PURETENSOR
INTELLIGENCE DISTILLED

The 400G Link Was Fast at Idle and Slow Under Load

RDMA at line rate, TCP at 12 to 30 percent of it, and a host reporting 8.3 million PAUSE frames behind a sr

Paper: PT-R-2026-065 v1.0

Author: PureTensor Ltd

Date: 3 November 2026

Status: Published

Abstract

On 27 October we joined two workstations with a direct 400 Gb/s link, a ConnectX-7 network card in each, and on 2 November we measured it. With RDMA (remote direct memory access, which moves data between NICs without the kernel's TCP stack or host copies) the idle link carried 393.04 Gb/s one way and 392.03 the other, 98.3 and 98.0 percent of line rate. With TCP, one direction reached 380.9 Gb/s and the other 302.1. The wire, the optics, the PCIe Gen5 x16 slots and the NIC firmware were not the limit at idle.

Then we ran a ten-minute stress test: all-core CPU load and a staged memory load on three workstations at once, with bidirectional TCP on the 400G and 200G links. Over 600 seconds the 400G link averaged 117.85 Gb/s from A to B and 46.93 Gb/s from B to A, which is 29.5 and 11.7 percent of line rate and 30.9 and 15.5 percent of the idle TCP figures. The 200 Gb/s link from B fell to 69.80 and 78.00 Gb/s, but that link had a ceiling of its own, and against its one-way idle figures of 115.1 and 112.3 the loss is smaller: 60.6 and 69.5 percent remained.

Two host defects turned up on the way. The receiving port of the 400G link on one workstation ran a 1,024-entry receive ring while every other port ran 8,192, because the boot-time tuning predated the link and never resized it. That host reported 8.33 million PAUSE frames (the Ethernet flow-control message a receiver sends to ask its peer to stop) sent since its morning boot, one unrecorded reading. Separately, the 200G card on a third workstation was capped near 114 Gb/s; a firmware lane-split setting was removed and the cap went away, taking RDMA on that link from 113.6 to 195.6 Gb/s, a before-and-after with a reboot between the runs.

The early internal explanation had been that TCP was CPU-bound under load, and that the ring explained the PAUSE storm during the stress. The raw data does not establish either. This paper reports what held and what did not, and Section 7 sets out exactly which claims survive.

1. The Question After a Clean Install

The two workstations in this paper are called A and B. A has a Threadripper PRO 9975WX (Zen 5) and one ConnectX-7, its only fabric NIC, which replaced an earlier 200 Gb/s ConnectX-6. B has a Threadripper 7970X (Zen 4), one ConnectX-7 and one ConnectX-6. A third workstation, C, has an EPYC 7713 (Zen 3) on a server board and one ConnectX-6. The 400G link runs from A to B directly: Ethernet mode, firmware 28.46.3048 on A, PCIe Gen5 x16 at each end, 850 nm

multimode 400G optics, MTU 9000. B forwards A's traffic onward through its own 200G ConnectX-6, so A reaches C through B. We call B to C the 200G link. It uses passive copper, so there are no optical diagnostics to read.

RDMA ran at line rate, then a ten-minute stress run showed TCP on the same links at a fraction of that. We wanted to know what limited the link, and which limits lived in the hosts rather than the network.

We have published on 200 GbE fabrics before and do not retell that here. New are the first public 400G measurement from us, a load result we could not fully explain, and a slot-width cap that went away with a firmware setting.

2. How We Measured

For TCP we used iperf3 3.20. For RDMA writes we used ib_write_bw from perftest 6.20. The idle benchmark ran the same script for each case. A TCP case was four iperf3 processes in parallel with four streams each (16 streams in total), 10 seconds, summing the received bits per second. An RDMA case was one ib_write_bw with 8 queue pairs, RoCE v2 (RDMA carried over routable Ethernet) on one fixed GID index, 65,536-byte messages, 10 seconds. Each case ran once per label. There are no repeats and no confidence intervals anywhere in this paper.

Tuning in place was MTU 9000, TCP buffers of 256 MB, adaptive interrupt coalescing, and rings of 8,192 entries on every NIC except B's 400G port (Section 5).

