PURETENSOR
INTELLIGENCE DISTILLED

The Instrument Was Wrong

Stream count, slot topology and receive-path backpressure on a 200 GbE fabric

Paper: PT-TN-2026-007 v1.0

Author: PureTensor Ltd

Date: 22 April 2026

Status: Published

Abstract

A three-node 200 GbE fabric appeared to plateau at 56–70% of line rate while the 25 GbE storage fabric and 10 GbE LAN both reported 94%. Five weeks of investigation revealed four independent throughput deficits. Three of the four turned out to be defects in the measuring apparatus or its plumbing, not in the fabric itself.

The headline plateau reconstructs arithmetically from the benchmark configuration. The test harness used eight parallel TCP streams; each stream, being single-threaded, topped out at roughly 14–18 Gbps per core. Eight streams at 14 Gbps yields 112 Gbps, and eight at 17.5 Gbps yields 140 Gbps. The measured values were 112.01 Gbps and 139.65 Gbps. Once stream count was raised from 8 to 32, all six directed paths reached approximately 198 Gbps with sub-millisecond round-trip times and zero packet loss.

A second deficit, storage traffic running at 9.4 Gbps on nominally 25 Gbps links, traced to four routing directives that had been placed outside their interface stanza in a configuration file. They never executed at boot; traffic fell back to a 10 Gbps onboard NIC and saturated it. A third deficit, a NIC capped at 58 Gbps, traced to a chipset-connected PCIe slot hardwired at x4. The NIC was moved to a CPU-direct x16 slot and throughput rose 3.4-fold.

The fourth deficit, inbound traffic to one node degrading to 194–196 Gbps with 238,000–280,000 retransmits per 30-second run, is a genuine host-level receive-path defect. Pause frames, softnet time squeezes, and the absence of any CRC or PHY errors point at host contention rather than cabling or link corruption. Two runtime tuning attempts made things worse and were reverted. The defect remains unresolved.

1. The setup and the symptom

Three nodes sit on a 200 GbE fabric, each fitted with a Mellanox ConnectX-6 NIC. All ports negotiated at 200000 Mb/s with MTU 9000. Node A is a Threadripper PRO workstation with two Blackwell GPUs (192 GB VRAM), 64 threads, and 503 GB RAM; it runs heavy resident workloads including an agent harness, Docker, and vLLM. Node B is a compute node with one Blackwell Max-Q (96 GB VRAM), 64 threads, and 251 GB RAM. Node C is an EPYC compute node with no GPU, 128 threads, and 503 GB RAM. A separate 25 GbE storage fabric connects the compute tier to four storage nodes through a spine switch, and a 10 G / 2.5 G LAN carries management traffic.

The instruments were iperf3 for TCP throughput with parallel streams, ib_write_bw for RDMA/RoCE, lspci -vvv for PCIe link state, ethtool counters, /proc/net/softnet_stat, and a containerised network-diagnostics dashboard running a 43-check battery.

The symptom was a plateau. The 200 GbE fabric sat at 112–139 Gbps, which is 56–70% of line rate. The 25 GbE storage fabric ran at 94%. The 10 GbE LAN ran at 94%. Every hardware hypothesis was eliminated:

Check	Result	Verdict
PCIe negotiation	Gen4 x16 on all three nodes (256 Gbps)	not the bottleneck
Link speed	200 Gbps negotiated on all CX-6 ports	not the bottleneck
MTU	9000 jumbo on all 200 G interfaces	not the bottleneck
TCP buffers	256 MB max on Node A and Node B	well-tuned
Ring buffers	maxed at 8192 RX / 8192 TX on all nodes	not the bottleneck
Queue count	63 combined queues on all nodes	not the bottleneck
NUMA topology	single NUMA node on Node A, no cross-socket penalty	not the bottleneck
Offloads	TSO, GSO, GRO all enabled	not the bottleneck

The fabric looked healthy by every measure except the one that mattered.

2. The arithmetic reconstruction

The benchmark script set IPERF_PARALLEL=8. Each iperf3 TCP stream runs in a single thread and cannot exceed what one core can push, which on this hardware is roughly 14–18 Gbps depending on the node and its resident load. Multiply the stream count by the per-stream ceiling and you get the plateau:

- 8 × 14 Gbps = 112 Gbps. Measured Node B → Node A: 112.01 Gbps.
- 8 × 17.5 Gbps = 140 Gbps. Measured Node B → Node C: 139.65 Gbps.

The 112-versus-139 spread is attributable to Node A's heavier resident workloads reducing per-core throughput. The 25 GbE and 10 GbE links needed only 3–6 Gbps per stream to saturate, well inside single-core capacity, which is why the same stream count reported 94% there and 56–70% here. The per-stream ceiling was invisible until the link outran a single core.

