Network Configuration Settings

The Network configuration section (**Config > Network**) controls how ESP32-based Meshtastic nodes connect to IP networks - WiFi and Ethernet - and associated services like NTP and remote logging. These settings are only relevant for ESP32 hardware; nRF52-based boards (like the RAK4631) do not support WiFi natively and these settings have no effect on them, with the exception of Ethernet on RAK builds fitted with a RAK13800 module.

Access these settings in the Meshtastic app under **Settings > Radio Configuration > Network**, or via the Python CLI with `meshtastic --set network.*`.

Note: Network connectivity unlocks important Meshtastic features: the web interface, MQTT gateway, APRS bridging, NTP time synchronization, and remote syslog. If you are using ESP32-based hardware (T-Beam, T-Lora, Heltec WiFi LoRa, Station G2, etc.), understanding these settings is important for gateway and infrastructure deployments.

WiFi SSID

Config key: `network.wifi_ssid`
Default: Empty (WiFi disabled)

The SSID (network name) of the WiFi network the node should connect to as a client. Case-sensitive. Maximum 32 characters.

To enable WiFi: Set both WiFi SSID and WiFi Password. The node will attempt to join the network on boot and reconnect if the connection drops.

Important considerations:

- Enabling WiFi disables Bluetooth - only one connection method works at a time on ESP32 devices. Enabling WiFi also increases power consumption noticeably compared to running LoRa with Bluetooth only.
- A node with WiFi enabled and a good internet connection can serve as an MQTT gateway, APRS-IS gateway, and NTP time source for the mesh
- On battery-powered mobile nodes, WiFi should generally be disabled to preserve battery life. Enable WiFi on mains-powered infrastructure nodes.

WiFi Password

Config key: `network.wifi_psk`
Default: Empty

The passphrase for the WiFi network specified by WiFi SSID (max 64 characters). Stored in the device's flash memory. Supports open networks (leave password empty if the network has no password, though this is strongly discouraged for security reasons).

Security note: The WiFi password is stored in plaintext in the device's NVS (Non-Volatile Storage) flash partition. Anyone with physical access to the device and a serial connection can potentially extract it. Do not configure a node with your primary home WiFi password on a device that could be physically compromised; consider using a dedicated IoT VLAN or guest network.

WiFi Mode

ESP32-based Meshtastic nodes connect to WiFi in **client (station / STA) mode only**. The firmware does **not** support SoftAP (Access Point) mode - there is no setting to make the node create its own WiFi access point, and there is no `network.wifi_mode` config key. WiFi is controlled simply by whether a WiFi SSID and password are set.

In client mode the node connects to an existing WiFi network as a client (station). This is the standard mode for gateway nodes that need internet access. In client mode, the node:

- Obtains an IP address via DHCP (or a configured static IP)
- Can reach the internet for MQTT, NTP, APRS-IS, and other services
- Is accessible on the local network at its assigned IP address
- Serves the Meshtastic web client at `http://<node-ip>/` (or via the mDNS alias `http://meshtastic.local/`)

No phone-direct AP option: Because SoftAP mode is not supported, you cannot connect a phone or laptop directly to the node's own WiFi. For field configuration without an existing WiFi network, use Bluetooth or a USB serial connection instead.

Ethernet Enabled

Config key: `network.eth_enabled`
Default: `false`

Enables the Ethernet interface on hardware that supports it. The documented Ethernet reference hardware is the **RAK4631 paired with the RAK13800 Ethernet module** (a W5500-class SPI Ethernet controller). Note that JSON MQTT output is not supported on the nRF52 platform.

Advantages of Ethernet over WiFi for infrastructure nodes:

- More reliable and stable connection - no RF interference, no re-association delays
- Lower and more consistent latency
- Generally lower and steadier power consumption than active WiFi
- No PSK to manage or expose

Use when: Deploying a fixed infrastructure node (ROUTER or gateway) at a location with Ethernet infrastructure - server room, communications closet, network rack. Ethernet-connected gateway nodes are more reliable than WiFi-connected ones for long-term unattended operation.