The burn-in on 2 November ran an all-core CPU stress and a memory stress ramped in three stages, at nice 19 (the lowest scheduling priority) on A, B and C at the same time, for 600 seconds. A and B started at 10:06:27 and 10:06:23 UTC, C at 10:07:18. The stress ended on the three hosts at 10:16:29, 10:16:26 and 10:17:26. GPUs were not stressed. Sensor and NIC counters were sampled every 5 seconds. The thermal watchdog limits never fired.

Alongside the stress ran bidirectional iperf3 (the --bidir mode, 600 seconds) on each link: four processes of four streams on the 400G link with A as client, started at 10:07:13, and four of four on the 200G link with B as client, started at 10:07:27. The iperf3 processes were not niced. It matters in Section 4.

3. The Idle Link: RDMA at Line Rate, TCP Short in One Direction

Path	TCP, 4 x 4 streams	RDMA, 8 QP
400G, A to B	380.9	393.04
400G, B to A	302.1	392.03
200G, B to C	115.1	113.62
200G, C to B	112.3	113.69
A to C via B (forwarded)	115.4	not measured
C to A via B (forwarded)	112.2	not measured

Table 1. Idle throughput in Gb/s on 2 November, one run per cell.

On the 400G link RDMA reached 393.04 and 392.03 Gb/s, which is 98.3 and 98.0 percent of line rate. That settles the physical side at idle: the optics, both Gen5 x16 slots (width taken from the install report, not re-read in the retained records) and the firmware move 400G when the kernel's TCP path and the host copy are out of the way. RDMA was not run under the stress, and the result says nothing about host-stack cost.

TCP did less. A to B reached 380.9 Gb/s (95 percent of line rate), B to A only 302.1 (76 percent). The asymmetry is real in this data set, in that the same script and layout gave it. We do not know its cause. An early note attributed it to receive-side flow hashing collisions, where several streams land on the same receive queue and CPU core, and inferred that from a different stream layout that we did not retain results for; we leave it as a hypothesis. Another guessed at sender CPU on B. Neither explanation was isolated by a controlled test. What we can say is that B to A moved 302.1 Gb/s on 2 November, and that a later run after the ring change gave 284.4 (Section 5), which is the same order and, with one run each, not distinguishable.

Both forwarded paths through B matched the 200G link's own figures, 115.4 and 112.2 Gb/s against 115.1 and 112.3, so the 200G hop limits any path touching C (Section 6).

4. Under Load: 600 Seconds of Bidirectional TCP

Link	Direction	Idle TCP	600 s mean under load	Share of idle	Share of line rate	Sender retransmits
400G	A to B	380.9	117.85	30.9 %	29.5 %	75,966
400G	B to A	302.1	46.93	15.5 %	11.7 %	301
200G	B to C	115.1	69.80	60.6 %	34.9 %	1,271,049
200G	C to B	112.3	78.00	69.5 %	39.0 %	2,074,654

Table 2. The 600 second loaded run, Gb/s received, summed over the four iperf3 processes of each link. Retransmits are summed over the sender's four processes.

