

# Troubleshooting Advanced Issues

- [My node is flooding the channel with repeated messages](#)
- [My Meshtastic node and MeshCore node cannot communicate](#)
- [RF interference is affecting my node — how do I diagnose it](#)

# My node is flooding the channel with repeated messages

## My node is flooding the channel with repeated messages

If you notice unusually high channel utilization, see the same message appear many times in quick succession, or other mesh users report your node is hammering the airwaves, your node may be contributing to a flooding condition. Here is how to understand it, diagnose it, and fix it. (Note: the role and menu details below are Meshtastic-specific. If you run a MeshCore repeater that is over-flooding, MeshCore repeaters use selective/path-based forwarding rather than managed flooding - reduce duty cycle and how often the node broadcasts its position/adverts to cut congestion.)

## Is flooding always a problem?

Some re-broadcasting is normal and intentional in a mesh network. When a node receives a packet it has not seen before and its hop limit allows, it re-broadcasts so the message can reach nodes that did not hear the original. Meshtastic uses a *managed flood routing* approach where every ROUTER node re-transmits. What becomes a problem is when the same packet circulates far more times than needed, driving channel utilization above the recommended 25% level. Meshtastic treats roughly 25% as the green/optimal soft ceiling for channel utilization: above it the firmware's contention window scales up and TX is deferred, collisions increase, and overall throughput degrades for every node on the channel.

## Common causes

- **Two ROUTER nodes in direct line-of-sight of each other:** When both nodes hear each other perfectly and both re-broadcast the same packet, they can trigger a forwarding loop if the hop count has not decremented to zero. Each re-broadcast is heard by the other, which may queue another retransmit depending on firmware version.

- **Bridge nodes on the same channel:** Nodes configured as bridges can inadvertently re-introduce packets to a segment that already received them if the bridge logic is not configured correctly, creating a loop.
- **Hop limit set too high:** The default hop\_limit is 3, meaning a packet can traverse up to 3 intermediate nodes before being dropped (the maximum is 7). If it has been set to 7 or higher, packets live long enough to circle back through the network.
- **Multiple nodes with the same node key or ID:** Duplicate node identities confuse deduplication logic, causing packets that should be discarded to be treated as new.

## How to diagnose

1. Open the [Meshtastic app](#) and check your node's channel utilization in its device metrics. If utilization is above 25% and rising when no one is actively sending, something is re-broadcasting excessively.
2. Check the node list for any duplicate node names or IDs that might indicate a misconfigured clone.
3. Temporarily power down nodes one at a time and watch whether utilization drops - this isolates the culprit node.

## Fixes

- **Reduce hop\_limit to 3** (Radio Config → LoRa → Hop Limit). For most deployments, 3 hops is sufficient to cross a well-designed mesh.
- **Change ROUTER nodes to CLIENT role** for nodes that do not need to relay traffic. CLIENT nodes do not re-broadcast, which dramatically reduces airtime.
- **Review bridge node configuration:** Ensure bridge nodes have channel isolation configured correctly so they do not echo packets back onto the originating channel.
- **Enforce firmware-level deduplication:** Keep all nodes on current stable firmware. Older firmware versions had less aggressive deduplication windows.
- **Stagger node placement:** If two ROUTER nodes are in perfect line-of-sight, consider whether both need to be ROUTERs, or whether one can be demoted to CLIENT.

Keep in mind that each relaying node adds another full transmission of the packet, so total airtime scales with the number of relays (roughly one transmission per relay plus the original), not just with hop count.

# My Meshtastic node and MeshCore node cannot communicate

## My Meshtastic node and MeshCore node cannot communicate

This is one of the most common points of confusion for new mesh users: both Meshtastic and MeshCore run on similar LoRa hardware, operate in the same 915 MHz band, and can even share the same physical channel frequency - yet they cannot exchange messages with each other. Here is why, and what your options are.

### Why they are incompatible

Meshtastic and MeshCore are completely independent software stacks. While both modulate data over LoRa radio, they differ in every layer above the physical radio signal:

- **Packet format:** Meshtastic uses Protocol Buffers (protobuf) serialized payloads inside its own envelope structure. MeshCore uses a different binary packet format with its own header fields, addressing, and payload encoding. A MeshCore node receiving a Meshtastic packet sees garbage bytes it cannot interpret, and vice versa.
- **Encryption:** Meshtastic encrypts channel traffic with AES (AES-128 or AES-256 depending on the pre-shared channel key length; the default "LongFast" channel uses a short, publicly known key that provides no real privacy). MeshCore uses a different scheme - AES-128 for channel traffic with Curve25519 public keys for direct messages. The two use incompatible key material and algorithms, so even if a node could parse the outer envelope, it cannot decrypt the other's payload.
- **Routing logic:** Meshtastic uses managed flood routing with hop decrement. MeshCore uses a different routing model oriented around repeater nodes and room servers. The routing metadata fields in each protocol are incompatible.
- **Node identity:** Meshtastic identifies nodes by a 32-bit node number derived from hardware MAC. MeshCore uses a public-key-based identity system. There is no translation

layer between them.

## There is no bridge - yet

As of the current date, no production bridge firmware or software exists that can transparently relay messages between a Meshtastic mesh and a MeshCore mesh in real time. Research and community projects have explored the concept, but none have reached a state suitable for general use.

