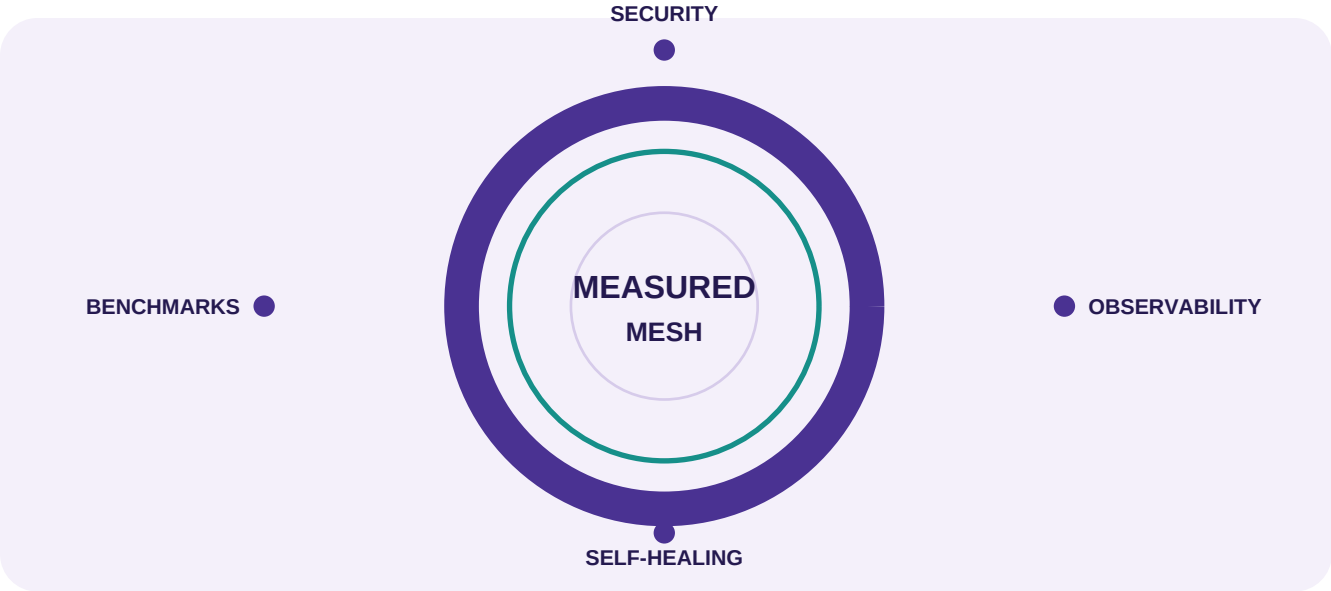
TECHNICAL PAPER / 01

# Intelligent mesh. Measured proof.

Architecture, platform integration and qualification for an MIT-licensed Go control plane on Qualcomm-based T2 devices.




An engineering design and validation plan, not a performance certification. Current evidence includes host tests, static ARM64 builds and T2 capability observations. Full RF qualification, current-build Wi-Fi 7 readback, ten-hop forwarding and standards interoperability remain open.

| | |
|---|---|
| Release target | T2 / AArch64 Qualcomm IPQ firmware |
| Original software | Go userspace control plane, MIT license |
| Platform components | Linux, QCA driver/firmware, hostapd and supplicant retain their own licenses |

Independent WantasticMesh publication. Qualcomm is referenced as the platform vendor; this document does not imply vendor authorship or endorsement.

SYSTEM DESIGN / 02

# Architecture and ownership

Keep policy decisions, wireless operation and forwarding independently observable.

WantasticMesh exchanges authenticated topology and policy messages between a controller and its agents. The installed Wi-Fi stack performs association, encryption, transmission and regulatory enforcement. A private control channel does not itself implement EasyMesh interoperability or a multi-hop forwarding protocol.

### WantasticMesh / MIT Go

Topology • authentication • policy • recovery

### Verified platform adapters

nl80211 • hostapd control • validated vendor IPC

### QCA wireless and forwarding stack

EasyMesh option • AP/STA • driver • firmware

## One owner for each radio action

QCA policy services and WantasticMesh must not independently change the same channel, station steering or transmit power. Integration begins with observation. Actions need explicit ownership, capability checks, deadlines, readback and rollback.