This is the strongest single artefact in the investigation. A plateau that reproduces to three significant figures from the benchmark's own configuration is a benchmark artefact, not a fabric limit.

3. The true baseline

On 21 March, with stream count raised from 8 to 32, all six directed paths reached approximately 198 Gbps:

Path	Throughput
Node A → Node B	~197.97 Gbps
Node B → Node A	~198.00 Gbps
Node A → Node C	~198.05 Gbps
Node C → Node A	~197.62 Gbps

Path	Throughput
Node B → Node C	~197.91 Gbps
Node C → Node B	~197.62 Gbps

Ping loss was 0% on every path. Round-trip times were sub-millisecond throughout.

Earlier single-node validation on 8 March, after Node A's NIC was moved to a PCIe riser, had already shown the riser innocent. The link reported Speed 16GT/s, Width x16. RDMA write bandwidth reached 193.25 Gbps. An 8-stream iperf3 aggregate reached 197.3 Gbps with per-stream values ranging from 14.1 to 28.2 Gbps. The riser was not the problem. The stream count was.

One caveat: the 21 March baselines were measured with a soak harness in which one run timed out. That path was re-verified healthy by directed testing, but the timeout means the soak harness itself had an intermittent failure mode.

4. The orphaned routing stanza

Storage traffic ran at approximately 9.4–9.46 Gbps on nominally 25 Gbps links. Node B's /etc/network/interfaces carried four post-up ip route add directives for the four storage-node addresses, intended to route them over the CX-6 port toward the spine. The directives sat orphaned outside any interface stanza, on lines 30–33, and never executed at boot. All storage traffic fell back to the default path over a 10 Gbps onboard NIC and bottlenecked at line rate for that NIC.

Compounding it: Node B's CX-6 was at MTU 9000 while all four storage nodes' 25 GbE NICs were at MTU 1500.

The fix was to move the four directives inside the correct interface stanza (now lines 25–28, persistent across reboots), add a secondary LAN address for correct source routing, apply routes live without reboot, and set MTU 9000 on all four storage nodes. Jumbo frames were verified end-to-end through the spine with ping -M do -s 8972.

Path	Before	After
Node B → storage node 1 (25 G)	9.46 Gbps	23.57 Gbps
Node B → storage node 2 (25 G)	9.46 Gbps	23.56 Gbps
Node B → storage nodes 1–4 (25 G)	9.4 Gbps	24.79 Gbps (peak)
Node B → Node A (200 G)	15 Gbps	121.6 Gbps
Node B → Node C (200 G)	16 Gbps	130.4 Gbps
Node B → Node A (LAN)	FAIL	2.36 Gbps (2.5 G NIC, 94% of line)
Node B → monitoring tier (1 G)	FAIL	0.95 Gbps

The diagnostics dashboard went from 21 of 43 checks passing to 43 of 43. All 22 original failures were dashboard bugs, not infrastructure faults: curl absent from the container image (rewritten to use aiohttp), two wrong node addresses hardcoded, a pod binding the wrong host's 200 GbE address, a read-only SSH directory breaking host-key checks, an unreachable stub resolver, and a _running flag left latched by client disconnect (fixed with try/finally). The container pod's CPU limit was raised from 500m to 4 cores because the harness itself could not drive the link.

The 21/43 → 43/43 pass count measures the harness, not the fabric. It must never be quoted as a fabric improvement.

5. The receive-path defect that remains

One deficit is real. Inbound traffic to Node C degrades in a directional and repeatable pattern. Node C transmits cleanly; receiving at line rate from either peer degrades:

Direction	Throughput	Retransmits per 30 s run
Node A ↔ Node B (either direction)	~197.98 Gbps	essentially zero
Node C → Node A	~197.60–197.65 Gbps	0–3
Node C → Node B	~197.60–197.65 Gbps	0–3
Node A → Node C	~195.43–196.53 Gbps	~255,000–272,000
Node B → Node C	~194.04–196.32 Gbps	~238,000–280,000

Supporting evidence: on every noisy inbound run, Node C emitted large bursts of pause frames (tx_pause_ctrl_phy), typically 20,000 to 51,000 per run. The softnet time_squeeze counter incremented intermittently on Node C during inbound saturation. No CRC, PHY, or physical discard errors were observed on the affected link, which excludes cabling and low-level link corruption and points at host receive-path contention plus link-level flow control.

