	Status
Client A → R1	4.2 km	55.9	Excellent
R1 → R2	9.8 km	36.5	Good
R2 → EOC	18.5 km	21.0	Marginal (address indoor loss)

End-to-end reliability of this chain is constrained by the R2 → EOC hop. Installing an external rooftop antenna at the EOC is the highest-priority action.

SNR and RSSI Thresholds for Reliable Forwarding

Theoretical link margins are estimates. In a live network, use measured RSSI and SNR to assess actual link quality. The thresholds below are approximate planning heuristics, not measured constants. They are also spreading-factor dependent: the figures here assume a mid-range preset (around SF9 - SF10). LoRa's SNR decode floor ranges from roughly -7.5 dB at SF7 down to about -20 dB at SF12, and at SF11/SF12 a link can decode reliably well below -120 dBm RSSI (sensitivity is around -131 dBm at SF11/250 kHz and -137 dBm at SF12/125 kHz). Treat the cutoffs as *margin relative to your preset's sensitivity* rather than fixed absolute numbers.

Metric	Minimum (marginal)	Target (reliable)	Good	Excellent
RSSI	-130 dBm	-120 dBm	-110 dBm	> -100 dBm
SNR	-15 dB	-10 dB	-5 dB	> 0 dB

Notes on these thresholds (approximate, and SF-dependent — assume roughly SF9 - SF10 here):

- **RSSI -130 dBm / SNR -15 dB:** A rough rule of thumb is a 50 - 70% success rate at a mid-range SF — packets may be decoded but unreliably. (At SF12 these levels are

comfortably decodable.) Use only as an emergency fallback. Do not plan routes through links at this level.

- **RSSI –120 dBm / SNR –10 dB:** Practical minimum for planned routes at a mid-range SF. As a heuristic, expect roughly 85 - 95% packet delivery under normal conditions. Link will degrade in rain, vegetation growth, or when nearby interference increases the noise floor.
- **RSSI –110 dBm / SNR –5 dB:** Reliable for infrastructure links. Acceptable for primary repeater-to-repeater connections. Will maintain >98% delivery in most conditions.
- **RSSI > –100 dBm / SNR > 0 dB:** Strong link. Typical of well-placed nearby repeaters. These links rarely fail under normal operating conditions.

When reviewing live network telemetry, links consistently near the marginal end of the scale for your preset are candidates for remediation. Check the repeater placement, antenna alignment, and cable connections. If the physical setup is already optimal, a fill node on that path may be necessary.

The Weakest-Link Rule in Practice

When troubleshooting poor end-to-end delivery on a multi-hop path:

1. Collect per-hop SNR readings using the Meshtastic traceroute command (from firmware $\geq$ 2.5 it records the route back along with the Signal-to-Noise Ratio for each link — it reports SNR, not RSSI), or MeshCore path diagnostics. Per-link RSSI must be read locally from the node, not via traceroute.
2. Identify the hop with the lowest SNR (or locally-measured RSSI) - this is your weakest link.
3. Improving the weakest link will improve end-to-end delivery more than any other intervention. Do not chase marginal improvements on already-good hops.
4. After fixing the weakest link, re-test the full chain. A new weakest link may emerge.
5. Repeat until all hops meet your margin and SNR targets for the preset in use.

Network Monitoring and Management

Building a Mesh Network Dashboard

A community mesh network dashboard gives operators a real-time view of network health - which nodes are online, battery levels, channel utilization, and connectivity maps. This page covers building a monitoring stack for a Meshtastic network.

Architecture Overview

The standard monitoring stack for Meshtastic:

```
Meshtastic nodes
  ↓ (MQTT uplink)
MQTT Broker (Mosquitto)
  ↓
Telegraf (or Python consumer)
  ↓
InfluxDB (time-series database)
  ↓
Grafana (dashboard and alerting)
```

This stack can run on a Raspberry Pi 4 or any small Linux server. As a rough guide, budget at least 1 - 2 GB RAM (InfluxDB 2.x is memory-hungry and can exceed 500 MB under load) and on the order of 10 GB of storage per month of retained data for a ~50-node network. Actual usage varies with telemetry rate and retention - test on your own hardware before committing to a Pi.

