Episode Book

EPISODE #6561

What Access Point Firmware Updates Actually Change

PUBLISHED	13 November 2026 at 20:54 UTC
DURATION	28:43
LISTEN	myweirdprompts.com/episode/access-point-firmware-updates-11303/
HOSTS	Corn, Herman, Dorothy, Hilbert
SCRIPT MODEL	z-ai/glm-5.3
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.26
SOURCES	25 cited by the research (listed at the end)
BOOK GENERATED	14 November 2026 at 06:47 UTC

SYNOPSIS

Clicking update on a Ubiquiti access point hides a surprisingly detailed changelog. Examining firmware 8.8.14, 8.7.11, and 6.7.54 reveals that AP updates are dominated by wireless bugfixes, WiFi 7 Multi-Link Operation, and 6 GHz regulatory table changes for eight countries—while security CVEs actually live in separate UniFi application bulletins. The code itself is a four-layer stack: an aging OpenWrt or LEDE base, MediaTek or Qualcomm vendor SDKs, and proprietary binary radio blobs.

TRANSCRIPT



Corn

You click update, the box reboots, it comes back, and you go on with your life knowing nothing. That's Daniel's entire experience, and honestly it's everyone's. He updated his Ubiquiti access point recently and describes the ritual perfectly: notification, click, reboot, "feel responsible, and move on." His phrase. And then the feeling he's left with, which I know well, is that you've done something deliberate to a piece of hardware and you cannot say what.



Herman

That's the whole tension in his message, isn't it. He's not asking a lazy question either — he wants the anatomy. What does a typical access point firmware release actually ship beyond security patches, who writes the code, and what are we physically and legally doing to the device. Four questions hiding in one paragraph.



Corn

Three and a half. But his instinct that it's a closed-source black box with no visibility — that's what we should test, because I don't think it survives contact with the release notes.



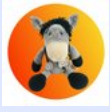
Herman

It half survives. That's the fun part.



Corn

Let's open the box, then. Layer by layer.



Herman

Here's the reframe I want to plant early. The black box is less black than it feels, but only if you go looking. Ubiquiti publishes release notes for every access point build that are, frankly, more detailed than most consumer software companies manage — itemized bugfix lists with real technical specifics. The controller interface just doesn't show you any of it. It says "new version available," you click, done. So the information exists and the moment of clicking hides it. That mismatch is the episode.



Corn

The documentation exists and the button is designed for people who don't want it.



Herman

Which is most people, to be fair.



Corn

I'm most people.



Herman

You are not most people, Corn. You read the history of things overnight.



Corn

I read the changelog overnight, once, years ago. It's how I learned never to trust a dot release. Anyway — specimen?



Herman

Firmware 8.8.14. Published to Release Candidate on November third, promoted to Official two days ago, November twelfth, for a long list of U7 and U6 models. This is about as current as an access point release gets — it's literally a week old. And the shape of it is the shape of the whole answer: two new features, seven improvements, eighteen bugfixes.



Corn

Eighteen bugfixes to two features. That ratio is the whole taxonomy, isn't it.



Herman

On a current platform, roughly. Call it ten percent features, twenty percent performance and reliability, seventy percent making the previous release's behavior stop happening. And the two features are worth walking through, because they're not gimmicks. The first is WiFi 7 Multi-Link Operation on mesh backhaul — U7 access points can now run their mesh links across 5 gigahertz and 6 gigahertz simultaneously, transmit and receive on both bands at once. That's the flagship WiFi 7 feature, and until now it was for client connections only. Extending it to the backhaul means your mesh units talk to each other over what is effectively two cables instead of one.



Corn

And the second?



Herman

Multicast Suppression for WiFi networks. Broadcast and multicast traffic is the classic inefficiency on wireless — a packet that every client has to process at the lowest common data rate. Suppressing it where it isn't needed is the kind of unglamorous change that quietly buys you real airtime.



Corn

The improvements list is where I got interested, though. Walk me through it, because there's an item in there that I did not expect.