## Workarounds for cross-protocol communication

If your community includes both Meshtastic and MeshCore nodes and you need them to share information, the following approaches can help:

- **Room server as intermediary:** A MeshCore room server is normally a mesh contact reached over RF (or Bluetooth) by selecting it from your contact list - it is not an internet service by default. Some operators bridge a room server to the internet as a separate, non-default and uncommon setup; this is an advanced configuration, not something that works out of the box. Even where such a bridge exists, it only connects MeshCore clients to that room server - it does not let a Meshtastic node join.
- **Manual human or gateway relay:** The only reliable cross-protocol path today is a human (or a custom script) reading messages off one network and re-posting them on the other. Note that **Winlink is not a bridge between Meshtastic and MeshCore:** Winlink is an amateur-radio email system that requires a valid amateur radio license and separate HF/VHF gear or an internet connection, and neither Meshtastic nor MeshCore connects to it natively. Do not plan an interop path around Winlink.
- **Dual hardware stacks:** Deploy two physically separate nodes at the same location - one running Meshtastic and one running MeshCore. A human operator or automated script on a co-located computer can relay messages between the two networks over their respective serial or Bluetooth APIs.

## Recommendation

For a new community deployment, choose one protocol and standardize. Mixed-protocol communities incur significant coordination overhead. If you are integrating with an existing regional mesh, match whatever protocol that mesh uses.

# RF interference is affecting my node — how do I diagnose it

## RF Interference Is Affecting My Node - How to Diagnose It

LoRa spread-spectrum modulation gives it excellent resistance to narrowband interference, but it is not immune. If your node is experiencing unexplained packet loss, poor RSSI from nearby nodes, or erratic behavior that does not correlate with distance or obstacles, RF interference may be the culprit.

### Symptoms of interference

- **High SNR variation:** For instance, a link that normally shows around -5 dB SNR may suddenly drop to -15 dB or worse for no physical reason. (Baseline SNR varies from link to link; the diagnostic signal is the sudden, unexplained change, not the absolute number.) If SNR fluctuates wildly over minutes without any change in node position, an external RF source is likely corrupting packets.
- **Unexpectedly poor RSSI from nearby nodes:** RSSI substantially worse than your link budget predicts can indicate that wideband noise is raising the noise floor. Link-budget estimates carry large uncertainty, so treat a clear, persistent gap between predicted and observed RSSI - not a precise dB figure - as the warning sign.
- **Time-of-day failure patterns:** If failures cluster during business hours, when appliances run, or at regular intervals (e.g., every 30 minutes), a duty-cycling device such as a smart meter network is a likely source.
- **One direction fails more than others:** If nodes in one geographic direction consistently perform worse, a directional interference source may be oriented toward your antenna.

### Diagnosis tools

The most effective way to see what is happening in the 902 - 928 MHz band is with a Software Defined Radio (SDR):

- **Hardware:** An RTL-SDR dongle (~\$25 USD) with a 915 MHz antenna is sufficient. Higher-end options like HackRF or Airspy offer better sensitivity.
- **Software:** SDR# (Windows) or GQRX (Linux/macOS) - both are free. Open the spectrum analyzer view and zoom to 902 - 928 MHz.
- **What to look for:** Constant carriers (a vertical spike that never moves) indicate a narrowband source. Sweeping signals suggest frequency-hopping devices such as cordless phones. Periodic bursts at regular intervals suggest smart meter networks (ITRON, Sensus, etc.) which use this band extensively in North America.

## Common interference sources in the 902 - 928 MHz band

- Baby monitors (some older analog models use 900 MHz, though most modern units operate at 2.4 GHz or on DECT at ~1.9 GHz)
- 900 MHz analog/digital cordless phones (older models - note these are not DECT, which is a separate ~1.9 GHz standard)
- Smart meter networks (AMI/AMR systems from utilities)
- Industrial wireless sensors and SCADA equipment
- Some older Wi-Fi extenders and video senders

## Mitigation strategies

- **Change frequency slot (channel) in Meshtastic:** Meshtastic lets you select a different frequency slot within the ISM band while keeping the same modem preset. If a specific frequency is congested, moving to a different frequency slot shifts your center frequency away from the interference - this is the correct way to dodge a narrowband interferer. **Important:** the modem preset (LongFast vs. LongSlow vs. MedFast) sets the spreading factor and bandwidth (speed vs. range), *not* the center frequency. Changing the preset does not move you off an interfering frequency, and every other node in your mesh must use the identical region and preset to communicate - if you change your preset you will lose contact with all nodes still on the old preset. Change the frequency slot, not the preset, and keep your mesh on a common preset.
- **Use a directional antenna:** A Yagi or patch antenna pointed toward your intended mesh nodes, and away from the interference source, provides front-to-back rejection typically on the order of 10 - 20 dB, depending on the antenna design (a small patch may give less; a larger multi-element Yagi can give more).
- **Relocate the antenna:** Moving the antenna even a few meters can place a building or terrain feature between your node and the interference source, providing significant shielding.

- **Reduce antenna height:** Counter-intuitively, lowering an antenna can reduce pickup of distant interference while maintaining adequate coverage of nearby nodes.