White Paper

Scaling AI Beyond the Data Center

Why Buffer Depth—Not Distance—Determines Job Completion Time in Multi-Site AI Clusters

TABLE OF CONTENTS

Executive Summary

The Shift to Scale-Across AI

Rethinking Buffer Architecture: Hierarchical, Not Uniform

The Real Metric: Job Completion Time, Not Average Latency

Beyond the Fabric: The Front-End Network

Putting It to the Test: Simulation Methodology

Results: Buffer Depth Determines Completion Time

Engineering the System, Not Just a Switch

Conclusion: A Foundation for the Path to AGI

Executive Summary

As AI models train, they all eventually encounter the same problem: they outgrow the buildings they started in. Power budgets and floor space push GPU clusters across campuses, and soon they'll need to expand across regions. When two GPUs that need to communicate are separated by more than a few kilometers of fiber, the assumptions that provide the foundation for reactive congestion control start to give way. Round-trip times climb into the hundreds of microseconds, feedback loops that used to arrive in time to head off a queue overflow now arrive too late, and a single dropped packet can stall an entire training job far longer than the delay that caused it.

This white paper examines why that scenario happens, and what can be done to mitigate it. Through direct testing using a buffer-sizing simulation spanning 10-km, 30-km, and 100-km links, two congestion-control protocols, and message sizes from 50 MB to 500 MB, one result held up: sizing switch buffers at or above 1.5× the link's bandwidth-delay product (BDP) keeps job completion time within a few percent of ideal, and drives retransmission to zero. Below that 1.5×BDP line both metrics degrade, with retransmission degrading first and completion time close behind. The threshold doesn't move with distance or protocol, and only softens slightly with message size. To put that threshold in concrete terms: at 100T of switch throughput, 1.5× BDP works out to roughly 1.9 GB at 10 km, 5.6 GB at 30 km, and 18.8 GB at 100 km.

The architecture that puts this insight into practice is a hierarchical, hybrid buffering model that keeps shallow, fast on-chip buffers everywhere they're sufficient, and reserves deep HBM buffering as tier-appropriate insurance exactly where distance and failure risk are highest: supported end-to-end by matching NIC and switch capabilities.

1.5× BDP	0%	CC Agnostic
The buffer threshold that keeps completion time near ideal and retransmission at zero, confirmed at 10 km, 30 km, and 100 km.	Retransmission rate at that threshold, for every message size tested from 50 MB to 500 MB.	The threshold holds under both NSCC (the UEC default) and Broadcom RCCC. This is a property of buffer sizing, not a tuning trick.

The Shift to Scale-Across AI

Every large model trained today eventually runs into the same wall: there is only so much power and so much floor space in one building. As clusters grow past what a single facility can host, the collectives that make distributed training work, including AllReduce, AllGather, start crossing building lines, campus lines, and eventually regional lines. This is called scale-across, and it changes the physics of the network in three specific ways:

FIBER ADDS ~100 µs OF ROUND-TRIP TIME FOR EVERY 10 KM OF DISTANCE

- **The Latency Tax:** Fiber adds roughly 100 microseconds of round-trip time for every 10 kilometers of distance. While that sounds like a small amount, remember that PFC and ECN, the feedback loops congestion control depends on, were designed around intra-building round trips measured in single-digit microseconds. Once RTT stretches into the hundreds of microseconds, those loops simply can't react before a queue has already overflowed.
- **Fan-In Bursts:** AI collectives don't produce steady, well-behaved traffic. AllReduce and AllGather generate synchronized fan-in spikes, where many senders converge on one receiver at once. On-chip buffers are shallow by design and tuned for cost and power. A burst like that can exhaust them in microseconds.
- **Failures Are the Norm, Not the Exception:** At this scale, link and path failures aren't edge case; they happen continuously, simply because there are so many links. Even the best proactive routing still has to react once a failure occurs, and at 10 km and beyond, that reaction window is exactly where a network needs headroom the most.

Rethinking Buffer Architecture: Hierarchical, Not Uniform

THE EXCEPTIONAL BURST NEEDS DEEP BUFFERING, NOT EVERY PACKET THAT CROSSES THE FABRIC

The instinct, once this problem is identified, is to reach for the deepest buffer available everywhere. That instinct is understandable but wrong, and the reason why is worth examining.

Why Headroom Matters

AI collectives are barrier-synchronized: every GPU waits for every other GPU before the step is considered done. One dropped packet doesn't just slow one flow, it stalls the entire job. Most traffic, most of the time, runs well below the point of congestion. It's the exceptional burst that needs deep buffering, not every packet that crosses the fabric.

Why Not Deep Buffering Everywhere

Deep buffering solves a real problem, but applying it everywhere isn't the sensible answer. HBM-class buffering costs more, in both silicon and power, than the shallow on-chip buffering used for everyday traffic and congestion is the exception, not the norm. Paying that premium on every port, all the time, to guard against a burst that only shows up occasionally is a poor trade.

The more sensible design lets on-chip buffering (OCB) handle the common case, and reserves deep buffering for the tier and traffic pattern where it actually earns its cost: the exceptional burst or failure event, not everyday operation.

A Common Objection: Doesn't a Full Buffer Just Move the Delay?

This is a fair question that's worth answering directly: if a buffer fills up, doesn't draining it just push the delay further down the path instead of removing it? For a buffer that's engaged constantly on every packet, that concern is real: a permanently deep queue adds serialization delay on the way out, and in the worst case that delay can rival the round-trip time that made the buffer necessary in the first place.