Herman

Mesh uplink roaming — mesh access points now switch to a better uplink seamlessly instead of clinging to a mediocre one. RADIUS servers specified by hostname instead of IP. Better connectivity for legacy clients, the WiFi 4 laptops of the world. Smarter background neighbor scanning, specifically to reduce client latency spikes, which is a very polite way of saying the AP was freezing your traffic while it looked around.



Corn

And?



Herman

"Improved firmware update and boot reliability."



Corn

The updater updates itself.



Herman

The updater is a feature. It has bugs. It gets patched by... itself, in the best case. There's something almost recursive about it — the thing you trust to not destroy the device while replacing its software is on the list of things that gets replaced.



Corn

That's like the surgeon having his own surgery scheduled for that afternoon.



Herman

I'd want to know the order of operations, frankly. And then the last item, the one I'd argue matters more than everything above it: "Updated 6 gigahertz regulatory for Australia, Bangladesh, Bermuda, Iraq, Nigeria, North Macedonia, Paraguay and South Korea."



Corn

Hold on. That's a law change shipping as a firmware item.



Herman

That's exactly what it is. The regulatory domain tables — allowed channels, transmit power ceilings, how the device behaves around automated frequency coordination — live in firmware. Eight countries just got their radio rules rewritten inside this image. When you click update, you are potentially changing what your hardware is legally permitted to emit.



Corn

I update my access point and Bermuda gets a say.



Herman

Bermuda gets a very small say, yes.



Corn

Then the eighteen bugfixes. My read of that list is that it's almost entirely wireless behavior — not a security patch in sight.



Herman

Go down it. Clients occasionally failing to pass traffic after connecting. IPv6 connectivity to wireless clients lost over time. Fast Roaming not working until the access point restarts. Voice and video traffic losing WiFi priority on WiFi 7 access points. Six gigahertz radios stuck at reduced transmit power. Two separate VLAN leakage fixes — traffic from one network leaking into the management network during configuration changes, and VLAN leakage on Auto-Link ports with Network Override. Mesh uplink reliability, mesh configuration sync. The status LED not indicating uplink loss, which is the one item I'd classify as cosmetic, except the LED lying to you is its own genre of problem.



Corn

VLAN leakage into the management network is not a bugfix, that's a nightmare with a polite title. And stuck-at-reduced-transmit-power is the inverse of the regulatory point — the radio's legal ceiling exists in software, and the bug is it stuck to the floor.



Herman

You've just made the connection I was going to make two segments from now. Save that.



Corn

I'll be here. Slowly.



Herman

Compare it with 8.7.11, the previous major release from July. Same overall shape but visibly heavier on features. It had a whole Roaming Improvements section — access points discovering and advertising neighbors that aren't directly within wireless range, which is a mesh topology thing. A Force WiFi 4 Mode toggle for environments full of ancient clients. Extended VLAN support on the U7 In-Wall across the full 802.1Q range, all four thousand and nine tags. Restricted Target Wake Time on WiFi 7. RADIUS Accounting Stop packets on reboot and firmware upgrade — the accounting system now knows the session ended, instead of just wondering. Improved boot speed on access points with many WiFi networks configured, and automatic certificate renewal for the captive portal.



Corn

And the bugfix side?



Herman

About two dozen, and again it's wireless behavior. The two I'd single out: "Fixed severe 5 gigahertz WiFi performance degradation after automatic off-channel scans" — severe degradation is the release note admitting the AP was hobbling itself while doing that neighbor scanning — and "Fixed 6 gigahertz radios ignoring configured reduced transmit power after AFC responses." The frequency coordination system told the radio to turn down, and the radio ignored it.



Corn

The radio ignoring the thing that keeps it legal. Two releases running, on that theme.



Herman

There's one more specimen worth thirty seconds, and it's the contrast class. 6.7.54, for the older U6-LR, U6-Lite, U6-Plus and the HD-generation hardware. Three improvements, four or five bugfixes, nothing you'd call a feature.