Step 1: MQTT Broker Setup

Install Mosquitto on your monitoring server:

```
sudo apt install mosquitto mosquitto-clients
sudo systemctl enable mosquitto
```

Minimal config at `/etc/mosquitto/mosquitto.conf`:

```
listener 1883
allow_anonymous true
persistence true
persistence_location /var/lib/mosquitto/
```

For production, add TLS and password authentication.

Step 2: Configure Nodes to Uplink

On each node you want to monitor. Note that `uplink_enabled` is a per-channel setting, not an `mqtt.*` module key - there is no `mqtt.uplink_enabled`. Set the module-level keys first, then enable uplink on the channel:

```
# Module-level MQTT settings
meshtastic --set mqtt.enabled true
meshtastic --set mqtt.address "YOUR_SERVER_IP"
meshtastic --set mqtt.json_enabled true

# Per-channel: enable uplink on channel index 0 (repeat for other channels)
meshtastic --ch-index 0 --ch-set uplink_enabled true
```

JSON mode outputs human-readable JSON instead of protobuf binary - easier for custom consumers but includes less data. For full data including all telemetry, use protobuf mode and decode with the meshtastic Python library.

Step 3: InfluxDB

Install InfluxDB 2.x via the official InfluxData apt repository (the snippet below is abbreviated - follow the full add-key + add-repo steps in the InfluxData install guide at docs.influxdata.com/influxdb/v2/install/ before running `apt install`):

```
# Abbreviated - see docs.influxdata.com/influxdb/v2/install/ for the
# complete key import and repository setup steps
wget -q https://repos.influxdata.com/influxdata-archive_compat.key
# ... import the key and add the influxdata apt source, then:
sudo apt install influxdb2
```

```
sudo systemctl enable --now influxdb
```

Create a bucket named "meshtastic" via the InfluxDB UI at <http://localhost:8086>. Set the retention period to match your storage budget - the ~10 GB/month figure above assumes roughly 30-day retention, so a 90-day retention would store about three times as much.

Step 4: Python MQTT-to-InfluxDB Bridge

Note: the bridge below subscribes to `msh/+2/json/#`, the JSON subtopic, so that `json.loads()` only ever sees JSON payloads. If you instead subscribe to `msh/#` you will also receive raw protobuf topics, and `json.loads()` will throw on those binary payloads - the `try/except` simply skips anything that is not valid JSON.

```
pip install paho-mqtt influxdb-client meshtastic

# bridge.py - subscribe to Meshtastic MQTT, write metrics to InfluxDB
import paho.mqtt.client as mqtt
from influxdb_client import InfluxDBClient, Point
from influxdb_client.client.write_api import SYNCHRONOUS
import json, time

INFLUX_URL = "http://localhost:8086"
INFLUX_TOKEN = "YOUR_TOKEN"
INFLUX_ORG = "meshamerica"
INFLUX_BUCKET = "meshtastic"

client = InfluxDBClient(url=INFLUX_URL, token=INFLUX_TOKEN, org=INFLUX_ORG)
write_api = client.write_api(write_options=SYNCHRONOUS)

def on_message(mqttc, userdata, msg):
    try:
        data = json.loads(msg.payload)
    except (ValueError, UnicodeDecodeError):
        # Non-JSON (raw protobuf) payload - skip it
        return
    try:
        node_id = data.get("from", "unknown")
```

```

if "payload" in data:
    p = data["payload"]
    pt = Point("node_telemetry").tag("node_id", node_id)
    if "battery_level" in p:
        pt = pt.field("battery_pct", p["battery_level"])
    if "voltage" in p:
        pt = pt.field("voltage", p["voltage"])
    if "channel_utilization" in p:
        pt = pt.field("channel_util", p["channel_utilization"])
    write_api.write(INFLUX_BUCKET, record=pt)
except Exception as e:
    print(f"Error: {e}")

mqttc = mqtt.Client()
mqttc.on_message = on_message
mqttc.connect("localhost", 1883)
# Subscribe only to the JSON subtopic so json.loads never sees protobuf
mqttc.subscribe("msh/+/2/json/#")
mqttc.loop_forever()

