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

---

Revision #3
Created 2026-05-03 05:37:35 UTC by Mesh America Admin
Updated 2026-06-09 22:38:55 UTC by Mesh America Admin