Firmware Updates on Deployed Repeaters

Meshtastic has no over-the-LoRa firmware push between mesh nodes - firmware cannot be sent from one node to another over the radio mesh. Firmware is updated via a USB connection to the device, or via Bluetooth OTA on supported nRF52 devices (e.g. RAK4631). For a repeater deployed on a rooftop, tower, or remote site, this generally means a site visit. Planning these maintenance windows carefully - and documenting configuration before you go - prevents accidental service outages and preserves network continuity for the community.

Over-the-Air Firmware Update: Not Over LoRa

It is worth stating clearly: there is no mechanism in Meshtastic firmware to push a firmware binary from one mesh node to another over LoRa. The "OTA" term in Meshtastic documentation means *Bluetooth OTA* - a supported firmware update path over BLE from a smartphone app for nRF52 devices, not a transfer over the LoRa mesh. BLE OTA is a documented, first-class update path on nRF52 boards (such as the RAK4631), not a stray term.

USB flashing is the most universally supported and reliable path across all platforms; nRF52 devices additionally support reliable BLE OTA, which is often the only practical path for a sealed enclosure. Plan accordingly when choosing repeater sites - a location that requires climbing a locked tower or a two-hour drive is a location you will want to update infrequently, so prioritise stability over bleeding-edge firmware on remote infrastructure nodes.

Planning a Maintenance Window

- **Notify the community** before updating a widely-used repeater. Post in your community channel or MQTT-linked group chat at least 24 hours in advance. Include the expected downtime window and the node name/ID.

- **Schedule during low-usage periods** - typically overnight or early morning on a weekday.
- **Check release notes** at `github.com/meshtastic/firmware/releases` before updating. Look for breaking changes in configuration format, channel encoding, or protocol version that could prevent the updated node from communicating with older clients.
- **Do not update all infrastructure nodes simultaneously**. Update one node, confirm it reconnects to the mesh and interoperates with other nodes, then proceed to the next.
- **If a site has only ONE repeater serving an area, treat its update as a planned outage.** Schedule it when the network is not needed, notify users that the path will be DOWN, and have a pre-flashed known-good spare or the prior firmware binary on hand to roll back on-site if the flash fails (a failed flash can leave the device in a crash loop). Never update a sole-path repeater remotely or without a rollback plan. Note that a "same-site second node" is not true redundancy against site power loss, lightning, or fire.

Exporting Configuration Before the Update

Configuration is not preserved across all firmware versions. Some updates change protobuf field IDs or introduce new mandatory fields, causing the stored configuration to load with default values after the firmware wipe. Always export before touching the device.

```
# Export current configuration to a YAML file
meshtastic --export-config > repeater_config_backup_$(date +%Y%m%d).yaml
```

This produces a YAML file with all channel configurations, device settings, LoRa settings, position, and module config. Store this file in a safe location - version control or your operations shared drive. (Note: `--info` output is *not* a restorable backup; only the `--export-config` YAML can be re-applied with `--configure`.)

Review the exported file before the update and note any values that are not visible in the app UI (e.g. custom channel PSK keys, non-default hop limits, MQTT credentials). These are the settings most likely to need manual restoration if automatic re-import fails.

Flashing Firmware via the Web Flasher

The Meshtastic web flasher at `flasher.meshtastic.org` is the recommended tool for **ESP32** hardware platforms (T-Beam, Heltec, etc.). It runs entirely in a Chromium-based browser and handles partition layout, bootloader, and firmware in a single operation. **nRF52 boards (RAK4631, T-Echo) are not flashed with the web serial flasher** - they update by drag-and-dropping a UF2 file onto the device's mass-storage drive after entering bootloader mode (double-tap reset), or via BLE OTA.