```

The telemetry JSON keys used above (`battery_level`, `voltage`, `channel_utilization`) correspond to the device-metrics telemetry fields; verify the exact key names and nesting against a live TELEMETRY_APP JSON sample from your firmware version, as the serializer can change.

Step 5: Grafana Dashboard

Install Grafana and add InfluxDB as a data source. Useful panels:

- **Battery Voltage by Node** - Time series, grouped by node_id tag
- **Online Node Count** - Count distinct nodes seen in last 30 minutes
- **Channel Utilization Heatmap** - Spot congestion patterns over time
- **Last Seen Table** - Node name and minutes since last packet

Alerting

Configure Grafana alerts to notify via email, Slack, or Telegram:

- Battery below 20% - node needs attention
- Node offline (no data in 2 hours) - repeater may be down

- Channel utilization above 30% - network congestion warning

Using meshmap.net and Community Maps

meshmap.net is the primary public Meshtastic network map. It shows only Meshtastic nodes that have enabled MQTT uplink to the public broker (mqtt.meshtastic.org) with position reporting - an opt-in subset, not all nodes on the mesh. MeshCore nodes are not shown there. It gives operators a quick view of community coverage without building their own infrastructure.

What meshmap.net Shows

- Node positions on an interactive map
- Node names, last seen timestamps, and hardware type
- Neighbor relationships - which nodes have heard each other directly
- Node telemetry where reported (battery, channel utilization)

Note: meshmap.net does **not** display per-link SNR/RSSI values. For per-hop signal quality, use Meshtastic's traceroute (which reports SNR per hop) on your own nodes.

Getting Your Node on the Map

Requirements to appear on meshmap.net:

1. Your node must have a GPS fix or fixed position configured
2. MQTT uplink must be enabled, pointing to mqtt.meshtastic.org (the default Meshtastic public broker)
3. Position reporting must be enabled (not position_precision = 0)

```
meshtastic --set mqtt.enabled true
meshtastic --set mqtt.address "mqtt.meshtastic.org"
meshtastic --ch-index 0 --ch-set uplink_enabled true
meshtastic --ch-index 0 --ch-set module_settings.position_precision 16
```

Note: `uplink_enabled` and `position_precision` are per-channel settings (set with `--ch-set`), not module-level `mqtt.*` keys. There is no `mqtt.uplink_enabled` key.

Nodes usually appear within a few minutes once a valid position is uplinked (the map data refreshes roughly every minute). Position is updated each time the node broadcasts a position packet.

Privacy Considerations

Your node's position is publicly visible on meshmap.net once MQTT uplink is enabled. Note that for `position_precision`, a **higher** number means MORE precise / a SMALLER obfuscation radius (less privacy), and a lower number means coarser / more privacy. Reducing `position_precision` rounds the broadcast location to a coarser grid. Consider:

- Use position precision to reduce location accuracy. Approximate radii: precision 32 = exact, 16 $\approx$ 364 m, 14 $\approx$ 1.5 km, 13 $\approx$ 2.9 km, 12 $\approx$ 5.8 km, 11 $\approx$ 11.7 km, 10 $\approx$ 23.3 km. For roughly 1 km of obfuscation use precision 14 (not 10); for $\sim$ 10 km use precision 11. This shows the general area without revealing exact location.
- Fixed infrastructure repeaters: full precision is usually acceptable and helpful for community planning
- Personal/portable nodes: reduced precision (or disabling MQTT uplink) may be preferred

```
meshtastic --ch-index 0 --ch-set module_settings.position_precision 14
```

Alternative and Regional Maps

Beyond meshmap.net, several regional communities maintain their own maps:

