MeshCore vs Meshtastic: Technical Comparison

- Protocol Comparison Reference

Protocol Comparison Reference

This page provides a technical comparison between MeshCore and Meshtastic - the two most widely deployed open-source LoRa mesh networking platforms. Both run on similar hardware and serve similar goals, but make very different design choices.

Feature Comparison Table

Feature	MeshCore	Meshtastic
Routing	Hybrid flood-first / direct-route-after: the first message to an unknown destination is flooded (carrying the payload), and the path it took is recorded as a byproduct; subsequent messages are sent directly along that learned path. There is no separate route-request/route-establishment phase.	Controlled (managed) flooding with hop limit and duplicate suppression. Most roles rebroadcast, but rebroadcast depends on role and rebroadcast mode (e.g. CLIENT_MUTE does not rebroadcast). See *meshtastic.org/docs/overview/mesh-algo/*.
Encryption	Channel traffic: AES-128 in ECB mode with a 2-byte (16-bit) truncated HMAC-SHA256 MAC (encrypt-then-MAC, MAC prepended). Direct messages use a per-pair shared secret from ECDH (Ed25519 identity keys transposed to X25519). **Note:** ECB mode leaks repeated/structured plaintext blocks and the 16-bit MAC is forgeable by an active attacker (~32k attempts); key length (128 vs 256) is not the main security difference. Keys are static, so there is no forward secrecy.	AES-256-CTR with a shared PSK per channel.
Key Exchange	ECDH using each node's Ed25519 identity keypair transposed to X25519/Curve25519; each node pair derives a unique 32-byte shared secret (AES-128 uses 16 bytes of it). Keys are static (no forward secrecy or key revocation).	Static pre-shared key (PSK) distributed out-of-band; no per-pair key agreement for channels

Direct messages	End-to-end encrypted using a per-pair ECDH-derived key (AES-128-ECB with a 2-byte MAC and static keys, so no forward secrecy).	End-to-end encrypted via X25519 ECDH + AES-CCM (introduced with PKI direct messaging in recent firmware; verify the exact version against meshtastic.org).
Infrastructure role	Explicit firmware types: Companion (BLE or USB serial), Repeater, Room Server, and Sensor.	Router/Repeater/Client/Tracker among ~12 device roles in current firmware (some, e.g. REPEATER, are deprecated); the exact count varies by firmware version. As of 2026; see meshtastic.org device-config Roles.
Node discovery	Advertisement packets (flood or zero-hop)	NodeInfo broadcast flood
Position sharing	In advertisements (optional)	Continuous broadcast to channel (configurable interval)
Scalability	Better at high node counts due to path-based unicast reducing channel utilization	Best under ~100 nodes; flooding overhead grows with network size
Network mapping	App shows routing topology; community map at meshcore.co.uk/map.html	meshmap.net aggregates public data
Message storage	Room servers (store-and-forward)	Store and Forward module (node-based)
App ecosystem	MeshCore app (iOS/Android)	Meshtastic app (iOS/Android/web)
Web interface	config.meshcore.io (config). Community-run interfaces (e.g. app.meshcore.nz) also exist and may change. As of 2026.	client.meshtastic.org
Firmware update	Web flasher (flasher.meshcore.io) over USB. OTA updates are also supported on nRF52 devices via the `start ota` command.	Web flasher + OTA via app
Primary hardware	T114, RAK4631, T-Beam v1.2+ and similar. SX126x/LR11xx radios are strongly preferred, but SX127x (SX1276) is also supported in current firmware (limited build variants); the real constraint on older ESP32 boards is MCU/flash size, not the radio chip.	All of the above + many more (supports SX1276, SX1262, and others)
License	Open source, MIT (upstream: github.com/ripplebiz/MeshCore; community fork: github.com/meshcore-dev/MeshCore)	Open source (github.com/meshtastic)

When to Choose MeshCore

- Building dedicated network infrastructure - repeaters on towers, rooftops, or hilltops where path-based routing reduces channel congestion.
- Your community already uses MeshCore and you need to integrate with an existing deployment.
- You want stronger per-pair direct message encryption - ECDH per-pair keys provide better isolation than a shared channel PSK.
- Deploying a large-scale network (100+ nodes) where flooding creates significant channel congestion.

When to Choose Meshtastic

- You need the widest hardware compatibility - Meshtastic has the largest catalog of supported boards. (Note: MeshCore now also supports some SX127x/SX1276 boards in current firmware, so SX1276 is no longer MeshCore-incompatible.)
- You need WiFi/MQTT bridging for internet-connected nodes.
- Your community or region already has an established Meshtastic network.
- You need TAK/ATAK integration or other Meshtastic-specific integrations.
- You prefer a larger community and more third-party tooling.

Sources: MeshCore packet format documentation (github.com/meshcore-dev/MeshCore), Meshtastic documentation (meshtastic.org), Meshtastic protobufs (github.com/meshtastic/protobufs)