1. Connect the repeater hardware to your laptop via USB. Use a data-capable cable - power-only cables will not expose a serial port.
2. Navigate to `flasher.meshtastic.org` in **Chrome or Edge** (these are the officially supported browsers; other Chromium browsers such as Brave are based on the same engine and may work, but Web Serial support is not guaranteed).
3. Click **Flash Connected Device**. The browser will request access to the serial port - select the device from the list (typically `CP2102`, `CH340`, or `FTDI` depending on the board).
4. Select the correct hardware variant. Flashing the wrong variant can result in a non-booting device, so **always confirm the exact variant string against the live device dropdown on flasher.meshtastic.org before flashing** - variant labels can change between releases. Common examples (verify before use):
   - T-Beam v1.1 → `tbeam`
   - T-Beam v1.2 / AXP2101 → `tbeam-s3-core` (the AXP2101 PMU appears on more than one board; confirm your board's exact variant in the device list)
   - Heltec v3 → `heltec-v3`
   - RAK WisBlock 4631 → `rak4631` (flashed via UF2 drag-and-drop, not the web serial flasher)
5. Select the firmware version. For infrastructure nodes, prefer the latest stable release rather than alpha/nightly builds.
6. Click **Flash** and wait for the progress bar to complete. Do not disconnect the cable during flashing.

Alternative: CLI / esptool flashing

The web flasher (flasher.meshtastic.org) is the supported path for ESP32 devices. A command-line flashing route using `esptool` is documented in the official "Flashing Firmware" guide. The standalone `meshtastic-flasher` GUI project is legacy/unmaintained in favour of the web flasher; if you use any CLI tool, verify its current syntax against the official docs rather than relying on a fixed command line, as flags change between versions.

Restoring Configuration After the Update

```
# Restore a previously exported configuration (YAML) with --configure
# (there is no --import-config flag)
meshtastic --configure repeater_config_backup_20250101.yaml
```

After restoring, verify critical settings manually:

```
# Verify device role
meshtastic --get device.role

# Verify region (master legal control for band and power cap)
meshtastic --get lora.region

# Verify channel 0 PSK matches your network
meshtastic --info | grep -A 3 "channel_0"

# Verify MQTT settings
meshtastic --get mqtt

# Verify fixed position
meshtastic --get position

# Verify hop limit
meshtastic --get lora.hop_limit
```

Reboot the device after restoring configuration to ensure all settings take effect from a clean state:

```
meshtastic --reboot
```

Version Compatibility: What to Check Before Updating

Meshtastic nodes rebroadcast packets based on matching radio parameters even when they cannot fully decode newer packet types (managed flooding rebroadcasts on radio params, not on decode success). There is no formal version-negotiation handshake, so nodes running very different firmware versions may fail to decode some packet types from each other. Specific areas to check in release notes:

- **Protocol buffer schema changes** - new required fields in existing message types can cause older firmware to reject packets from a newly updated node.
- **Channel encryption format changes** - rare but occurred between some major version transitions. If your channel uses a custom PSK, test packet delivery after updating one node before updating others.
- **Modem preset changes** - some presets have been deprecated over time (e.g. `LONG_SLOW`, `VERY_LONG_SLOW`), but existing modem-preset enum IDs are stable and are not renumbered or reordered (`LONG_FAST` has always been ID 0). New presets are appended rather than reused, so a given preset name maps to the same setting across versions. Still verify the active modem preset by name after updating in case a default changed.

```
# Verify active modem preset after firmware update
meshtastic --get lora.modem_preset
```

Post-Update Validation Checklist

1. Device boots and appears in Meshtastic app node list.
2. Device role is correct. For infrastructure, use `ROUTER` (or `ROUTER_LATE` for nodes that should rebroadcast only after others). Note that `ROUTER_CLIENT` was deprecated in v2.3.15 and `REPEATER` is deprecated as of ~v2.7.11, so prefer `ROUTER` for new and updated deployments.
3. **LoRa region is set correctly for your location** (`meshtastic --get lora.region`). A firmware update that resets the region to UNSET or changes a default can cause out-of-band or wrong-power transmission - the region is the master legal control for both your authorized band and your power cap, so always confirm it after updating.
4. Channel 0 (primary) PSK matches community configuration.
5. Fixed position is present and correct on map.
6. MQTT uplink is publishing telemetry (check Grafana or broker).
7. Hop limit is correct for your network.
8. Position broadcast interval is set to a long static-node value (e.g. 43200 or 86400 seconds), not the default short interval.
9. At least one other mesh node can receive packets from the updated repeater.

Only after this checklist passes should you remove the USB cable and close up the enclosure.

Revision #7
Created 2026-05-03 05:50:07 UTC by Mesh America Admin
Updated 2026-06-09 00:23:22 UTC by Mesh America Admin