- Custom Grafana + Leaflet maps fed by self-hosted MQTT brokers give communities more control over who appears and what data is shown
- Some communities use ATAK (Android Team Awareness Kit) with Meshtastic integration for a more sophisticated operational picture
- map.meshcore.io is the MeshCore node map, showing MeshCore nodes in regions with active MeshCore communities

Reading Neighbor Info for Coverage Analysis

The Neighbor Info module in Meshtastic (Config → Modules → Neighbor Info) broadcasts a list of directly-heard nodes to the mesh. When this data reaches meshmap.net, it draws link lines between nodes that can hear each other. These link lines are the basis for understanding your network's topology:

- Isolated nodes with no link lines - check if repeaters are within range
- Nodes connected only through a single relay - identify single points of failure
- Dense link clusters - confirm your urban repeater placement is achieving coverage

Mesh Network Change Management

A community mesh network is shared infrastructure. Changes to configuration - channel presets, node roles, frequency settings - can disrupt all users if done carelessly. This page covers change management practices that keep the network stable and community trust intact.

Why Change Management Matters

Unlike a traditional IT network with change control systems, community mesh networks are informal. A single operator changing the channel name or PSK on a key repeater can silently disconnect all users who have that repeater in their channel config. Changes that seem small to one operator can have large effects on the whole network.

Changes That Require Community Coordination

Change Type	Impact	Required Notice
Channel name or PSK change	All users on that channel lose connectivity until they update	Required - announce in advance, coordinate cutover time
Modem preset change	All nodes on different preset cannot hear this repeater	Required - network-wide coordination needed
Frequency slot change	Same as preset change	Required
Hop limit change (reduce)	May isolate edge nodes from network core	Recommended
Node role change	Affects routing if changing to/from Router	Recommended
Firmware update (minor)	Minimal - backward compatible	Good practice to announce

Change Type	Impact	Required Notice
Firmware update (major)	May affect interoperability with older nodes	Required for key infrastructure
Node taken offline (temporary)	Routing reroutes; may cause gaps	Recommended - notify community
Node permanently removed	Route cache entries become stale	Required - update network documentation

Change Communication Channels

Establish at least one out-of-band (non-mesh) communication channel for network announcements:

- Community Discord server with a #network-changes channel
- Signal or Telegram group for operators
- Email list for major announcements
- Ham radio net for ARES-affiliated networks

Announce significant changes at least 48 hours in advance for planned maintenance. Emergency changes (a node causing harm to the network) can happen immediately but should be communicated as soon as possible.

Testing Changes Before Deploying

1. Test configuration changes on a non-critical node first (a personal portable node, not the main community repeater)
2. Verify the change works as expected with a test partner node
3. Document the before-and-after configuration
4. Have a rollback plan: know how to undo the change if it causes problems
5. Apply to production during low-traffic hours

Maintaining a Network Configuration Ledger

Keep a simple spreadsheet or wiki page with current configuration for every community node:

- Node name and operator
- Physical location

- Current firmware version
- Channel configuration (name, PSK, preset)
- Role setting
- Last configuration change date and who made it

This ledger prevents "mystery configuration" situations where no one knows why a node is behaving unexpectedly because the original operator is no longer reachable.

Spectrum Analysis and RF Tools

Using an SDR for 915 MHz Band Analysis

A Software Defined Radio (SDR) is one of the most useful tools for mesh network operators: it lets you visualize the actual RF environment your nodes operate in, identify interference sources, and verify that your nodes are transmitting on the correct frequencies.