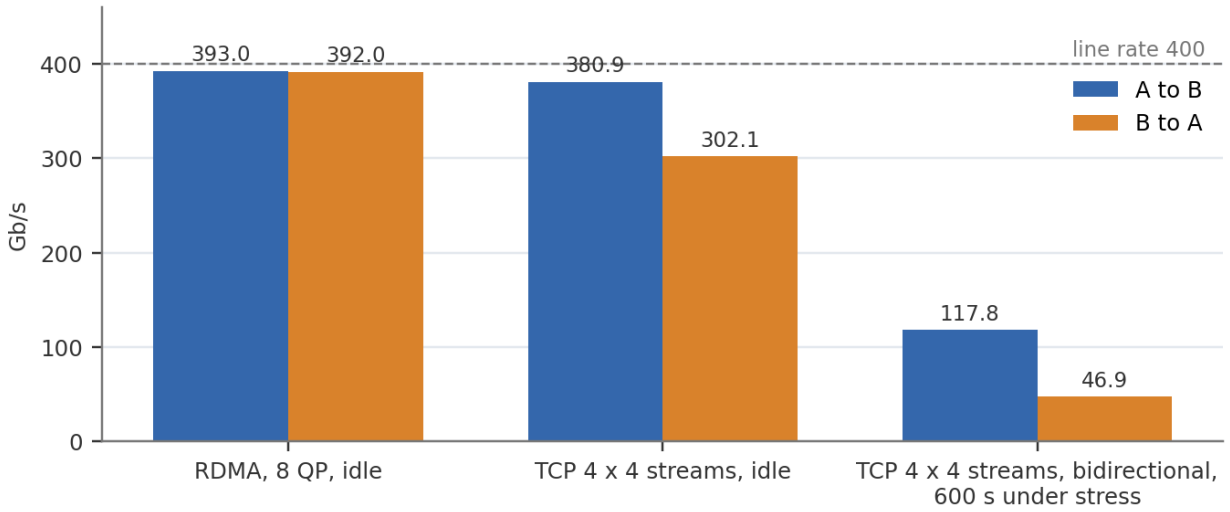


Figure 1. The 400G link at idle (RDMA, then TCP) and under 600 s of stress with bidirectional TCP. Values are in Tables 1 and 2.

The 400G link ran at a third or less of its idle TCP figure in A to B and at about a sixth in B to A. Per process, A to B delivered 30.24, 29.96, 27.02 and 30.63 Gb/s; B to A delivered 11.70, 11.92, 11.95 and 11.37. The NIC counters are within about 8 percent of the iperf3 sums (108.96 against 117.85 Gb/s; 43.43 against 46.93): A transmitted at an average of 108.96 and received at 43.43, with a peak 5-second transmit of 141.18; B received at an average of 108.21, transmitted at 43.09, and peaked at 142.79 receive.

B's 200G port also carried forwarded traffic during the run, and the percentages below set bidirectional loaded figures against one-way idle ones. The 200G link must be read against its own ceiling. It was already capped near 114 Gb/s (Section 6), so the right comparison is 112 to 115, not 200. Against that baseline the loaded run retained 60.6 and 69.5 percent, not the 35

and 39 percent that a line-rate share suggests. It also retransmitted heavily: 1.27 million and 2.07 million segments over the run, against 75,966 and 301 on the 400G link.

Minute window (s)	A to B	B to A
0-60	125.5	50.0
60-120	120.3	46.9
120-180	117.8	45.1
180-240	114.8	44.0
240-300	117.0	43.9
300-360	112.9	46.9
360-420	109.6	51.2
420-480	112.8	48.8
480-540	116.5	43.6
540-600	131.0	49.0

Table 3. Mean 400G throughput in Gb/s per 60 second window, from one-second iperf3 intervals summed over the four processes.

