Advanced Meshtastic Repeater Topics

Store and Forward, position reporting, and telemetry for infrastructure repeater nodes.

- Store and Forward
- Position and Telemetry for Infrastructure Nodes

Store and Forward

What Is Store and Forward?

Store and Forward (S&F) is a Meshtastic server module that **buffers text messages for nodes that are temporarily offline**. When a client node comes back within range, the S&F server can replay text messages the client missed while it was away.

Important - the buffer is volatile and S&F is not a delivery guarantee. Stored messages live only in the server node's PSRAM. Any reboot, brownout, or power loss on the server node erases ALL stored messages with no recovery. Replay is best-effort, request-driven, and limited by count and time window; it covers text messages only (not telemetry, position, or other packet types). Never treat the buffer as durable storage, and never rely on it for life-safety message assurance - for incident use, assume the buffer can vanish at any moment.

How It Works

1. A node with S&F enabled in *server mode* listens to text messages on its primary channel (channel 0) and stores them in a ring buffer in PSRAM. Note: history retrieval over LoRa is **not available on the default public channel**, so the server must be configured with a non-default primary channel for S&F history to work.
2. When a client node requests history (see Client Configuration below), it sends a replay request.
3. The server transmits buffered messages up to the configured **History Return Max** count and within the **History Return Window** time range.

Hardware Requirements

The S&F server requires an ESP32 device **with PSRAM** - nRF52-based boards cannot run the Store-and-Forward server module at all:

- **Recommended (ESP32-with-PSRAM only):** T-Beam (v1.0 and later), LilyGo T3S3, or other ESP32-S3 devices that include PSRAM.
- **Not supported as an S&F server:** nRF52840-based devices (RAK4631, Nordic development kits) and any board without PSRAM. These cannot act as a Store-and-

Forward server.

Server Configuration

1. Navigate to **Config → Module Config → Store and Forward**.
2. Set **Enabled → true**.
3. Set **Is Server → true**.
4. Set **History Return Max** - the maximum number of messages to replay per request (default: 25 messages, which is the value used when this is left at 0).
5. Set **History Return Window** - the time window in minutes from which to replay messages (default 0 = 240 minutes / 4 hours).
6. Save and reboot the node.

Client Configuration

A client node does **not** need the Store and Forward module enabled to request history. To retrieve missed messages:

1. Set role to **CLIENT** (or **CLIENT_MUTE**, labelled "Client Mute" in the app menu, if the device should not rebroadcast).
2. Request history manually: on Android, send a direct message containing **SF** to the server node; on Apple, use the **Client History** option.

History retrieval over LoRa is manually requested as above. Automatic retrieval happens only when an app connects directly to the server node (firmware 2.4+) - it does not happen automatically over the mesh just by coming into range.

Ideal Use Cases

- Hikers or cyclists who move in and out of mesh coverage.
- Vehicles that periodically leave and re-enter a covered area.
- Remote monitoring stations that reconnect on a schedule.
- Community nets that want a convenience replay of recent text traffic - but note S&F is best-effort only and must never be relied on as guaranteed delivery for emergency or life-safety messages.

Limitations

- S&F history retrieval over LoRa is **not available on the default public channel**; you must configure a non-default primary channel to use it. S&F operates only on the primary channel (channel 0) - if your net runs on a private or secondary channel, store-and-forward will not buffer or replay that traffic. Verify replay actually works on YOUR channel before depending on it.
- Text messages only - telemetry, position, and other packet types are not buffered or replayed.
- Large message volumes will cause the ring buffer to overwrite older messages quickly.
- **The buffer lives in PSRAM only.** The server node must be continuously powered and on-mesh; any reboot, brownout, or power loss erases ALL stored messages with no recovery.
- S&F adds overhead to the server node; monitor channel utilisation on high-traffic networks.

Position and Telemetry for Infrastructure Nodes

Why Position Accuracy Matters

An accurate position lets your repeater appear correctly on meshmap.net and in the Meshtastic app's node list. (Note that meshmap.net is a third-party map and only shows nodes reporting to the public MQTT server, so it is not an authoritative or complete view of the mesh.) Other operators use your node's reported position to plan coverage, model signal paths, and verify that packets are actually being relayed from the expected location.

Fixed Position vs. GPS

Most unattended infrastructure repeaters do not need GPS hardware. Instead, configure a **fixed position** using the known coordinates of the deployment site:

1. Look up the coordinates of your site with any map application (Google Maps, OSMand, etc.) - right-click the location and copy the lat/lon.
2. In the Meshtastic app: **Config → Position → Fixed Position → Enable**.
3. Enter the **latitude**, **longitude**, and **altitude** (in metres).
4. Save. The node will broadcast this position at the configured interval without needing a GPS fix.

If your device has GPS hardware and the repeater is mobile (e.g. a vehicle relay), leave GPS enabled and set the Smart Position Broadcast threshold appropriate for your speed.

Position Broadcast Interval

This setting controls how often the node announces its location on the mesh.

- **Default:** 15 minutes (with Smart Broadcast enabled). Smart Broadcast increases the frequency when the node is moving.

- **Recommended for static infrastructure:** 1 hour or longer (`3600` seconds and up); for a permanently fixed node, many operators use 12 - 24 hours (`43200` - `86400` seconds). A repeater that never moves does not need to announce its position frequently.
- Increasing (lengthening) this interval lowers airtime consumption and channel utilisation - important on busy networks.

Smart Position Broadcast

Meshtastic's Smart Position feature broadcasts a new position only when the node has moved beyond a configurable distance threshold. **For a static repeater, disable Smart Position and use a fixed timed interval instead** - Smart Position is designed for moving nodes and may behave unexpectedly on hardware without GPS.

Device Telemetry

Meshtastic can broadcast device health metrics over the mesh. The device metrics are:

- Battery voltage and charge percentage.
- Channel utilisation (ChUtil) - the percentage of time the radio channel is occupied by transmissions (this node's own transmissions plus traffic it senses from others).
- Air utilisation for transmit (AirUtilTX) - the percentage of airtime this node itself spends transmitting.

Enable and configure telemetry under **Module Config → Telemetry**, in the **Device Metrics** section (exact menu labels vary by client and version). Set the broadcast interval - the default of `1800` seconds (30 minutes) is appropriate for most infrastructure nodes; lengthen it to reduce airtime.

Remote Health Monitoring

Once device telemetry is enabled, any operator who can see your node on the mesh can view its reported battery voltage and channel utilisation in the Meshtastic app's node detail view. This provides passive, no-cost status visibility:

- A gradual drop in reported battery voltage warns of a failing solar charge or depleted battery before the node goes offline.
- High channel utilisation (above ~25% ChUtil, or AirUtilTX above ~7 - 8%) indicates congestion - consider lengthening broadcast intervals or relocating the node.
- Uptime resets alert you to unexpected reboots (power glitches, firmware crashes).

Passive telemetry shows health only while the node is alive and in range. It provides **no active alert when the node actually fails** - you simply stop seeing updates, which is easy to miss. For infrastructure you depend on, pair telemetry with an active offline-detection alert (something that notifies you when telemetry stops).

For critical infrastructure nodes, consider setting up a **MQTT bridge** to push telemetry to an external monitoring system - see the Meshtastic MQTT documentation for details. Note that Wi-Fi/MQTT-based monitoring depends on internet connectivity and will fail in exactly the grid-down or internet-out emergencies you may most need it.