Getting Started with RTL-SDR

The RTL-SDR (RTL2832U based USB dongle) is the most accessible SDR for mesh operators:

- **Cost:** roughly \$30 dongle-only, or about \$40 with the dipole antenna kit, for an RTL-SDR Blog V4 (the current recommended version; MSRP as of 2026-06-08)
- **Coverage:** 500 kHz to 1750 MHz - covers the entire 902-928 MHz ISM band
- **Software:** SDR# (Windows), GQRX (Linux/Mac), SDRangel (cross-platform)
- **Antenna:** The included whip antenna works at 915 MHz; a dedicated 915 MHz antenna improves sensitivity

```
# Install SDR# on Windows:  
# Download from airspy.com/download, extract, run SDRSharp.exe  
  
# GQRX on Ubuntu:  
sudo apt install gqrx-sdr
```

Configuring SDR# for 915 MHz Observation

1. Set center frequency to 915,000,000 Hz (915 MHz)
2. Set sample rate to 2.4 MHz. Note this shows only ~2.4 MHz of spectrum at once (about 913.8-916.2 MHz at a 915 MHz center) — the RTL-SDR cannot display the full 26 MHz band simultaneously. To survey the whole 902-928 MHz band, step the center frequency

across the band in ~2 MHz increments (or use a wideband SDR / scan feature).

3. Enable WFM or Raw I/Q mode (you're looking at signal presence, not decoding)
4. Enable the spectrum analyzer and waterfall displays
5. Set FFT size to 32768 for high resolution

The waterfall shows frequency (horizontal) vs. time (vertical, scrolling). Each LoRa transmission appears as a faint chirp pattern - rising or falling tones, typically 250 kHz wide for the common Meshtastic presets (range 125-500 kHz depending on preset).

What to Look For

Normal LoRa Activity

LoRa transmissions are characterized by chirp spread spectrum - the signal appears as a diagonal streak in the waterfall (rising chirp = upchirp, falling = downchirp). A healthy mesh network shows occasional bursts of activity at the configured center frequency.

Interference Sources

- **Constant carrier (narrow spike):** Could be a CW interferer, oscillator leakage, or a malfunctioning device
- **Wide noise floor increase:** Could be FHSS device (900 MHz cordless phone), wideband noise from switching power supply
- **Pulsed narrowband:** Smart meter AMI networks (itron, Landis+Gyr) often operate in 902-928 MHz; appears as regular narrow pulses
- **Broadband hash:** Arc welders, brush motors, and variable-speed drives produce broadband electrical noise that raises the noise floor broadly

Measuring Channel Utilization Empirically

SDR# can be used to empirically measure how busy your mesh channel is:

1. Tune to your network's center frequency
2. Record 10-15 minutes of waterfall data
3. Count the number of LoRa packet events per minute

4. Estimate channel occupancy from the LoRa airtime, not from a raw bitrate. (LoRa data rate is low: SF9/250 kHz is only $\sim 1.7\text{-}3$ kbps, far below the 250 kHz bandwidth — do not confuse the two.) A typical ~ 50 -byte packet at SF9/250 kHz has an airtime of roughly 150 ms, so $10 \text{ packets/min} \times 0.15 \text{ s} \approx 2.5\%$ channel occupancy. Slower spreading factors (SF11/SF12) have much longer airtimes and reach high occupancy with far fewer packets.

The [Meshtastic app](#) reports Channel Utilization as a device metric (via the Telemetry module / node info, often shown as ChUtil) - check this before breaking out the SDR. The SDR is most useful when you suspect non-LoRa interference.

NanoVNA Guide for Mesh Antenna Work

The NanoVNA is an affordable vector network analyzer that every serious mesh network operator should own. It measures antenna SWR, impedance, and resonant frequency directly - letting you verify antennas before installation and diagnose field problems.