## Forwarding remains a separate choice

AP/STA backhaul, vendor EasyMesh and Linux 802.11s are different integration paths. The observed T2 driver advertises AP and managed modes, but no mesh-point mode. Missing mesh-point capability blocks 802.11s; it does not by itself block AP/STA operation.

## A platform adapter is not a certification

Vendor command FIFOs expose a limited trigger surface. Exact commands, response semantics and failure behavior need verification on each firmware build. Matching package names alone does not establish IEEE 1905 or EasyMesh conformance.

PLATFORM EVIDENCE / 03

# T2 capability baseline

Distinguish a supported build target from a qualified wireless deployment.

| | |
|---|---|
| Architecture / kernel | Observed AArch64, IPQ95xx OpenWrt snapshot and Linux 5.4.213. Exact QSDK release lineage remains unverified. |
| Radio interface modes | AP, AP/VLAN, managed and monitor observed. No mesh-point mode advertised. |
| Vendor components | QCA EasyMesh, IEEE 1905 libraries, hostapd, supplicant and wireless driver packages present. |
| Service state | Latest recorded observation found ezmesh on T2-B; earlier notes recorded it disabled. Fresh state capture is required before migration. |
| ARM64 userspace | Pure-Go static build succeeds. Current adapters still require paired-device RF and long-duration validation. |
| MLO / EHT | Earlier T2 notes record EHT160 and AP mode 11BEA_EHT160 4x4. Current-build EHT readback and active MLO remain to be verified. |
| Vendor candidate | Four supplied IPKs packaged unchanged with hashes and metadata. No driver installation performed. |

## Dependency compatibility gate

The supplied hostapd/supplicant candidate requires libopenssl3; observed firmware contains libopenssl1.1. Kernel packages also require cfg80211-internal and additional QCA modules. A version string of 5.4.213 does not prove matching vermagic, symbol versions or firmware ABI.

## License boundary

Project-authored userspace is MIT licensed. A full-stack distribution contains separately licensed components. Preserve original licenses, notices and applicable source obligations; do not describe the complete vendor stack as MIT.

WIRELESS GENERATION / 04

# Wi-Fi 7: verify the link

EHT should be used when both peers and the platform permit it. Proximity is only one input.

A nearby pair of devices may still operate with 802.11ax/HE if the AP configuration, supplicant, radio mode or firmware does not enable EHT. A 160 MHz channel can carry Wi-Fi 7; 320 MHz operation and MLO are separate capabilities and must not be inferred from an EHT label.

## Historical EHT and the current dashboard

Earlier device notes record EHT160 and AP mode 11BEA_EHT160 4x4. This establishes a historical working configuration, not the present mode. The displayed link is labeled probable, 6595 MHz, 160 MHz, center 6665 MHz, and 802.11ax. Capacity of 480 Mbps is explicitly a MAC-efficiency estimate; the 10 m distance is a path-loss estimate. These values are not measured application goodput or verified range.

| | |
|---|---|
| Capability | Record per-radio EHT support, supported modes, regulatory channels and interface-combination limits. |
| Configuration | Inspect root backhaul BSS and station configuration. Determine whether EHT is enabled or disabled and whether the hostapd/supplicant build supports it. |
| Negotiation | Read station and association status from both peers. Prefer explicit EHT/HE rate metadata over band or channel-width inference. |
| Traffic evidence | Generate repeatable traffic and capture rate, MCS, spatial streams, width, retries, loss and application goodput together. |
| Fallback | Keep a working HE connection if EHT cannot be negotiated; expose the reason or unknown status instead of inventing a Wi-Fi 7 label. |

## Connection readback and close-range benchmark

Run wmeshd backhaul status -if wmbhsta0. The CLI reports explicit supplicant wifi_generation; missing data stays unknown. Use a measured separation and documented antenna arrangement. Keep power and regulatory settings recorded. Compare HE and EHT only when both configurations are supported and confirmed by readback. Report warm-up time and repeated trials; a strong RSSI alone does not prove higher throughput.

