“Open. Wait. Repeat.”

Why Wireshark Collapses Under Modern Car Data Volumes

What “car-data” really means today

Modern architectures are multilink, multi-speed, always-on. Even modest loads explode in volume when several links run together.

Modern vehicles generate datacenter-scale traffic. At these volumes, Wireshark’s single-file, packet-by-packet workflow turns every basic action into hours of dead time… before analysis even starts.

The math is linear, brutal, and impossible to ignore.

The Waiting Game (a.k.a. your day, gone)

Time to open standard ".pcapng" files from measured load times of standard PCs:

- 2 GB (~16.5 M packets) → 40 s
- 4 GB (~33 M packets) → ~ 80 s (~ linear)
- Add one simple filter → ~ +70 % time

That's ~ 20 s/GB just to open and prep a file.

Now scale that up:

Time to open a 500GB file with Wireshark:

500 GB (~ 2.4 - 5 min of tapping multiple links on an SDV Network)

- Open only: 500 × 20 s = 10,000 s → **2 hours 46 min.**
- With a simple filter: 10,000 s × 1.7 → **4 hours 43 min.**

Time to open a 1TB file with Wireshark:

1 TB (~ 5-10 min of tapping multiple links on an SDV Network)

- Open only: 1000 × 20 s = 20,000 s → **5 hours 41 min.**
- With a simple filter: 20,000 s × 1.7 → **9 hours 40 min.**

*Remember: This is just the time spent to **open one** test file!*

And that's just the first ingest/index pass. Tweak a view, change a filter, re-scan a time slice… you pay large chunks of that cost again.

Every change: "Open. Wait. Repeat."

…but even if you have the time…

In a modern zonal architecture of a SDV, validation requires multiple probes all over the network to measure timestamp and correlate the data under test.

Data streams in at several gigabytes per second, creating two main challenges: synchronizing multiple data taps to a common clock and ingesting the data fast enough into a standard PC to prevent packet loss, especially during bursts.

Why Wireshark's (and most other analyser models) struggle and will ultimately break here

1) Monolithic file thinking in a distributed world

Wireshark shines for interactive, human-scale traces:
… a NIC, a span port, a lab test, a few minutes.

But modern vehicles are dozens of concurrent links and hundreds of millions of packets per minute. Shoving all of that into one giant PCAPNG concentrates everything into a single I/O and indexing chokepoint. A PC&NIC is NOT built for this.

GUI-first, index-later

Wireshark's strength is the rich, packet-level UI. Its weakness at scale is that every "simple" action cascades into massive re-work:

- Open file → read headers, build indexes
- Apply filter → re-evaluate millions to billions of packets
- Jump time range → seek, re-hydrate state
- Change columns or decode options → touch a ton of data again

When each of those steps is measured in hours, the workflow is no longer interactive—it's nerve-racking and time-consuming batch processing in disguise.

Packet-level everything

Full-fidelity decoding for every packet across every link at every moment is wonderful for root cause at human scale.

At fleet-scale though, it's wasteful by default. Most questions don't require every byte of every frame; they need to catch session/flow facts, counters, or anomalies first.

Find the neuralgic spot first, then do the deeper dive!

Packet-first means you pay the maximum compute and I/O tax up front that you NEVER need.

Capture is the easy part

At an estimated ~ 2-3 GB/s (~ 16 - 24 Gbit/s), you already need multi-10 GbE logging just to keep up. But storage ingest isn't the only real limiter - the analysis is as well. With Wireshark, The analysis phase is throttled by single-node parsing, serialized GUI workflows, and re-indexing costs.

The Math is Linear. The Pain is Exponential.

The linear law of hurt

As mentioned above, load time scales approximately linear with size (~ 20 s/GB base, +70 % for one filter), doubling trace size doubles your "nothing is happening" wait.

But the number of times you need to reopen/re-scan grows with team size, questions asked, and hypothesis pivots—so the pain experienced grows even faster than linear.

Rule of thumb:

Minutes of capture × GB/s × 20 s/GB × filter multipliers = hours lost.

"But we can just pre-filter… right?"

Pre-filtering presumes you already know what to keep.

In validation, integration, and field triage, you often don't. The result is either:

- Over-filtering → missing the bug, or
- Under-filtering → you're right back to 1 TB+ artifacts Wireshark can't handle interactively.

"What about splitting the file?"

Splitting helps avoid single-file cliffs, and sometimes that the files loads at all, but it doesn't change the total work. You still pay 20 s/GB per shard, multiplied by the number of shards you need to open, re-open, and cross-correlate.

Now add time alignment across links and shards—your stopwatch keeps running.

"Use tshark then?"

tshark is invaluable for automated work, but it uses the same decode engine.

Headless helps with automation, not the fundamental throughput and re-index looping when questions evolve.

Some Numbers to consider from now on:

Link capacities at 100 % load (sanity check):

- A²B 50 Mbit → 6.25 MB/s
- Eth 10 Mbit → 1.25 MB/s
- Eth 100 Mbit → 25 MB/s
- Eth 1 Gbit → 250 MB/s
- Eth 10 Gbit → 2,500 MB/s

Sum: 2,782.5 MB/s = 2.72–2.78 GB/s → ~ 10 TB/h → ~ 240 TB/day

A more realistic mixed-load example with multiple links (but not saturated):

- A²B (10× at 60 %)
- Eth 10 M (10× at 30 %)
- Eth 100 M (10× at 25 %)
- Eth 1 G (6× at 25 %)
- Eth 10 G (4× at 25 %)

Resulting flow: ~ 2,962.5 MB/s ≈ 3 GB/s, ~ 180 GB/min, ~ 10 TB/h, ~ 250 TB/day.

That is ≈ 24 Gbit/s sustained (call it ~ 20–25 Gbit/s).

The four-minute wall

At ~ 3 GB/s, a 4-minute capture is ~ 720 GB.

- Open-only time (20 s/GB): 720 × 20 s = 14,400 s → ~ 4 hours
- With a simple display/capture filter (~ × 1.7): ~ 6 h 48 m

If your links burst higher (aggregates near 1 TB in 4 min), you're right back in the ~ 9–10 hour opening range with a single filter—before clicking your first conversation, flow, or graph.

The verdict

- Opening 500 GB: ~ 2 h 46 m (no filter) → ~ 4 h 43 m (one simple filter)
- Opening 1 TB: ~ 5 h 33 m – 5 h 41 m (no filter) → ~ 9 h 26 m – 9 h 40 m (one filter)
- Realistic mixed-load vehicle: ~ 3 GB/s → ~ 10 TB/hour → ~ 250 TB/day
- A 4-minute capture at this rate takes ~ 720 GB → ~ 4 hours just to open;
  ~ 6 h 48 m with a simple filter.

Conclusion:

Wireshark is a great tool! But in modern automotive architectures, Wireshark as the primary analysis tool is operationally infeasible.

The tool is extraordinary at micro-scale, forensic inspection, and that's exactly how it should be used by the expert who can use it!

Alternatives:

At GB/s and TB/day, the only viable path is distributed, stream-first telemetry with an integrated and a dedicated Capture and Analysis Tool that can ingest massive amounts of data, parallel, synchronously and across all valid transport layers, in a nanosecond accurate and reliable way and is capable to present the necessary data in real time to the user.

It must be automatable, scalable and able to do perform deep meaningful analysis of packets - over the whole bulk of data; Lightning fast and exactly like TSN CoreSolution 4 toolchain does.

If you are interested, you can find us here:

www.tsn.systems

TSN Systems GmbH

Dalbergstraße 7

36037 Fulda, Germany

Phone: +49 661 410 951 80