What a NanoVNA Measures

- **SWR (Standing Wave Ratio)** - How well your antenna is matched to 50 ohms at each frequency
- **S11 / Return Loss** - The reflection magnitude in dB, reported as a **positive** number (larger means a better match). It is mathematically related to SWR through the reflection coefficient (return loss = $-20 \log_{10} |\Gamma|$; $SWR = (1 + |\Gamma|) / (1 - |\Gamma|)$) and shows resonance as a dip
- **Impedance (R + jX)** - The complex impedance of the antenna at each frequency
- **Smith Chart** - Graphical representation of impedance; useful for matching network design

NanoVNA Selection

For 915 MHz work, any NanoVNA covering 300 kHz to 1.5+ GHz will work (prices as of 2026-06-08; verify current pricing and specs against the vendor listing):

- **NanoVNA-H4** - \$55-70, 4-inch screen, covers to 1.5 GHz. Best for comfortable field use.
- **NanoVNA-F v2** - around \$120; commonly listed with coverage to ~3 GHz and improved calibration (confirm the exact frequency ceiling against the manufacturer/vendor spec). Good if you also do 2.4 GHz work.
- Avoid no-name clones below \$40 - calibration and accuracy are often poor.

Calibration Procedure

Calibration must be done before every measurement session, set for the exact frequency span you're testing. If you later change the sweep span (or any adapter or cable), you must re-run calibration:

1. Open the menu and select **CAL → RESET** to clear any previous calibration
2. Set the frequency span first: **STIMULUS → START/STOP** = 850 MHz / 980 MHz (bracket the 902-928 MHz band). Calibration is only valid for the span you set here
3. Select **CAL → CALIBRATE**, then with the **OPEN** standard attached to the CH0 port, press OPEN
4. Replace it with the **SHORT** standard; press SHORT
5. Replace it with the **LOAD** (50 ohm) standard; press LOAD
6. For antenna SWR (an S11-only, one-port measurement) you can skip the **THRU** and **ISOLN** steps - those apply to two-port (S21) measurements
7. Press **DONE**, then **SAVE** to a calibration slot (e.g., SAVE 0)

Critical: Calibration is performed at the end of your test cable (the SMA port that will connect to the antenna). Every adapter or cable change - and every change to the frequency span - requires recalibration.

Measuring a Mesh Antenna

1. Calibrate NanoVNA at the test port
2. Connect antenna under test to CH0
3. Enable S11 display in SWR mode
4. Set Y-axis to SWR 1-3 range for easy reading
5. Identify the frequency where SWR dips to its minimum - that's the antenna's resonant frequency
6. Read the SWR at 915 MHz specifically

Interpreting Results

SWR at 915 MHz	Interpretation	Action
1.0 - 1.5	Excellent match	Deploy with confidence
1.5 - 2.0	Good match	Acceptable; 89-96% power transfer
2.0 - 3.0	Fair match	Investigate antenna type/connector
3.0+	Poor match	Likely a wrong-frequency or damaged antenna, or a connector/feedline fault
Flat (no dip anywhere)	Open or short circuit	Check connector and cable continuity

Tuning a DIY Antenna

If your DIY antenna resonates slightly off 915 MHz, correct it by changing the element length. Note that a too-high resonance requires *adding* length (you cannot fix it by trimming), while a too-low resonance is corrected by trimming:

- **Resonant frequency too high** (antenna resonates at 920 MHz instead of 915) - antenna is too short; lengthen it (a longer element or added wire). Trimming will not fix this case
- **Resonant frequency too low** (antenna resonates at 910 MHz) - antenna is too long; trim carefully in 2mm increments
- Re-measure after each change until resonant frequency matches 915 MHz

Channel and Frequency Planning

Meshtastic Channel Number Selection Guide

Meshtastic has two distinct concepts that are easy to confuse. The **LoRa Frequency Slot** (`lora.channel_num`) selects the single center frequency the radio transmits on. This is separate from the up-to-8 **logical Channels** (each with its own name and PSK, indexes 0-7), which **all share the same frequency**. Adding logical channels gives you separate encrypted message streams — it does **not** change your RF frequency, provide RF isolation, or avoid collisions. Only changing the frequency slot (or the modem preset) actually moves you to a different frequency.

How the Frequency Is Chosen

Meshtastic does **not** compute the center frequency from a logical channel number using a simple base + spacing formula. Instead, the firmware divides the region's band into a fixed number of **frequency slots** (slot width = the preset's bandwidth) and picks one slot:

- If `lora.channel_num` is **0 / UNSET** (the factory default), the firmware computes a **hash of the PRIMARY channel's NAME** (a djb2 hash in `RadioInterface.cpp`) and uses it to select a slot.
- If `lora.channel_num` is set to **1-104**, that value is an **absolute slot index** and overrides the name hash.