Corn

That's a mature platform in maintenance mode.



Herman

Same vendor, same product line, and you can tell from the changelog alone which hardware still has engineers assigned to it and which has a caretaker. That's information Daniel can't get from the update button but gets for free from the notes.



Corn

So his hope is confirmed — features and performance do ship, in the current line, and bugfixes dominate. But his "security patches are an obvious yes" assumption. I looked at that and something's off.



Herman

Something is structurally off, and it's my favorite finding in this whole topic. The AP firmware release notes contain no CVE identifiers. None. Zero across these releases. Security fixes live in a completely separate channel — Security Advisory Bulletins. Bulletin 067, from late August, covers twenty-two CVEs with severity scores up to a perfect ten. But look at where they land: UniFi Protect, the Network Application, Access, Talk, UniFi OS. The application layer. The controller and its ecosystem, not the access point firmware. Bulletin 072, end of October, covers one CVE that can be chained to root command execution — also in the application side.



Corn

So the patch Daniel clicks on the AP is overwhelmingly behavior fixes, and the vulnerabilities mostly live one layer up in the software he's already watching anyway.



Herman

His assumption is accurate but mislocated. The security work is real, it's just not in the box he's opening. And whether AP firmware itself has CVEs published anywhere is unresolved — the notes never cite them.



Corn

That's what's in the box. Now the question I actually care about, because this is where "black box" stops being a feeling and starts having an address. Who writes it?



Herman

Four layers, and each one has different authors, different incentives, different transparency. We did an episode on the bootloader hierarchy ages ago, the immutable-code ladder from mask ROM upward — today is the level above that: who authored each rung. Layer one, the base. UniFi access points run a customized OpenWrt — properly, LEDE, the fork era. People SSH into their units and post the evidence. The openwrt release file on a nanoHD reads LEDE 17.01.6, kernel 4.4.153, a 2017-era toolchain. Firmware from 2023 was still carrying it.



Corn

A 2017 operating system under a 2026 product.



Herman

A 2017 operating system doing its job, in fairness. But it tells you how far the open-source base can lag behind the product. One commenter on a Hacker News thread disputes the whole framing — claims UniFi switched to unifiOS based on Vyatta over a decade ago. The SSH dumps are stronger evidence, but I like that the disagreement exists, because even layer one is contested.



Corn

Layer two.



Herman

The chipset maker's SDK. The SoC and radio vendor — MediaTek, Qualcomm — ships a board support package, and the device vendor builds on it. And the people who work with these things are blunt about it. A forum post from March, on a MediaTek-based Zyxel device: "Technically, it's a vendor SDK, provided by Mediatek, from our point of view it's not even Openwrt, despite the name." MediaTek does maintain a public mirror of their OpenWrt SDK on GitHub, so it's visible — but a large fraction of what the vendor "writes" is integration on top of a foundation they didn't author.



Corn

So when Daniel asks "the device vendor, the chipset maker's SDK, or open-source components underneath," the answer is yes.



Herman

All three, stacked. And then layer three, which is the one I'd call the actual black box inside his black box. The Linux WiFi drivers — ath10k, ath11k, ath12k — are open source. Fine. But they don't run the radio. They load proprietary binary firmware from the firmware directory, and that code executes on the chipset itself. OpenWrt's own build packages these blobs for seven Qualcomm chipset families — IPQ5018, IPQ8074, QCA2066, QCA6390, QCN9074, WCN6750, WCN6855 — under a licence file that amounts to "Atheros firmware, don't ask." And there's a separate package supplying the per-board calibration data, tuned by the vendor for each specific board. Even a fully open-source router build depends on opaque binaries and per-device calibration files.



Corn

You can compile your entire router from source and still ship a stranger's sealed envelope.



Herman

That's the cleanest description of it I've ever heard, actually. The envelope can't be opened, only loaded. And layer four is Ubiquiti's own code — the controller integration, the mesh logic, the roaming implementation, the LED behavior, the captive portal. That's the layer the release notes describe, and it's closed.