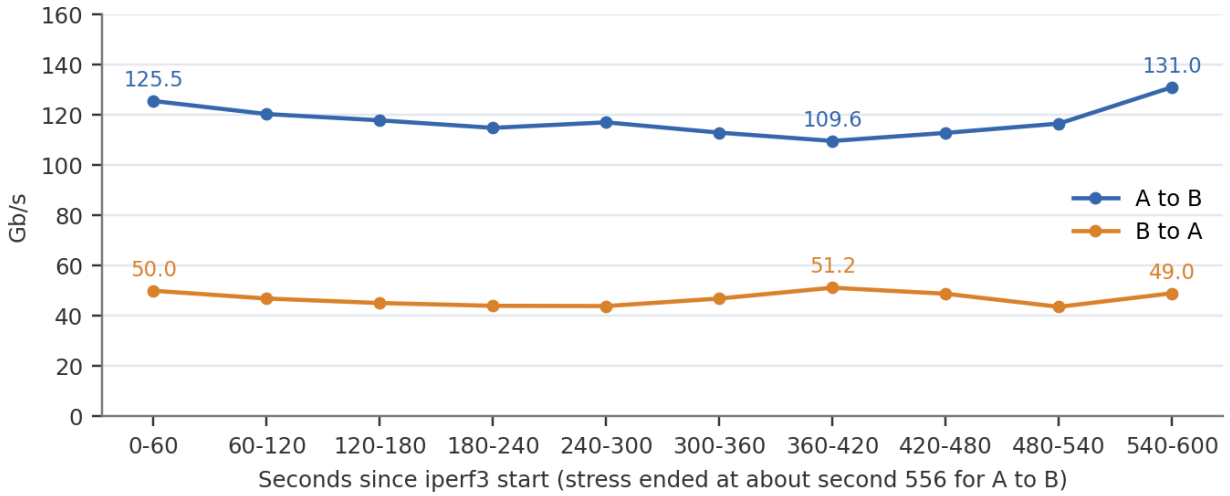


Figure 2. Per-minute 400G throughput during the loaded run. Values are in Table 3.

The working explanation was that TCP is CPU-bound under load: the hosts were saturated, so the kernel's TCP path starved. The raw intervals do not establish it, for four reasons.

First, the stress ran at nice 19 and the iperf3 processes ran at normal priority, so the scheduler would be expected to favour the network test; we examined no per-process CPU data. Second, the stress on A and B ended at about second 556 and 553 of the 400G iperf3 window, and B's at about second 539 of the 200G window. The 400G clients started 46 s (A) and 50 s (B) after the stress began on those hosts, the 200G client 64 s after it began on B, so nearly the whole window was under stress and the last stretch was not. In the final 40 seconds, with the stress over, the 400G link averaged 134.0 Gb/s from A to B and 51.4 from B to A. That is better than the loaded mean, and the final window in Table 3 shows the bump, but it is still only 35 and 17 percent of the idle TCP figures. If the CPU stress alone had held the link down, removing it would be expected to restore much more than this, though the last 40 seconds are 4 percent of one run and also include recovery. Third, there was no bidirectional idle control: the idle benchmark was one direction at a time, while the burn-in was bidirectional with four processes of four streams. Load and bidirectionality are confounded. Fourth, the receive-ring fix had not

been applied during the burn-in, so B's undersized ring (Section 5) may have contributed.

The honest statement is narrower: TCP ran far below its idle figure while the hosts were stressed, and the cause was not isolated among CPU contention (including shared caches, memory bandwidth and SMT siblings, the two hardware threads that share one core), bidirectional traffic, and the small ring. The retransmit counts (75,966 from A against 301 from B) fit an unhappy receiver on B, but a fit is not a test.

5. A Receive Ring That Never Got Resized

B's 400G port ran a receive ring of 1,024 entries. Every other NIC in the fabric ran 8,192. The boot-time NIC tuning service on B was installed before the 400G link existed, so it never resized that port, and nothing flagged the omission. The fix, found on 2 November, was to raise the receive and transmit rings to 8,192 live at about 10:50 UTC and to persist the setting as a drop-in of the tuning service. The link stayed up through the change.

The counters read before the fix, since that morning's boot, showed B had sent A 8.33 million PAUSE frames and logged 22.8 thousand out-of-buffer drops. This figure has weak provenance and we put it plainly. It is a single reading. B had been powered off and on that morning, around 07:00 UTC, for an unrelated maintenance window, and the exact boot time was not recorded. The counter output was not saved. The count therefore spans roughly the morning boot to about 10:50 UTC, which includes the burn-in, the idle baseline benchmark and whatever else crossed the link. The data cannot say how many of the 8.33 million belong to the 600 second run. Our first write-up said the frames were sent under load. That is not supported.

The investigating notes record that after the ring change B sent no PAUSE frames during the idle re-test and that a few out-of-buffer drops remained at line rate; no counter output was kept, so treat both as reported, not measured. The baseline cases ran before the change and the ring8192 cases after. Idle TCP from A to B read 388.1 Gb/s (it was 380.9) and from B to A 284.4 (it was 302.1). Neither difference is a claimable effect with one run each. We did not repeat the loaded run after the fix, so whether the small ring explains any of the loaded shortfall is unknown.

