# Path Discovery and Route Learning

MeshCore uses a hybrid **flood-first / direct-route-after** approach rather than persistent flooding. For stable repeat-unicast traffic the network quiets down as paths are learned; under mobility, link churn, or heavy group/broadcast traffic it can get louder again because of re-flooding.

# How Path Discovery Works: Step by Step

1. **First message to an unknown destination** - MeshCore floods the network to locate the target node. The flooded message carries the payload, and every node relays it until it reaches the destination, recording the path it travelled as a byproduct.
2. **Destination returns a path record** - The destination sends back a `PAYLOAD_TYPE_PATH` packet containing the recorded path (not merely a generic ACK).
3. **Sender stores the path** - The original sender records the returned path and uses it ( `ROUTE_TYPE_DIRECT`, embedding the path) for subsequent messages. Relay nodes forward based on the path embedded in each packet; they do **not** maintain per-destination route tables. This is the key step: the sender has now "learned" a path.
4. **Subsequent messages use the established path** - Only nodes on the known route retransmit. All other nodes stay silent.
5. **Retry and re-discover** - After 3 consecutive failed retries on a known path, MeshCore discards the cached route and floods again to find a new one.

# Group and Public Channel Messages

Group messages and public channel broadcasts **always flood** the network, because they are addressed to multiple destinations and no single path can serve all recipients. Path caching only applies to direct (unicast) messages.

# Path Hash Mode

The `path.hash.mode` setting controls the path-hash size used in a node's own advert broadcasts (on a repeater it affects only its advert broadcasts, not which packets it forwards). On companion nodes, the message path-hash size is set in the app's Experimental Settings.

```
set path.hash.mode 0 # 1-byte path hash (default; low overhead)
set path.hash.mode 1 # 2-byte path hash (balanced)
set path.hash.mode 2 # 3-byte path hash (highest precision)
```

A larger hash reduces the chance of path collision in dense networks at the cost of slightly larger packet headers. The firmware default is mode 0 (1 byte); only raise it after confirming the whole network runs firmware that supports the larger size.

# Advertisement Broadcasts

Nodes periodically broadcast advertisements so neighbors can discover them. The flood-advert interval is configurable (in hours):

```
set flood.advert.interval 6 # broadcast every 6 hours (default is 12)
```

To trigger an immediate advertisement (useful after changing location or name), use `advert` for a flood advert, or `advert.zerohop` for a zero-hop (neighbors-only) advert:

```
advert
```

---

Revision #3
Created 2026-05-03 03:00:17 UTC by Mesh America Admin
Updated 2026-06-09 14:31:45 UTC by Mesh America Admin