Corn

Which explains something I've noticed about those release notes. They document behavior with real specificity — stuck transmit power, VLAN leakage — and say nothing about code. Now I know why. The layer doing the talking is the one layer whose code is fully theirs.



Herman

There's a licensing wrinkle worth naming, because Daniel's an open-source developer and it'll land. GPLv2 section three obliges you to give source to the people you distribute binaries to — not to the public. So MediaTek handing SDK source to device vendors, who must then offer it to buyers on request, is arguably compliant. In practice, what a buyer



Herman (continued)

gets on request is often stale — the Zyxel case this year yielded a 21.02-era SDK tarball, years out of date.



Corn

The right exists on paper and expires in practice.



Herman

And the sharpest quote in all of this — Felix Fietkau, a core OpenWrt developer, on the openwrt-devel list in early 2024, about Qualcomm's CVE fixes: "The fixes were not shared with OpenWrt. Qualcomm does not care about OpenWrt support for their platforms." A second developer in that thread, Sven Eckelmann, was told by Qualcomm that the ath10k firmware is no longer supported and ath11k firmware, quote, "is not developed further" — and the repositories that replaced the old ones had locations that were, quote, "not revealed."



Corn

Not revealed. To the people maintaining the open driver that loads it.



Herman

To anyone. So the layer that writes the radio firmware has no legal obligation and, on that evidence, no interest in saying what changed — not to the public, not to OpenWrt, and at least sometimes not even to the vendor's own downstream. That's the structural answer to Daniel's black-box feeling. It's not that the box is uniformly dark. It's that the darkest drawer in it belongs to a company he's never heard of, writing code he can't see, for a radio he thinks he bought from Ubiquiti.

D

Dorothy

Hello dear, it's your mother. Time for my grim fact.

D

Dorothy

A fellow called Felix Fietkau, who helps make that OpenWrt thing, wrote that Qualcomm's security fixes were not shared with OpenWrt, and that Qualcomm does not care about OpenWrt support for their platforms. And a man from Qualcomm told Sven that the radio firmware isn't being developed any further.

D

Dorothy

Isn't that nice. Now, sweetheart, did you ever put that casserole dish back in the cupboard, and when are you coming to see me?



Herman

Mum, that was — that's the exact quote we were just discussing, near enough word for word.



Corn

She reads the mailing lists now. I don't know when that started and I'm not asking.



Herman

The casserole dish is on the shelf, Mum. We'll come Sunday.



Corn

So who actually writes the code in your access point? Whoever feels like telling you.



Herman

Moving to the last part of Daniel's question, then — what are we physically doing to the hardware. First and most basic: you're rewriting flash. The firmware lives in non-volatile memory, and the update writes a new root filesystem image over it. That's the physical event. Everything else follows from it.



Corn

And the thing that makes that not terrifying is a safety net nobody documents.



Herman

In most modern gear, yes — the A/B scheme. Android made it famous: two complete partition sets, the update goes onto the inactive slot, and if the device fails to boot afterward, the bootloader falls back to the old slot. U-Boot implements the same idea through boot counts and a fallback boot command, and a newer spec does it with trial states — the new bank has to prove itself before the system commits to it. NVIDIA's Jetson docs describe marking a failed slot unbootable. MediaTek documents it for their IoT platform.



Corn

Standard practice, everywhere, except we have no idea if his access point does it.



Herman

That's the honest position. Whether UniFi access points use true A/B or a single image plus a recovery partition is not stated in any release note or forum post I've seen. And I keep coming back to that line in 8.8.14 — "Improved firmware update and boot reliability." You don't list that as an improvement unless the path itself has been a source of trouble.



Corn

The one item in the notes describing the mechanism of the notes.



