

v0.7.0 Gateway Failover — CHANGELOG

Phase 1: Pi 5 Auto-Reconnect (Feb 27, 2026)

Goal: Gateway automatically detects BLE disconnection and reconnects without manual restart.

Changes

dc_gateway.py

- Added reconnect state fields: `_was_connected`, `_reconnecting`, `_last_connected_address`
- Added `_auto_reconnect_loop()` — background task checks connection every 2s, rescans + reconnects on disconnect
- Added reconnect guard on `send_command()` — returns early with warning during reconnection
- Set `_was_connected` and `_last_connected_address` on successful connect
- **Bug fix:** `connect_to_node()` now returns `False` if subscribe fails (wrong device — no DC01 service), disconnects, and lets the caller try the next device
- **Bug fix:** Uses `dangerous_use_bleak_cache=False` in `BleakClient` to force fresh GATT service discovery (prevents stale BlueZ cache)

power_manager.py

- Added `_paused` flag to `__init__`
- `poll_loop()` skips polling when `_paused` is `True` (set by reconnect loop during disconnection)

tui_app.py

- `connect_ble()` now iterates through ALL discovered devices until one connects AND subscribes successfully (eliminates need for `--address`)
- Target address device is tried first if `--address` is given
- Starts `_auto_reconnect_loop()` on the BLE thread after successful connection
- `on_unmount()` sets `gateway.running = False` before stopping BLE thread for clean shutdown

Bugs Fixed (bonus)

- **--address no longer required** — gateway probes each discovered ESP-BLE-MESH device and connects to the first one with the DC01 GATT service
- **Stale GATT cache** — forced fresh service discovery prevents "Characteristic not found" errors after device reboots

Phase 2: Universal Node Firmware (Feb 28, 2026)

Goal: Consolidate GATT gateway capabilities into the sensor node firmware, creating a single "Universal Node."

New Files (ESP-Mesh-Node-sensor-test/main/)

- `gatt_service.c/h` — NimBLE GATT service (DC01), chunked notify, GAP events, advertising as "DC-Monitor"
- `command_parser.c/h` — Pi 5 command parsing, self-addressing (local processing when target == self), ALL: support
- `node_tracker.c/h` — Known node tracking with dedup and self-exclusion
- `monitor.c/h` — FreeRTOS timer-based continuous polling

Modified Files

- `main.c` — Added GATT init calls (still a thin 69-line orchestrator)
- `mesh_node.c/h` — Added Vendor CLIENT model alongside SERVER (dual-role), vendor client state exports, GATT notify integration
- `CMakeLists.txt` — Added all 4 new .c files
- `sdkconfig.defaults` — Added `CONFIG_BT_NIMBLE_ENABLED=y` and `CONFIG_BT_NIMBLE_ATT_PREFERRED_MTU=185` (was using Bluedroid, needed NimBLE for GATT APIs)

Build Issues Encountered

- **host/ble_hs.h: No such file or directory** — `gatt_service.h` included NimBLE header unnecessarily; removed from .h (only needed in .c)
- **sdkconfig still had Bluedroid enabled** — the sensor node's `sdkconfig.defaults` was missing `CONFIG_BT_NIMBLE_ENABLED=y`. Fix required deleting `sdkconfig + build/` and running `idf.py set-target esp32c6 && idf.py build` to regenerate from updated defaults

Verified

- Compiles cleanly with `idf.py build`
- Provisioner detects and binds all 4 models (OnOff SRV/CLI, Vendor SRV/CLI)
- `ALL:READ` returns sensor data from both NODE0 (local) and NODE1 (mesh)

Known Bugs (fixed in Phase 3)

- ~~NODE0 data appears twice on `ALL:READ` (local + mesh echo) — fixed~~
- ~~`sensing_node_count` shows 1 instead of 2 (hardcoded `len(devices)` — 1 assumes old dedicated gateway) — fixed~~

Phase 3: Python Failover + Bug Fixes (Feb 28, 2026)

Goal: Update Python gateway to discover universal nodes by name, failover on disconnect, and fix Phase 2 bugs.

Changes

`constants.py`

- Added "DC-Monitor" to `DEVICE_NAME_PREFIXES` (universal nodes advertise with this name)

`dc_gateway.py`

- Replaced reconnect logic with **failover logic**: on disconnect, scans for ALL available nodes, tries each one (skipping the dead node first), falls back to the dead node as last resort
- Logs now say `[FAILOVER]` instead of `[RECONNECT]` when connecting to a different node

`tui_app.py`

- Fixed `sensing_node_count`: changed from `len(devices) - 1` to `len(devices)` — with universal nodes, every node (including the connected one) is a sensor

`power_manager.py`

- **Bug fix**: Removed strict duty confirmation that was preventing convergence. Previously, if a mesh node's response showed the old duty (due to ~1s mesh round-trip), PM would reject the change and revert `commanded_duty` to 100%. Now always updates `commanded_duty` after sending — the next poll cycle self-corrects if the command was truly lost.

`mesh_node.c` (ESP firmware)

- **Bug fix**: Added self-echo guard in vendor server callback. When the gateway node sends an `ALL:READ` group command, the mesh delivers it back to itself. Since `command_parser.c` already processes the local part, the vendor server now skips commands from its own address. Eliminates NODE0 duplicate data on `ALL:READ`.

Issues Encountered

- **NODE0 duplicate data** — `ALL:READ` showed NODE0 twice: once from `command_parser.c` local processing, once from mesh group echo. Fixed by adding `src_addr == node_state.addr` guard in `mesh_node.c`.
- **PM slow convergence / oscillation** — PM was fighting itself: sending correct duty targets but rejecting them because mesh response showed the old duty. Showed as `"N1 did not confirm duty:75%, keeping cmd=100%"`. Fixed by trusting the send and updating `commanded_duty` unconditionally.
- **Sent:ALL:READ ordering** — the `"Sent: ALL:READ"` log appears after NODE data because the ESP responds so fast via GATT notify that the response arrives before the send log prints. Cosmetic only, not a bug.

Verified

- `gateway.py` discovers nodes by both `"ESP-BLE-MESH"` and `"DC-Monitor"` names
- `ALL:READ` returns one entry per node (no duplicates)
- `sensing_node_count` shows correct count (2 for 2 nodes)
- PM converges in 2-3 cycles (was oscillating indefinitely before)
- Power-cycling connected node → Pi 5 fails over to a different node automatically
- PM resumes operation after failover