6. The 200G Cap and Its Repair

Before any repair, the ConnectX-6 on C measured RDMA at 113.62 Gb/s from B to C and 113.69 from C to B, with TCP at 115.1 and 112.3. Every path touching C was capped near 114. The 2 November notes say the card had negotiated PCIe Gen4 x8, reported as downgraded, with a root port advertising only x8. The retained PCI dump has no link-status lines, so that width is unverified from raw records. The notes put a Gen4 x8 link at the 512-byte maximum payload size at about 114 to 116 Gb/s of usable data (we did not compute this; the raw rate is 126), which would match the measured ceiling.

We tried raising the maximum read request size from 512 to 4,096 bytes on both ends of the 200G link. RDMA read 113.74 and 113.99 Gb/s against the baseline of 113.62 and 113.69, and TCP read 115.1 and 113.3. No gain, so we reverted it. The maximum payload size was already at its 512 limit.

What changed was a BIOS lane-split setting (bifurcation, which divides one physical slot's lanes among several devices), and the cap went away. On 3 November a local dump of the board's firmware settings showed six of the seven slots set to four-way x4x4x4x4 splits and one already x16. The remote inventory had said otherwise, and the board manager's remote settings interface was months stale and refused writes, so we used the vendor's local BIOS tool. We changed three slot widths (the NIC's slot and two empty ones) to x16; the three slots with NVMe adapter cards stayed split. The change followed a controlled drain and a clean shutdown of two

databases, with a reboot at 12:42:22 UTC. A comparison of all 780 firmware questions before and after showed exactly three value changes, and every drive, array and service came back.

State	B to C	C to B
Baseline, 2 November	113.62	113.69
Larger read request size, reverted	113.74	113.99
After lane split removed, 3 November	195.59	195.67

Table 4. RDMA on the 200G link in Gb/s. The final row used the same script and parameters, run once per direction at 12:45 to 12:46 UTC.

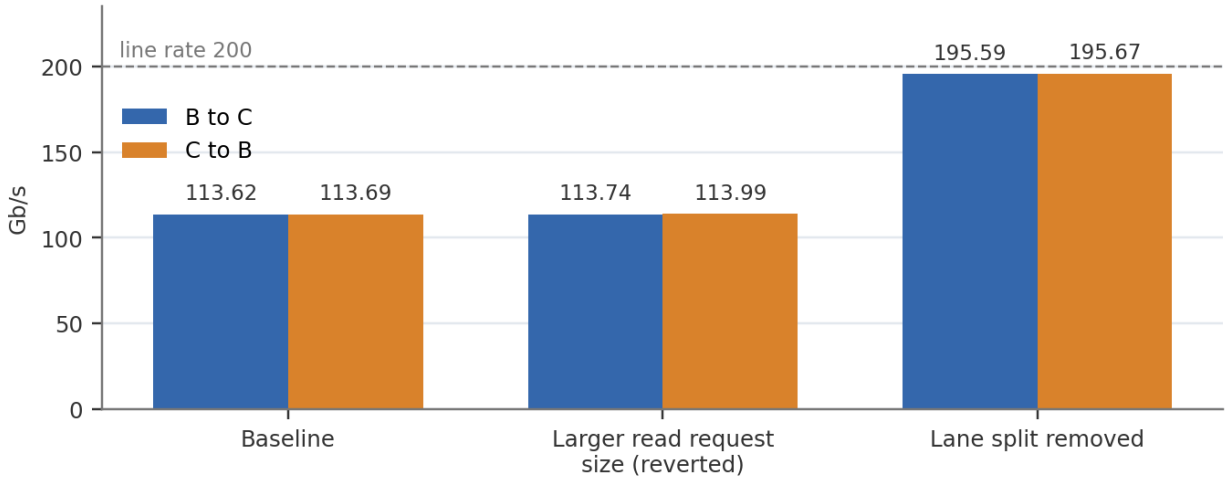


Figure 3. RDMA on the 200G link before and after removing the lane split. Values are in Table 4.

