TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

cnit consorzio nazionale interuniversitario per le telecomunicazioni

NEW AGE TOOLING BOF @ NETDEV 0X1A

The Cost Asymmetry of AI-Generated Code
a Case Study in the SRv6 Subsystem

Stefano Salsano[1,2]

[1]University of Rome Tor Vergata, Italy · [2]CNIT, Italy

Netdev 0x1A — The Technical Conference on Linux Networking

New Age Tooling BoF — *13–16 July 2026, Rome, Italy*

Disclaimer

- We are **not here to blame anyone** — this is a case study to learn from, and to improve the kernel development process.
- AI is a new and genuinely useful tool — this is **not a case against it**. How it should best be used in kernel development is **still being worked out**, so this is offered in a spirit of learning, not blame.
- The **patch author is not named** in these slides.

> **On the review this case study builds on.** It is based on the **public, on-list review** of the patchset; the discussion and every quote shown here are part of the **public LKML record**. The **reviewers are not authors** of this presentation and did not contribute to its framing, opinions or proposals — which are solely the presenter's — and nothing here is intended as criticism of the patch author beyond the technical points already made publicly on-list.

The case: one patchset, everything green

[PATCH v2 0/7] seg6: add SRv6 Mobile User Plane (RFC 9433)

- **6 new behaviors** + documentation, in one series
- **5,155 insertions**, 12 files — seg6_local.c grows 2.7k → 5k lines
- **v1 → v2 in one day**, CI findings all addressed
- **2,402 lines of selftests** (47% of the series) — all green
- Textbook commit messages, precise RFC citations

Takeaway. Every cheap quality signal looked good. None of them measured correctness.

0/7 cover letter

1/7 End.MAP (+191)

2/7 End.M.GTP4.E (+1,232)

3/7 End.M.GTP6.E (+717)

4/7 End.M.GTP6.D (+1,013)

5/7 End.M.GTP6.D.Di (+652)

6/7 H.M.GTP4.D

7/7 Documentation

The asymmetry, measured in the wild

1 **Generation: ~1 day per iteration.**

5,155 lines, six behaviors; v1 on May 4, v2 on May 5.

2 **Expert review: one month.**

8 emails, ~40 findings (May 8 – June 8) — and it covered only 5 of 7 patches.

3 **Outcome: start over.**

Nothing merged; full rework agreed — and the maintainer walked away with *more* work (two prep workstreams).

Takeaway. A 1 : 5–10 generation-to-review cost ratio is not a thought experiment — this thread measured it.

Anatomy of ~40 review findings

Class	What we found	Why it hurts
Spec conformance	Patch 4 implements **neither** RFC 9433 §6.3 **nor** §6.4; RFC 6040 mis-cited	**Fatal, invisible to CI**
Security / validation	Malformed SRH accepted as absent → HMAC bypass; HMAC invalid by construction	**Silent**
UAPI design	Required-but-unused attributes; NH6 / SRH / OIF semantics overloaded	**Irreversible once merged**
Data path & structure	GSO broken, stale checksums, unchecked fragments; ×5 duplicated logic	Bugs hide in the copies

Takeaway. The most damaging classes — spec, security, UAPI — are exactly the ones no automated pipeline caught.

The selftest that passed by coincidence

1

The bug.

End.MAP rewrites the IPv6 DA — the ICMPv6 checksum covers the DA, so it must be updated. It isn't.

2

The mystery.

The AI reviewer flagged it — yet the generated selftest stayed green.

3

The resolution.

The two test addresses have equal 16-bit word sums: `2001:db8:f::1` vs `2001:db8:2::e` — `0x000f + 0x0001 = 0x0002 + 0x000e`. Change one address → `Icmp6InCsumErrors`.

"The AI bot was right, but when I ran the selftest it passed." — Andrea Mayer, digging in

"Plausible-but-wrong": the signature failure mode

The code looks idiomatic. The prose is precise. The substance is wrong:

- Comments cite RFC 9433 **step numbers** — the code implements **neither §6.3 nor §6.4**
- Comments describe **code that is not there** (hop-limit decrement, LOCAL_OUT claim)
- RFC 6040 cited **for something it does not cover**
- The commit message's own example **never uses a required attribute**
- Commit message lists a drop reason **the diff never adds**

> **Punchline.** The AI reviewer scored this very patch cleanest of the series: **0 critical · 0 high · 0 medium · 1 low.**

> **Takeaway.** Surface signals of quality are decoupled from correctness — checking code against the *intent* of a spec is still human work.

Human vs AI review, on the same patches

Same perimeter (patches 1–5) — human reviewer: **Andrea Mayer** · AI reviewer: Sashiko (Patchwork) — every finding re-verified against kernel source.

	Human (5 weeks)	AI (hours, ~$34 in tokens)
Outcome-deciding classes (spec · UAPI · architecture)	**all of them**	**zero**
Overlap	~40 findings	reproduced 9 of them (~22%)
Unique findings	~31, incl. every blocker	5 → **2 real but minor** + **3 false positives**
False positives	0	**3 — each on an invented kernel mechanism**, incl. its only *Critical*

One human sentence on hop-limit handling (*"the forwarding path handles it"*) pre-refuted two AI attack scenarios.

Takeaway. AI review re-enacts the asymmetry one level up: findings are cheap to generate, expensive to verify — here, verification *debunked more than it confirmed*.

Lessons: dos & don'ts

- **Disclose AI use** — an honesty norm, orthogonal to acceptance
- **Small series** — one behavior per patchset (the End.DT4/DT6/DT46 model)
- **UAPI is forever** — design it by hand, review it first
- **Don't trust generated tests** — they share the generator's blind spots
- **AI-review output = claims to verify**, not findings
- **Humans keep the merge decision** — and the spec, UAPI, architecture calls
- **Cheap guards work** — the early RFC-downgrade cost nothing and helped

Takeaway. The scarce resource is *expert maintainer attention* — every rule above exists to protect it.

Two proposals for this BoF

1. Improve AI review — the margins are wide

- **Feedback loop**: track "% wrong" next to "% unique" — every debunked finding becomes eval signal
- **Maintainer-owned, per-subsystem review prompts** — the mechanism already exists; the contribution pipeline doesn't

2. Desk rejection — on quality, not provenance

- Reject code because it's **badly written**, not because it's AI-written
- Mechanical, provenance-neutral gates: **clone density**, dead-code & cast fingerprints, commit-msg ↔ diff consistency
- Below the bar: **reject wholesale**, return the cost to the sender

Take-home. Price the review *before* it starts — no AI-detector arms race needed.

This case study builds on the **public, on-list review** by **Andrea Mayer** — part of the public LKML record (see the *Disclaimer*).

Thank you. Discussion: who defines the bar — CI, Patchwork, MAINTAINERS?

stefano.salsano@uniroma2.it