For the **US 902-928 MHz band with the LongFast preset, the bandwidth is 250 kHz**, which yields **104 frequency slots** of 250 kHz each across the 26 MHz usable band. (125 kHz is the bandwidth of the Long Moderate / Long Slow presets, not LongFast — those presets have more slots.)

```
# US region, LongFast preset
# Bandwidth (slot width): 250 kHz
# Number of frequency slots: 104

# Default channel_num = 0/UNSET -> frequency comes from a HASH of
# the PRIMARY channel NAME, NOT from a linear base+offset formula.
# The default LongFast channel name hashes to slot 20 = 906.875 MHz.
```

```
# Slot-to-frequency formula (only when you set an explicit slot):  
#   freq_MHz = freq_start + (slot * bandwidth) + (bandwidth / 2)  
# US: freq_start = 902.0 MHz, bandwidth = 0.250 MHz, slot is 1-based (1..104)  
#  
# Example, default LongFast slot 20:  
#   902.0 + (20 * 0.250) + 0.125 ... is reached via the name hash,  
#   giving the documented default of 906.875 MHz.
```

Note that **906.875 MHz corresponds to frequency slot 20** (the result of hashing the default channel name), *not* "channel 0." There is no `base 902.0 + channel × 0.125` linear progression, and the slot index is not limited to 0-7 — those are the logical channels, a completely separate concept.

Why Change the Default Slot?

Most Meshtastic nodes ship on the default LongFast channel, which hashes to slot 20 (906.875 MHz). If your area already has significant Meshtastic traffic, you may benefit from staying on the default to maximize connectivity with existing nodes. However, if:

- You're running a private community network that wants to operate on its own frequency (note: changing frequency means those nodes can no longer hear the default mesh, and vice versa)
- You've identified interference at the default frequency with an SDR
- You want to operate on a less crowded frequency in a dense metropolitan area

...then setting an explicit frequency slot (or choosing a different primary channel name that hashes to a different slot) makes sense. Changing logical channels alone will **not** do this — logical channels all share the same frequency.

Frequency Coordination

For all nodes in your community network to hear each other, they must transmit on the **same frequency** — meaning the same modem preset *and* the same frequency slot (which means either the same primary channel name, since the name-hash sets the slot, or the same explicitly set `lorawan.channel_num`). They must also share the same channel name/PSK to decode each other's traffic. Keep a community coordination document:

```
# Check current frequency slot:  
meshtastic --get lorawan.channel_num
```

```
# Set an explicit frequency slot (absolute slot index, 1-104 for US LongFast):
meshtastic --set lora.channel_num 3

# This selects frequency SLOT 3, an absolute slot - NOT an offset added
# to 906.875. For US LongFast (250 kHz):
# 902.0 + (3 * 0.250) + 0.125 = 903.375 MHz
#
# Setting channel_num to 0 reverts to the channel-name hash (default).
```

Bandwidth and Frequency-Slot Separation

Two independent networks on the same band coexist by transmitting on **different frequency slots** (set via different slot numbers, or different primary channel names that hash to different slots). Separating two networks by ≥ 2 slots reduces adjacent-channel interference, but it does **not** give complete RF isolation — LoRa's spectral skirts extend beyond the nominal bandwidth and the 902–928 MHz band is shared ISM spectrum, so some interference is always possible. (Separate channel names/PSKs alone provide only *logical* separation — same frequency, same airtime, same collisions.) Practical guideline:

Modem Preset	Bandwidth (slot width)	Separation for substantial isolation
LongFast	250 kHz	≥ 1 slot (250 kHz); ≥ 2 slots (500 kHz) for better margin
MediumFast	250 kHz	≥ 1 slot (250 kHz); ≥ 2 slots (500 kHz) for better margin
ShortFast	250 kHz	≥ 1 slot (250 kHz); ≥ 2 slots (500 kHz) for better margin
ShortTurbo	500 kHz	≥ 1 slot (500 kHz); ≥ 2 slots (1000 kHz) for better margin

If two networks operate on adjacent slots with the same bandwidth, they'll still experience some adjacent-channel interference and won't be completely isolated. Remember that putting your network on a different frequency slot from the local default mesh means the two networks can no longer relay for each other — that's the intended trade-off when you want a private frequency.