After the repair RDMA read 195.59 and 195.67 Gb/s, 97.8 percent of 200G in both directions and 1.72 times the pre-repair 113.6. The PCIe link read Gen4 x16. This is a before-and-after on the same script and parameters, with a reboot and a changed bus address and device name between the two runs, so we cannot rule out that the card was reseated or moved, and a four-way x4 split would by itself give x4, not x8, so the mechanism is not fully shown. What the records show is 113.6 before, 195.6 after, and Gen4 x16 read afterwards. We recorded no TCP result after the repair.

One part of the story does not reconcile. The 2 November notes place the card at one bus address and Gen4 x8 (the sensor log confirms only the address). The notes from the morning of 3 November place it at a different bus address and Gen4 x4. A figure of 114 Gb/s is not possible over Gen4 x4, whose raw rate is 63 Gb/s, so the measurement points to x8 rather than x4. The card's bus address and its RDMA device name both changed between the two days, and the records do not show whether the card was moved, a firmware setting renumbered the topology, or one note is wrong. 114 Gb/s is consistent with x8 and not with x4; we treat the pre-repair width as unresolved.

This differs from our earlier slot-width finding, where the card sat in a chipset-attached x4 slot. Here the lanes were lost to a firmware setting on physically wide slots, and a remote inventory misreported it.

7. What These Numbers Will Not Carry

Every figure is a single run unless stated, and variation between runs is unmeasured. The differences between 381 and 388 Gb/s, and between 302 and 284, should not be read as effects. The 600 second loaded run is one event, bidirectional, with stress and iperf3 start times

offset by 46 to 64 seconds, compared against idle runs that were unidirectional. The hosts are production machines.

The PAUSE figure is one summary number from an unrecorded boot, read before about 10:50 UTC, and the no-PAUSE and out-of-buffer observations after the fix are also notes, with no counters retained. The ring change has no loaded-run re-test. We have no TCP number for the repaired 200G link, and the idle B-to-A shortfall has no cause beyond an inferred hashing collision. The physical layer of the 200G copper links was not investigated.

The claims that survive are these. RDMA reached 98 percent of line rate in both directions on the idle 400G link. Idle TCP read 380.9 and 302.1 Gb/s. Loaded bidirectional TCP read 117.85 and 46.93 Gb/s on the 400G link and 69.80 and 78.00 on the 200G link. The 200G slot cap was real and was repaired, from 113.6 to 195.6 Gb/s. The claims that do not survive are that TCP was proven CPU-bound as a cause, that the PAUSE frames belong to the load run, or that any TCP number exists for the repaired 200G link.

8. What We Changed

B's 400G receive and transmit rings are now 8,192 and persistent. NIC tuning written before a new link existed silently skips it, so a tuning service should name the property it enforces (every fabric port at 8,192) and be checked against the port list. The lane split on C's NIC slot is removed, via the local firmware route because the remote interface could not write. The larger read request size was reverted. We will not use iperf3 TCP as the line-rate probe under CPU load.

The next experiment is a bidirectional idle control and a repeat of the loaded run after the ring fix, with stress and iperf3 starting together at the same priority, saving the PAUSE counter output this time.

9. Methods and Artefacts

TCP throughput came from iperf3 3.20 and RDMA from ib_write_bw in perftest 6.20. The ConnectX-7 firmware is 28.46.3048. The kernel versions in use during the tests were not recorded. Statistics are sums of per-process iperf3 received rates and one-second interval sums averaged per 60 second window. There are no confidence intervals because there are no repeats. Retained raw material comprises the idle benchmark script and results table, the burn-in controller and event logs, the sensor and NIC-counter samples, the twelve raw iperf3 files (600 one-second intervals each), the raw RDMA results of the 3 November re-test, before-and-after dumps of the firmware settings and PCI topology, and the commit that changed the ring size. Not retained are the PAUSE and out-of-buffer counters.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. This paper is PT-R-2026-065. The measurements were taken on 2 and 3 November 2026, the last at 12:46 UTC on 3 November; raw artefacts are retained for all but the PAUSE and out-of-buffer counters and the pre-repair link-width readings.