MeshCore Packet Format Reference

Packet Format

This document describes the MeshCore packet format.

- `0xYY` indicates `YY` in hex notation.
- `0bYY` indicates `YY` in binary notation.
- Bit 0 indicates the bit furthest to the right: `0000000X`
- Bit 7 indicates the bit furthest to the left: `X0000000`

Version 1 Packet Format

This is the protocol level packet structure used in MeshCore firmware v1.12.0

```
[header][transport_codes(optional)][path_length][path][payload]
```

- header - 1 byte
- 8-bit Format: `0bVVPPPPRR` - `V=Version` - `P=PayloadType` - `R=RouteType`
- Bits 0-1 - 2-bits - Route Type
- `0x00` / `0b00` - `ROUTE_TYPE_TRANSPORT_FLOOD` - Flood Routing + Transport Codes
- `0x01` / `0b01` - `ROUTE_TYPE_FLOOD` - Flood Routing
- `0x02` / `0b10` - `ROUTE_TYPE_DIRECT` - Direct Routing
- `0x03` / `0b11` - `ROUTE_TYPE_TRANSPORT_DIRECT` - Direct Routing + Transport Codes
- Bits 2-5 - 4-bits - Payload Type
- `0x00` / `0b0000` - `PAYLOAD_TYPE_REQ` - Request (destination/source hashes + MAC)
- `0x01` / `0b0001` - `PAYLOAD_TYPE_RESPONSE` - Response to `REQ` or `ANON_REQ`
- `0x02` / `0b0010` - `PAYLOAD_TYPE_TXT_MSG` - Plain text message
- `0x03` / `0b0011` - `PAYLOAD_TYPE_ACK` - Acknowledgment
- `0x04` / `0b0100` - `PAYLOAD_TYPE_ADVERT` - Node advertisement
- `0x05` / `0b0101` - `PAYLOAD_TYPE_GRP_TXT` - Group text message (unverified)
- `0x06` / `0b0110` - `PAYLOAD_TYPE_GRP_DATA` - Group datagram (unverified)
- `0x07` / `0b0111` - `PAYLOAD_TYPE_ANON_REQ` - Anonymous request
- `0x08` / `0b1000` - `PAYLOAD_TYPE_PATH` - Returned path
- `0x09` / `0b1001` - `PAYLOAD_TYPE_TRACE` - Trace a path, collecting SNR for each hop
- `0x0A` / `0b1010` - `PAYLOAD_TYPE_MULTIPART` - Packet is part of a sequence of packets

- `0x0B` / `0b1011` - `PAYLOAD_TYPE_CONTROL` - Control packet data (unencrypted)
- `0x0C` / `0b1100` - reserved
- `0x0D` / `0b1101` - reserved
- `0x0E` / `0b1110` - reserved
- `0x0F` / `0b1111` - `PAYLOAD_TYPE_RAW_CUSTOM` - Custom packet (raw bytes, custom encryption)
- Bits 6-7 - 2-bits - Payload Version
- `0x00` / `0b00` - v1 - 1-byte src/dest hashes, 2-byte MAC
- `0x01` / `0b01` - v2 - Future version (e.g., 2-byte hashes, 4-byte MAC)
- `0x02` / `0b10` - v3 - Future version
- `0x03` / `0b11` - v4 - Future version
- `transport_codes` - 4 bytes (optional)
- Only present for `ROUTE_TYPE_TRANSPORT_FLOOD` and `ROUTE_TYPE_TRANSPORT_DIRECT`
- `transport_code_1` - 2 bytes - `uint16_t` - calculated from region scope
- `transport_code_2` - 2 bytes - `uint16_t` - reserved
- `path_length` - 1 byte - Encoded path metadata
- Bits 0-5 store path hash count / hop count (`0-63`)
- Bits 6-7 store path hash size minus 1
- `0b00`: 1-byte path hashes
- `0b01`: 2-byte path hashes
- `0b10`: 3-byte path hashes
- `0b11`: reserved / unsupported
- `path` - `hop_count * hash_size` bytes - Path to use for Direct Routing or flood path tracking
- Up to a maximum of 64 bytes, defined by `MAX_PATH_SIZE`
- Effective byte length is calculated from the encoded hop count and hash size, not taken directly from `path_length`
- v1.12.0 firmware and older only handled legacy 1-byte path hashes and dropped packets whose path bytes exceeded 64 bytes
- `payload` - variable length - Payload Data
- Up to a maximum 184 bytes, defined by `MAX_PACKET_PAYLOAD`
- Generally this is the remainder of the raw packet data
- The firmware parses this data based on the provided Payload Type
- v1.12.0 firmware and older drops packets with `payload` sizes larger than 184