Operating Multiple Channels on One Network

Running multiple Meshtastic channels on the same network infrastructure enables privacy separation, role-based access control, and operational flexibility. This page covers design patterns for multi-channel community networks.

Why Multiple Channels?

- **Privacy tiers:** Public channel (open access) + private community channel (member-only key) + infrastructure/ops channel (operators only)
- **Functional separation:** General chat, emergency/tactical, telemetry/MQTT, admin coordination
- **Geographic zones:** Neighborhood channels within a metro-wide network
- **Inter-network bridging:** A bridge node can monitor a second network's channel while primarily operating on your community channel

Channel Capacity

Meshtastic supports up to 8 channels simultaneously on a single node (slots 0-7). Channel 0 is the primary channel (all nodes have it). Channels 1-7 are secondary and only active on nodes configured to use them.

All enabled channels share the same LoRa radio and the same RF frequency - the node transmits and receives on one frequency at a time. Adding channels does not add spectrum; it adds logical (name/PSK) separation only. Multiple channels increase channel utilization proportionally with message volume on each channel, because every channel's traffic competes for the same airtime.

Recommended Multi-Channel Configuration

The commands below use the official Meshtastic Python CLI. `--ch-add` creates a new secondary channel, and `--ch-index N --ch-set <field> <value>` sets a field on the channel at that index (see the Meshtastic CLI reference at meshtastic.org/docs/software/python/cli/).

```
# Channel 0: Public community channel
meshtastic --ch-index 0 --ch-set name "PDXMesh"
meshtastic --ch-index 0 --ch-set psk "community-key-base64=="
meshtastic --ch-index 0 --ch-set uplink_enabled true # optional: MQTT bridge
meshtastic --ch-index 0 --ch-set downlink_enabled false # prevent loops

# Channel 1: Operations/admin (limited to infrastructure operators)
meshtastic --ch-add
meshtastic --ch-index 1 --ch-set name "PDX-Ops"
meshtastic --ch-index 1 --ch-set psk "ops-key-base64=="
meshtastic --ch-index 1 --ch-set uplink_enabled false # keep off MQTT
meshtastic --ch-index 1 --ch-set downlink_enabled false

# Channel 2: Emergency (activated during events)
meshtastic --ch-add
meshtastic --ch-index 2 --ch-set name "PDX-EmComm"
meshtastic --ch-index 2 --ch-set psk "emcomm-key-base64=="
```

Channel Key Management

With multiple channels, key management becomes important:

- Store all channel keys in a password manager (Bitwarden, 1Password) shared with authorized operators
- Document which nodes have which channels configured - a node without channel 2 will miss emergency messages
- When a key is compromised, change only the affected channel key - other channels are unaffected
- Establish a key rotation schedule: community channel annually, ops channel quarterly, emcomm channel pre-activation

Traffic Impact

Each additional active channel increases channel utilization by the traffic that channel carries — all channels share one radio and one frequency, so their airtime adds up. Monitor total CU across all

channels:

- With 3 channels, target total CU under 20% combined (channel utilization is reported over a 1-minute window; firmware self-throttles, deferring its own transmissions, as CU rises around 25%)
- Low-activity channels (ops, emcomm) add minimal overhead: mostly periodic NodeInfo broadcasts
- A busy community-chat channel can contribute a substantial share of total CU; the exact percentage depends on message rate, packet size, and modem preset, so measure it on your own network rather than assuming a fixed figure