RESILIENCE & SECURITY / 05

# Trust, recover, reconcile

Fast recovery must preserve authority, prevent loops and retain useful local service.

## Authenticated control

The control plane uses AES-256-GCM and independently provisioned per-node control keys. Boot-bound keying, sequence handling and challenge correlation protect the exchange against replay. Wireless credentials are separate from the per-node control credentials. Keep secret files restricted and exclude them from benchmark artifacts.

## Bounded state and liveness

The implementation has bounded controller state, retries, discovery backoff and stale-link handling. A software simulation exercises 512 agents and adverse message conditions. This proves selected software behaviors; it does not establish physical radio capacity or multi-hop data-plane behavior.

## Recovery is a data-plane obligation

A controller becoming unreachable must not unnecessarily terminate working local forwarding. Local repair requires a supported forwarding path and an alternate route. A physical partition with no surviving connection cannot carry traffic between its components. Reconnection must reconcile state without stale policy overwrites.

| | |
|---|---|
| Failure injection | Observe and record |
| Controller stopped | Local client continuity, stale policy expiry, reconnection behavior |
| Backhaul blocked | Loss duration, alternate path selection, loops and blackholes |
| Node reboot / partition | Boot identity, replay rejection, topology reconciliation |
| Invalid control input | Authentication rejection, parser bounds, CPU/memory stability |
| Repeated link flaps | Convergence distribution and hysteresis; no endless oscillation |

Security and recovery qualification require adversarial device testing in addition to unit tests. No claim of independent certification or exhaustive security audit is made.

AIRTIME CONTROL / 06

# Adaptive PTMP scheduling

A measured policy for relative station service; deterministic TDMA is a separate research goal.

The implemented PTMP planner samples station traffic and radio counters, computes bounded relative weights and submits them through standard nl80211 when supported. Hostapd and firmware retain association, rate control, aggregation, retransmission and regulatory responsibilities.

| | |
|---|---|
| Inputs | Byte deltas, packet/retry/failure counters, rate, signal, inactivity and available airtime fields |
| Default bounds | Minimum 64; base 256; maximum 1024 |
| Stability controls | 8 percent hysteresis and 25 percent per-epoch slew limit |
| Scale bound | Up to 512 station outputs, normalized independently per AP |
| Actuation | Write station weight; require matching readback; attempt rollback of earlier changes after an epoch failure |
| Device status | T2 enforcement and service improvement remain unverified; feature disabled by default |

## Readback is necessary, not sufficient

A returned weight confirms a control value. It does not prove the firmware enforces the intended scheduling share. Compare observed airtime and per-client goodput under sustained contention, including slow clients, hidden nodes and mixed traffic.

## What strict TDMA would require

Deterministic slots need a supported MAC/firmware timing mechanism, synchronization and guard-time design, admission control, coexistence behavior and measured tail latency. The current weight controller does not provide synchronized microsecond slots or coordinated client clocks.

Implementation detail and operator commands: docs/PTMP-AIRTIME-PAPER.md. Apply only on a test AP with a separate management path and a demonstrated restoration procedure.

BENCHMARK METHODOLOGY / 07

# Prove the replacement

Compare on identical hardware, controlled configurations and reproducible traffic.

Use stock QCA EasyMesh as the primary baseline. Compare any additional licensed baseline only when its build and configuration are available. Record failures and unavailable features explicitly. Never substitute an estimated dashboard capacity for measured traffic.

| | |
|---|---|
| Topology stages | Two-node baseline; four-node diamond for alternate paths; eleven-node chain for ten physical hops. |
| Lab separation | Wired management and traffic endpoints independent of the tested wireless path; record any wired bypass. |
| Traffic matrix | TCP upload/download and bidirectional load; UDP load sweep; small-packet latency under load; concurrent clients. |
| Radio controls | Match channel, width, power, security, antennas, firmware, clients and negotiated HE/EHT mode. Record interference. |
| Run procedure | Proposed initial protocol: 30 s warm-up, 180 s measurement, at least five repetitions per condition. Randomize baseline/candidate order. |
| Metrics | Goodput, loss, jitter, p50/p95/p99 latency, airtime, retries, recovery outage, CPU, RSS and control bytes. |
| Evidence package | Timestamped raw logs, source commit, binary hashes, package manifests, configuration with secrets removed, topology and traffic commands. |

## Acceptance before experimentation

Pre-register limits for throughput regression, latency, loss and recovery before seeing candidate results. Thresholds depend on the client service requirement and baseline variability; no numerical performance threshold has yet been validated for this product.