Two reversible runtime tuning attempts produced negative results. Raising net.core.netdev_budget, net.core.netdev_budget_usecs, and net.core.dev_weight dropped inbound throughput to approximately 191.8–193.6 Gbps while retransmits stayed high. Lowering RX coalescing and disabling adaptive RX dropped inbound throughput to approximately 191.2–192.7 Gbps while pause-frame bursts persisted. Both were reverted to baseline; no persistent configuration was changed. Large Receive Offload (a NIC feature that aggregates incoming packets before the kernel sees them) was subsequently enabled on all three CX-6 interfaces and persisted via a networkd-dispatcher hook.

The defect is unresolved. Cabling and link corruption are excluded. The two attempted mitigations failed. The remaining hypotheses, pause flow control configuration, IRQ and queue-to-core placement, NUMA affinity, and BIOS or PCIe power management differences against the peers that do not exhibit the defect, were planned but not tested inside this measurement window.

6. RDMA versus single-stream TCP

On the same link and the same day, RDMA write bandwidth reached 193.25 Gbps while a single iperf3 TCP stream reached 47.1 Gbps (315 retransmits) to one peer and 55.6 Gbps (0 retransmits) to the other. A single stream with a 4 MB window reached 61.7 Gbps (7.7 GB/s). RDMA, which bypasses the kernel's TCP stack entirely and writes directly to remote memory, is not subject to the per-core TCP ceiling. Userspace TCP needs stream count to reach the same place.

7. The April slot finding

A fabric test on 16 April showed Node A → Node B capped at 58 Gbps while Node A → Node C was fine at 198 Gbps. Node B's CX-6 was running at PCIe 4.0 x4, downgraded from x16, for two reasons.

First, the NIC sat in a chipset-connected slot whose root port is hardwired at x4. That slot shares its link with an onboard 10 GbE NIC, a second NIC, WiFi, USB, and SATA. Second, the ASUS Pro WS TRX50-SAGE WIFI presents five physical x16 slots, but only two are electrically x16 Gen5 CPU-direct; the rest are x8, x4, or chipset-attached. The board's physical appearance does not

match its electrical topology.

The fix was to rearrange cards on Node B: slot 1 GPU (PCIe 5.0 x16, unchanged), slot 2 NIC (PCIe 5.0 x16, moved off the chipset slot), slot 3 empty, slot 4 IPMI (chipset x4, adequate), slot 5 NVMe adapter (PCIe 5.0 x16). A BIOS bifurcation change was also required: the Gen5 slot from RAID Mode to x16 Mode, the Gen4 slot to RAID Mode for the relocated NVMe adapter. The NIC came back at PCIe 4.0 x16 and all 7 NVMe drives were detected.

Path	Before	After
Node A → Node B	58 Gbps	198 Gbps
Node A → Node C	198 Gbps	198 Gbps
Node B → Node A	55 Gbps	195 Gbps
Node B → Node C	57 Gbps	178 Gbps
Node C → Node A	155 Gbps	176 Gbps

Node B's links improved 3.4-fold. The inbound-to-Node-C paths remain at 176–178 Gbps, which is the known receive-path backpressure described above, not a regression introduced by the slot change.

Two caveats apply. The April slot fix and the March plateau were diagnosed weeks apart; the 3.4-fold April improvement is measured against the x4 state and does not retroactively validate every March number. One node was found hung during the April session and recovered by an out-of-band power cycle; root cause was not established and is unrelated to the slot work.

8. What survived and what did not

Across five weeks and four independent throughput deficits, three of the four were defects in the measuring apparatus or its plumbing: stream count, a dashboard's own container, a routing directive in the wrong stanza, a slot choice. Only one, the inbound-to-Node-C receive-path defect, was a genuine host-level performance defect.

The diagnostic tell throughout was same-instrument-different-verdict. Identical stream count reported 94% on 25 GbE and 56–70% on 200 GbE. Identical routing configuration worked on one node and failed silently on another because of stanza placement. Identical NIC models performed at line rate in CPU-direct slots and at 29% in chipset-connected slots. When the instrument disagrees with itself across conditions, suspect the instrument.

The claims that survive: the fabric is capable of 198 Gbps pairwise throughput in both directions between Node A and Node B, and between Node A and Node C. The storage fabric is capable of approximately 24.8 Gbps aggregate to the four storage nodes once routing and MTU are correct. The per-stream TCP ceiling on this hardware is roughly 14–18 Gbps, and any benchmark that uses fewer than 12–16 streams will underreport a 200 GbE link.

The claim that does not survive: that all paths are healthy. Inbound to Node C remains degraded by approximately 2–4 Gbps with hundreds of thousands of retransmits per 30-second run. The defect is real, it is localised to one node's receive path, and the root cause is not yet established.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. PT-TN-2026-007. Measurements taken 8 March 2026 – 16 April 2026. Raw artefacts are retained.