Herman

Now, we said the regulatory tables and transmit power caps ship as firmware — that's the legal dimension of clicking update. And then there's the failure dimension, and there's a report from literally this morning that I want to put in front of you. A U6-Mesh owner, on the Ubiquiti community thread for 8.8.14. He updated from 6.8.2 — an older track — to 8.8.14. The update appeared to succeed. Roughly two hours later, at half past ten in the morning, the device, quote, "suddenly went completely offline and became unreachable through the UniFi Network application." Multiple restarts failed. Only a factory reset brought it back.



Corn

Two hours. It didn't brick, it defected.



Herman

The failure mode isn't the one anyone pictures. Not a dead box mid-flash. It's an update that succeeds, hands the device back, and then takes it off the network hours later. If you'd clicked update and gone to bed, you'd have woken up to a hole in your coverage with no recent cause.



Corn

And the factory reset — did it come back as a new device?



Herman

That's the strange detail. The controller recognized it as the existing device rather than treating it as newly adopted. So the controller's identity tracking survived what the device itself didn't. There's more in that thread, too. Another user traced a regression after the same jump — fast roaming followed by the phone going silent to ARP requests and IPv6 neighbor solicitations, reproduced twice out of two with Proxy ARP enabled on the SSID, and fine three out of three with it off. That's a proper isolated variable, which is more debugging rigor than most paid products get.



Corn

Somebody's getting to the bottom of their broken WiFi and the notes were no help.



Herman

A third user reported a Samsung TV failing to connect on 5 gigahertz after the update. Consumer electronics roulette — you update the access point and find out which of your devices was standing on a crack all along.



Corn

And that gets us to the part I think is the buried headline. That same 6.8.2 to 8.8.14 jump — it's not just an old version to a new one.



Herman

It's a platform migration. 8.8.14 says it outright, in a paragraph positioned above the feature list where almost nobody reads: "U6 is joining the same software feature track as U7. Going forward, U6 devices will receive the same applicable software features, improvements, and fixes, while hardware-specific capabilities remain dependent on the AP model."



Corn

He didn't apply a patch. He was migrated.



Herman

The U6 line was on its own codebase, the 6.x train, the maintenance-mode train we saw in 6.7.54. This release folds U6 hardware onto the U7 software. Same firmware image family, same feature flow, the hardware differences handled underneath. So that user's two-hour failure isn't surprising in that light — his device ran its first day on a codebase it had never run before, and the release notes he could read describe a fraction of what actually changed.



Corn

The most consequential thing in the update was the sentence above the list, not anything on it.



Herman

And Daniel's black-box feeling, restated in full: the behavior is documented better than he thinks, the code is layered under four different authors, the layer that matters most says nothing, and sometimes what you're really doing is moving your hardware to a new platform wearing the costume of a version number.



Hilbert

Black box is the wrong word.



Hilbert

A black box you can't see into. This one you can see into, you just don't like what's in it. We called that a "windowed brick" on the microwave links I maintained in seventy-nine. The radio module came sealed, no service docs, but the fault codes printed right on the front panel. Windowed brick. Corn's been saying blob all episode — it's not a blob, it's a brick.



Corn

Windowed brick. That's better than mine, which is annoying.



Hilbert

And the version thing you were both circling. The last segment of the build number isn't decoration — it counts the rebuilds. Eight eight point fourteen, dot twenty-oh-one-seven-nine means the image was built again that many times before anyone was allowed to ship it. I keep a sheet of every version my own unit has run, with the bugfix counts from the notes next to each one. Printed, taped inside the networking cupboard door. Three pens — black for installed, blue for the notes count, red for what actually went wrong. I update it by hand. The controller logs all of it, but the controller's logs are in a computer and the cupboard is where I am when it breaks.



Herman

The cupboard ledger. You stood in front of the AP with a pen.



Hilbert

In nineteen seventy-nine I wanted a portable data terminal that talked to the brick over a short radio link. Sixty-two hundred, I still know the price, from a man I met on the ferry to Hook of Holland who'd bought two and said the third was always cheaper if you waited for the letter. I waited for the letter. The letter never came, the terminal doubled in price, and I have maintained sheets of numbers ever since instead. I don't install anything whose build number ends in an odd digit. Rolled back a perfectly good update once for exactly that. Failed anyway, two days later. Which proves the rule was right and I applied it too late.