Report every repetition and uncertainty across runs. Analyze latency distributions with a stated method. A success in software simulation or a two-node lab is not a ten-hop qualification.

DEPLOYMENT & QUALIFICATION / 08

# Release with evidence

A small userspace installer and separately qualified platform packages.

## Release pipeline

GitHub Actions is configured to test and vet the Go source, build ARM64 archives, produce checksums and GitHub provenance, and create a draft release for version tags. Local builds and checks pass; hosted workflow execution and a published release are not yet established.

## Installer behavior

The bootstrap checks AArch64 and Qualcomm IPQ/OpenWrt metadata, downloads a userspace archive, verifies its SHA-256 checksum, preserves existing configuration and saves the previous binary. It does not automatically restart the daemon. Checksums detect corruption; on-device attestation verification is not implemented.

## Platform bundle

The supplied vendor IPKs are preserved unchanged in a local integration candidate. Its manifest records dependencies, architecture and file hashes. Installation is gated on dependency closure, exact driver ABI, firmware pairing and a tested recovery path. Public CI currently packages original userspace only.

| Qualification stage | Exit evidence |
| --- | --- |
| 1 / Platform | Hardware identity, packages, regulatory and interface capabilities captured |
| 2 / Two-node service | Current binary associates, carries client traffic and survives controlled failures |
| 3 / Recovery | Four-node alternate path restores service without persistent loops |
| 4 / Scale | Eleven-node chain and concurrent-load trials pass agreed thresholds |
| 5 / Interoperability | Required EasyMesh behaviors and mixed-vendor cases verified |
| 6 / Publication | Raw measurements, limitations, hashes and validated release profile published |

REFERENCES & STATUS / 09

# Traceable engineering

Source documents and evidence boundaries for revision 0.1.

| | |
|---|---|
| IEEE 1905.1 | Standard record: standards.ieee.org/ieee/1905.1/4995/ |
| EasyMesh association | PDCA: Practical Dynamic Client Association in Wi-Fi Mesh Networks Using EasyMesh. IEEE ICC 2025. DOI: 10.1109/ICC52391.2025.11161355 |
| Multi-radio routing | Software-defined networking-based adaptive routing for multi-hop multi-frequency wireless mesh. IEEE TVT 2021. DOI: 10.1109/TVT.2021.3116266 |
| Scheduling research | A Self-Healing Mesh Network without Global-Time Synchronization. arXiv:2401.15168. LoRa implementation; not direct Wi-Fi/T2 evidence. |
| Project technical sources | docs/FEATURES.md; docs/SECURITY.md; docs/PTMP-AIRTIME-PAPER.md; docs/RELEASES.md |
| Target research | research/QSDK-EASYMESH-T2-INTEGRATION.md; research/DEVICE-BASELINE.md; research/VENDOR-IPC.md |

## Literature review boundary

The research review used accessible abstracts, institutional records and public documentation. Paywalled papers were not read in full. Their algorithms and performance claims are not reproduced or treated as T2 benchmark results.

## Evidence at publication

Local Go tests and vet passed. ARM64 release archive and local vendor-bundle integrity were checked. Historical T2 notes record EHT160. Current Wi-Fi 7 mode, adapter performance, driver scheduling enforcement, ten-hop forwarding and replacement interoperability still require fresh device evidence.

## Document identity

WantasticMesh engineering draft, revision 0.1, 13 November 2026. Original project code is MIT licensed; platform components retain their own licenses. Rebuild this paper with tools/build-paper.py using ReportLab. The document deliberately contains no invented performance charts.