

# MeshCore Path Discovery

## Deep Dive

A detailed look at MeshCore's flood-first / direct-route-after path learning, derived from `docs/packet_format.md` and the firmware source. MeshCore does not use a dedicated route-discovery protocol (there is no AODV, Route Request, or Route Error); paths are learned as a byproduct of the initial flood.

## The Core Mechanism

MeshCore's routing uses two phases that together reduce the channel waste of pure flooding. Note that only the initial flood has path redundancy: once a direct path is learned, subsequent messages follow that single path with no alternate-route redundancy.

### Phase 1: Flood with Path Recording

As a `ROUTE_TYPE_FLOOD` packet (route type 0x01) traverses repeaters, each forwarding repeater appends its path-hash to the packet's `path` field, so the destination receives the full ordered path carried inside the packet itself. The path is tracked in the flooded packet, not recorded separately at the destination.

### Phase 2: Path Return via `PAYLOAD_TYPE_PATH`

The destination sends a `PAYLOAD_TYPE_PATH` response back to the original sender. This packet contains the ordered list of repeaters from the flood path. The path packet itself uses direct routing if a reverse path is known, otherwise floods back.

### Phase 3: Stored Direct Routes

The sender stores the learned path and uses it for all subsequent messages to that destination using `ROUTE_TYPE_DIRECT`. The path remains valid until a direct-routed message fails (no

acknowledgement within timeout), at which point the sender falls back to flooding and re-learns the path. Because a direct-routed message follows a single learned path with no redundancy, if one repeater on that path fails, messages are silently lost until the timeout triggers a re-flood — only the initial flood has path redundancy.

# Path Hash Modes

MeshCore supports configurable path hash modes via the `set path.hash.mode {0|1|2}` CLI command, which selects the advert path-hash *size* (0 = 1 byte, 1 = 2 byte, 2 = 3 byte). The **default is 0** (1-byte). This setting affects only the size of the path-hash a repeater uses in its *own* advert broadcasts; it does **not** change how path uniqueness is determined, how the routing table is managed, or what packet ID/hash size this repeater forwards. The feature requires firmware  $\geq$  1.14.

# Flood Limits

To prevent unbounded flooding, MeshCore allows configuration of `flood.max` (the maximum hop count a repeater will forward a flood packet to, related to the 64-hop firmware ceiling) and `flood.advert.interval` (the flood-advert cadence in hours). These limit worst-case channel usage.

# Regional Scoping

Transport route types (`ROUTE_TYPE_TRANSPORT_FLOOD` and `ROUTE_TYPE_TRANSPORT_DIRECT`) include a 2-byte transport code that allows multi-region networks to scope traffic appropriately. This transport code is calculated from the region scope — a user-defined named scope (e.g. `#Europe`, `#USA`) hashed into the 2-byte value — not an ISO country code.

*Sources: route types, the path field, transport codes, and MAX\_PATH\_SIZE (64) are per `docs/packet_format.md`. CLI parameters (`path.hash.mode`, `flood.*`, region scopes) are per the MeshCore CLI docs. Verified 2026-05-03.*

---

Revision #4

Created 2026-05-03 06:45:43 UTC by Mesh America Admin

Updated 2026-06-09 14:21:51 UTC by Mesh America Admin