Herman

Two days later sounds like the U6-Mesh, honestly. That delayed failure.



Hilbert

The thing Herman said about the updater patching itself — that's the part I never thought about. My sheet only counts what they tell me changed. Herman, you talk for ten minutes before the number, but it was a good number. Corn, your cupboard's got room for one sheet. Corn's the one who'd keep it honest.



Corn

I'd keep it in pencil. I don't commit in pen until the second failure.



Herman

The pencil amendment is accepted. And there's an open question Hilbert's ledger can't capture, which is the one I'd leave listeners with as the cutting-room floor item: nobody has published whether UniFi access points use true A/B slots. Android documents it, U-Boot documents it, Jetson documents it, MediaTek documents it — for this hardware, silence. The one line in 8.8.14 about update and boot reliability is the only public hint the safety net exists, and hints are not documentation.



Corn

Also hanging: no public source release for the UniFi AP firmware was ever found — the GPL request path exists in theory and yields a stale tarball in practice, when it yields anything. And the AP firmware CVEs, if they happen, are published somewhere we didn't find, because the release notes never cite one.



Herman

So the closing thought, built on the cupboard ledger: the gap that matters isn't between what the update does and what the notes say — that gap is small. It's between what's documented and what's consequential. The migration paragraph sits above the feature list, the regulatory tables ship without ceremony, and the brick in the radio never says anything at all. Whoever writes the last line of that stack is the one who decides what your hardware did that morning. Thanks, as ever, to Hilbert Flumingtop for producing the show.



Corn

If you've got ten minutes and a cupboard door, my weird prompts dot com has everything else. If this was your kind of episode, go back for episode twenty-five ninety-four, The Hierarchy of Immutable Code; and episode five, Fine-Tuning ASR For Maximal Usability. This has been My Weird Prompts.



Herman

See you soon.

SOURCES

What the research for this episode read before the script was written. Primary sources first.

1. **Ubiquiti Community, UniFi Access Point & Wireless Bridge 8.8.14, published 2026-11-03, Official 2026-11-12, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/UniFi-Access-Point-and-Wireless-Bridge-8-8-14/10247e29-044d-48b2-9e83-e7c5fb17305a

2. **Ubiquiti Community, UniFi Access Point (all U7 and E7 models) 8.7.11, published 2026-07-20, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/UniFi-Access-Point-all-U7-and-E7-models-8-7-11/42b6f9d9-3dba-4cda-bde1-b8157edc1299

3. **Ubiquiti Community, UniFi Access Point for U6-LR, U6-Lite, U6+ & more 6.7.54, edited ~5 months before retrieval, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/UniFi-Access-Point-for-U6-LR-U6-Lite-U6-and-more-6-7-54/758132b8-27dc-4a5f-ac6c-5b87aa80a777

4. **Ubiquiti Community, UniFi Access Point (all U7 and E7 models) 8.6.11, edited ~5 months before retrieval, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/UniFi-Access-Point-all-U7-and-E7-models-8-6-11/0d476b79-b684-4370-96af-ccb21ce35cbd

5. **Ubiquiti, Security Advisory Bulletin 067, published 2026-08-26, v1.0, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/Security-Advisory-Bulletin-067/fc4a3488-7c43-4628-8bab-f715e96dbfc9

6. **Ubiquiti, Security Advisory Bulletin 062, published 2026-03-18, updated 2026-03-21, v1.1, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/Security-Advisory-Bulletin-062-062/c29719c0-405e-4d4a-8f26-e343e99f931b

7. **Ubiquiti, Security Advisory Bulletin 072, published 2026-10-29, updated 2026-10-31, v1.2, retrieved 2026-11-13. [primary]**

   https://community.ui.com/releases/Security-Advisory-Bulletin-072/351a6d54-3ef7-49b4-ac99-b3a5388a9808