NTP Server

Config key: `network.ntp_server`
Default: `meshtastic.pool.ntp.org`

The hostname or IP address of the NTP (Network Time Protocol) server the node uses to synchronize its real-time clock when IP networking is available. Accurate time is important for:

- Correct message timestamps displayed in the app
- Log entries with accurate timestamps for troubleshooting
- MQTT message timestamps
- Coordinating time-dependent operations across the mesh

Meshtastic nodes without internet connectivity rely on time received from other nodes on the mesh (nodes share time information in packets). An NTP-synced node improves its own clock accuracy; that time can then propagate to other nodes via the normal packet exchange, but the firmware docs do not promise that a single NTP node becomes an authoritative time source for the entire mesh.

Recommended NTP Servers

Server	Description	Use When
`meshtastic.pool.ntp.org`	Default NTP pool used by the firmware	General use, internet-connected nodes

Server	Description	Use When
`time.cloudflare.com`	Cloudflare NTP (anycast, fast)	Reliable alternative with good global coverage
`time.google.com`	Google Public NTP	Reliable alternative
`time.nist.gov`	NIST time server	When US government standard time is needed
`192.168.x.x` (local)	Your own local NTP server	Isolated networks, high-accuracy requirements, no internet

Local NTP server: If your deployment has a local NTP server (common in enterprise and government networks, and in some emergency operations centers), set this to that server's address. This reduces internet dependency and may improve synchronization accuracy.

NTP without internet: If the node has no internet access but is on a local network with a router that provides NTP (most home routers do), using the router's IP address as the NTP server works well: `192.168.1.1` or similar.

---

rsyslog Server

Config key: `network.rsyslog_server`
Default: Empty (disabled)

Configures remote syslog logging. When set to a hostname or IP address (with optional port, e.g., `192.168.1.100:514`), the node sends its log output to a remote syslog server over UDP using the standard syslog protocol (RFC 3164/5424).

Why use remote logging:

- Centralized log collection from multiple nodes - see all infrastructure logs in one place
- Persistent log storage - node logs that would otherwise be lost on reboot are captured on the server
- Real-time alerting - syslog servers can trigger alerts on specific log messages (errors, reconnections, etc.)
- Troubleshooting unattended nodes - diagnose issues without physically connecting a serial cable

Setup requirements:

- A syslog server running on the local network (rsyslog, syslog-ng, Graylog, or any standard syslog receiver)
- The node and syslog server must be on the same network (or routable to each other)

- UDP port 514 is the IANA standard and conventional default syslog port, but the port is configurable per the docs

Recommended syslog servers for small deployments:

- **rsyslog** on Linux (Raspberry Pi, server): Simple, lightweight, standard
- **Graylog**: Full log management with search and dashboards - good for larger deployments
- **Loki + Grafana**: Modern log aggregation with excellent visualization

Example rsyslog configuration to receive Meshtastic logs on a Linux server:

```
# /etc/rsyslog.d/meshtastic.conf
module(load="imudp")
input(type="imudp" port="514")

# Save Meshtastic logs to a dedicated file
if $fromhost-ip == '192.168.1.50' then /var/log/meshtastic/node1.log
```

Practical Configuration Guidance

Standard Home Gateway Node

For a mains-powered T-Beam or Station G2 acting as a gateway and router at home:

- WiFi SSID: your home network SSID (or use IoT VLAN if available)
- WiFi Password: your network password
- NTP Server: `meshtastic.pool.ntp.org` (default is fine)
- rsyslog Server: empty unless you have a log server

Field Deployment - No Internet

For a node deployed at an event or emergency operation without internet access:

- Leave WiFi SSID empty - there is no AP mode, so configure the node over Bluetooth or USB serial instead
- NTP Server: empty or your EOC's local network NTP if available
- Time will sync from other mesh nodes that have NTP access

Ethernet-Connected Infrastructure

For a fixed ROUTER node on a rack or network closet:

- Ethernet Enabled: true
- WiFi: disabled (leave SSID empty)
- NTP Server: your organization's NTP server or `time.cloudflare.com`
- rsyslog Server: your organization's syslog collector
- Managed Mode: true (set via `security.is_managed true`, with admin keys configured)

Local Web Access

When the node is connected to your WiFi or Ethernet network, you can reach the Meshtastic web client in a browser:

- Open `http://<node-ip>/`, or the mDNS alias `http://meshtastic.local/`
- The node serves the web client over its API (port 4403); this is the web client, not a separate self-hosted admin site
- For configuration when no network is available, use Bluetooth or a USB serial connection

---

Revision #6
Created 2026-05-03 05:55:56 UTC by Mesh America Admin
Updated 2026-06-09 12:04:35 UTC by Mesh America Admin