Packet Format

Field	Size (bytes)	Description
header	1	Contains routing type, payload type, and payload version
transport_codes	4 (optional)	2x 16-bit transport codes (if ROUTE_TYPE_TRANSPORT_*)
path_length	1	Encodes path hash size in bits 6-7 and hop count in bits 0-5
path	up to 64 (`MAX_PATH_SIZE`)	Stores `hop_count * hash_size` bytes of path data if applicable

Field	Size (bytes)	Description
payload	up to 184 (`MAX_PACKET_PAYLOAD`)	Data for the provided Payload Type

> NOTE: see the Payloads documentation for more information about the content of specific payload types.

Header Format

Bit 0 means the lowest bit (1s place)

Bits	Mask	Field	Description
0-1	`0x03`	Route Type	Flood, Direct, etc
2-5	`0x3C`	Payload Type	Request, Response, ACK, etc
6-7	`0xC0`	Payload Version	Versioning of the payload format

Route Types

Value	Name	Description
`0x00`	`ROUTE_TYPE_TRANSPORT_FLOOD`	Flood Routing + Transport Codes
`0x01`	`ROUTE_TYPE_FLOOD`	Flood Routing
`0x02`	`ROUTE_TYPE_DIRECT`	Direct Routing
`0x03`	`ROUTE_TYPE_TRANSPORT_DIRECT`	Direct Routing + Transport Codes

Path Length Encoding

`path_length` is not a raw byte count. It packs both hash size and hop count:

Bits	Field	Meaning
0-5	Hop Count	Number of path hashes (`0-63`)
6-7	Hash Size Code	Stored as `hash_size - 1`

Hash size codes:

Bits 6-7	Hash Size	Notes
`0b00`	1 byte	Legacy / default mode
`0b01`	2 bytes	Supported in current firmware

Bits 6-7	Hash Size	Notes
`0b10`	3 bytes	Supported in current firmware
`0b11`	4 bytes	Reserved / invalid

Examples:

- `0x00`: zero-hop packet, no path bytes
- `0x05`: 5 hops using 1-byte hashes, so path is 5 bytes
- `0x45`: 5 hops using 2-byte hashes, so path is 10 bytes
- `0x8A`: 10 hops using 3-byte hashes, so path is 30 bytes

Payload Types

Value	Name	Description
`0x00`	`PAYLOAD_TYPE_REQ`	Request (destination/source hashes + MAC)
`0x01`	`PAYLOAD_TYPE_RESPONSE`	Response to `REQ` or `ANON_REQ`
`0x02`	`PAYLOAD_TYPE_TXT_MSG`	Plain text message
`0x03`	`PAYLOAD_TYPE_ACK`	Acknowledgment
`0x04`	`PAYLOAD_TYPE_ADVERT`	Node advertisement
`0x05`	`PAYLOAD_TYPE_GRP_TXT`	Group text message (unverified)
`0x06`	`PAYLOAD_TYPE_GRP_DATA`	Group datagram (unverified)
`0x07`	`PAYLOAD_TYPE_ANON_REQ`	Anonymous request
`0x08`	`PAYLOAD_TYPE_PATH`	Returned path
`0x09`	`PAYLOAD_TYPE_TRACE`	Trace a path, collecting SNR for each hop
`0x0A`	`PAYLOAD_TYPE_MULTIPART`	Packet is part of a sequence of packets
`0x0B`	`PAYLOAD_TYPE_CONTROL`	Control packet data (unencrypted)
`0x0C`	reserved	reserved
`0x0D`	reserved	reserved
`0x0E`	reserved	reserved
`0x0F`	`PAYLOAD_TYPE_RAW_CUSTOM`	Custom packet (raw bytes, custom encryption)

Payload Versions

Value	Version	Description
`0x00`	1	1-byte src/dest hashes, 2-byte MAC

Value	Version	Description
`0x01`	2	Future version (e.g., 2-byte hashes, 4-byte MAC)
`0x02`	3	Future version
`0x03`	4	Future version

Revision #1
Created 2026-05-03 05:05:14 UTC by Mesh America Admin
Updated 2026-05-03 05:05:14 UTC by Mesh America Admin