8. **Ubiquiti, Security Advisory Bulletin 056, published 2025-10-23, v1.0, retrieved 2026-11-13. [primary]**
https://community.ui.com/releases/Security-Advisory-Bulletin-056/ce97352d-91cd-40a7-a2f4-2c73b3b30191

9. **Ubiquiti, Security Advisory Bulletin 051, published 2025-07-16, v1.0, retrieved 2026-11-13. [primary]**
https://community.ui.com/releases/Security-Advisory-Bulletin-051-051/583fa6e1-3d85-42ec-a453-651d1653c9b3

10. **openwrt-devel mailing list, Question to recent Qualcomm CVEs, Sven Eckelmann and Felix Fietkau, 2024-02-26, retrieved 2026-11-13. [primary]**
https://lists.openwrt.org/pipermail/openwrt-devel/2024-February/042350.html

11. **OpenWrt source repository, `qca_ath11k.mk`, retrieved 2026-11-13. [primary]**
https://git.openwrt.org/openwrt/openwrt/tree/package/firmware/linux-firmware/qca_ath11k.mk

12. **Android Open Source Project, A/B (seamless) system updates, retrieved 2026-11-13. [primary]**
https://source.android.com/docs/core/ota/ab

13. **U-Boot documentation, FWU Multi Bank Updates, retrieved 2026-11-13. [primary]**
https://docs.u-boot-project.org/en/latest/develop/uefi/fwu_updates.html

14. **NVIDIA Jetson Linux Developer Guide, Update and Redundancy, retrieved 2026-11-13. [primary]**
https://docs.nvidia.com/jetson/archives/r39.2/DeveloperGuide/SD/Bootloader/UpdateAndRedundancy.html

15. **MediaTek IoT Yocto docs, A/B Firmware Updates, retrieved 2026-11-13. [primary]**
https://genio.mediatek.com/doc/iot-yocto/latest/sw/yocto/ab-part.html

16. **OpenWrt Forum, Ubiquiti Unifi is Openwrt firmware, retrieved 2026-11-13.**
https://forum.openwrt.org/t/ubiquiti-unifi-is-openwrt-firmware/57593

17. **OpenWrt Forum, Add ath12k**
WLE7002E56, 2025-03-26, retrieved 2026-11-13.
https://forum.openwrt.org/t/add-ath12k-wle7002e56-opensource-driver-firmware-support/228471

18. **OpenWrt Forum, Adding OpenWrt support for Zyxel EX5601-T0, 2026-03-14, retrieved 2026-11-13.**
https://forum.openwrt.org/t/adding-openwrt-support-for-zyxel-ex5601-t0/155914/1424

19. **DeepWiki, WiFi Firmware Management (OpenWrt), retrieved 2026-11-13.**
https://deepwiki.com/openwrt/openwrt/6.2.2-wifi-firmware-management

20. **DeepWiki, Licensing and Compliance (GL.iNet gl-image), retrieved 2026-11-13.**
https://deepwiki.com/gl-inet/gl-image/5-licensing-and-compliance

21. **hexaheximal, MediaTek and the GPL, 2024-04-25, retrieved 2026-11-13.**
https://hexaheximal.com/blog/mediatek-and-the-gpl

22. **GitHub, `mtk-openwrt-sdk/openwrt`, retrieved 2026-11-13.**
https://github.com/mtk-openwrt-sdk/openwrt

23. **Bootlin, ELCE 2022, Implementing A/B System Updates with U-Boot, retrieved 2026-11-13.**
https://bootlin.com/pub/conferences/2022/elce/opdenacker-implementing-A-B-system-updates-with-u-boot/opdenacker-implementing-A-B-system-updates-with-u-boot.pdf

24. **Wikipedia, Firmware, last edited 2026-11-08, retrieved 2026-11-13.**
https://en.wikipedia.org/wiki/Firmware

25. **Hacker News comment thread on UniFi/OpenWRT, 2023-05-18, retrieved 2026-11-13.**
https://news.ycombinator.com/item?id=35986330