That's precisely why this architecture doesn't place a deep buffer in the everyday path. The HBM buffering used here is standby insurance, not a queue that traffic routes through by default. It engages only for the exceptional burst or failure event, at the one tier (region-to-region) where the alternative, which is a dropped packet and a full retransmission, costs far more than any drain time. Intra-building traffic and building-to-building traffic never touches it. Buffer-induced delay is a genuine design constraint; it's the reason for a tiered model rather than an argument against deep buffering anywhere.

Three Tiers, Applied Where They're Needed

In production today, this architecture already looks hierarchical by design:

- Intra-building traffic: Shallow buffer, no oversubscription
- Building-to-building traffic (fewer than 10 km): Shallow buffer, modest oversubscription, managed by telemetry
- Region-to-region traffic (10 km and beyond): HBM deep buffer, significant oversubscription, multiple round trips of headroom, all reserved for exactly this tier

DEEP BUFFERS ARE TIER-APPROPRIATE INSURANCE, APPLIED PRECISELY WHERE ROUND-TRIP TIME AND FAILURE RISK ARE HIGHEST

Deep buffers aren't the default path. They're tier-appropriate insurance, applied precisely where round-trip time and failure risk are highest.

Table 1: Buffer Math for a 10-km Regional RoCE Link[a]

Parameter	Intra-building Traffic	Building-to-Building Traffic	Region-to-Region Traffic
Link Speed	800 Gb/s		
RTT (10-km reference)	~100 µs		
Required Buffer/Port (BDP)	~10 MB		
Actual On-Chip Buffer (fixed hardware)	~200 MB		
HBM Capacity Available	up to 8.0 GB		

a. The fixed on-chip buffer already exceeds the bandwidth-delay product at this distance. HBM is the added tier that gives headroom for the burst and failure cases described above.

The Real Metric: Job Completion Time, Not Average Latency

It's tempting to optimize a network for average latency or jitter. For AI training, that's the wrong target. Because collectives are barrier-synchronized, job completion time (JCT) is set by the slowest path, not the mean one, and jitter around the average isn't the dominant driver of JCT at this scale.

What is dominant: a single dropped RoCE packet stalls the entire collective until retransmission completes, and every GPU in the job idles for that whole duration. Absorbing a transient burst in a queue costs microseconds. A drop-and-retransmit costs orders of magnitude more. That asymmetry is the whole argument for deep buffers, and it's exactly what this simulation was designed to quantify.

Beyond the Fabric: The Front-End Network

The AI fabric itself is the tightly-tuned, purpose-built RDMA network that connects GPUs to each other. But every AI fabric also sits behind a front-end network, stitching together local storage, general-purpose compute, cloud storage, and the outside world for training data, checkpointing, and inference traffic.

That front end is a messier environment: rather than one tuned protocol, multiple transport protocols share the same links, including TCP in its many flavors, RDMA, NVMe, UEC, and IPSec. Flows are often on a single path rather than sprayed across many paths, so one flow's burst has nowhere else to go, and destinations vary widely in distance rather than sitting at one uniform regional RTT. These protocols aren't tuned for optimal congestion control the way purpose-built RDMA is, and tunnel encapsulation and encryption add their own overhead. Deep buffers play the same insurance role here, and they are applied to a less-controlled, more heterogeneous mix of traffic.

Putting It to the Test: Simulation Methodology

CONGESTION COMES FROM A DELIBERATE 2:1 OVERSUBSCRIPTION

To move from architectural reasoning to a proven model, a buffer-size sweep was completed. The setup was intentionally simple, so the result would be easy to trust:

- The test traffic was straightforward DC-to-DC traffic across a 2:1 oversubscribed long-haul link, with no local traffic to complicate the picture.
- The buffer size was swept from 3× BDP down to 0.005× BDP at each distance.
- The sweep was repeated independently at 10-km, 30-km, and 100-km round-trip distances.
- Two congestion-control protocols were compared under the same conditions: NSCC (UEC default) and RCCC (Broadcom), and each was tested with default and tuned parameters.
- A companion sweep with varying message size from 2 MB up through 200 MB, and later extended to 500 MB, was completed to confirm that whatever threshold emerged wasn't an artifact of one particular message size.

How Congestion Was Created

Congestion came from a deliberate oversubscription, not from adding extra competing flows. The long-haul link was provisioned at half the rate the sender's port could otherwise push—a 2:1 oversubscription—so sustained demand outstripped available capacity by design, independent of buffer size. That persistent gap between offered and available bandwidth is what builds a queue at the switch; shrinking the buffer at each sweep point only changes how much of that queue it can absorb before packets are dropped, which is what drives the retransmission and completion-time curves in the Results section of this white paper.

A Note on Scope

This sweep isolated the buffer-depth question by testing DC-to-DC traffic across a single long-haul link, deliberately without competing local traffic, so the threshold result wouldn't be muddied by any other variable. The fan-in bursts described previously are a many-to-one pattern that was not part of this sweep. The same threshold is expected to hold (or tighten) under incast; concurrent senders only add to the effective burst a buffer must absorb, but that configuration is the next step in this line of research, rather than a result that is being reported in this white paper.

Results: Buffer Depth Determines Completion Time

The clearest way to see the result is the message-size sweep at 10 km under NSCC, shown below across three flow sizes — 50 MB, 150 MB, and 500 MB — tracking flow completion time (FCT), the per-flow measurement used throughout this simulation and the practical stand-in for the job completion time (JCT) introduced earlier in this paper. The 1.5× BDP line is marked in each panel, and the shallow buffer zone, the range where today's on-chip-only buffers typically sit, is marked as well.

Figure 1: Message-Size Sweep at 10 km under NSCC, across 50-MB, 150-MB, and 500-MB Flow Sizes

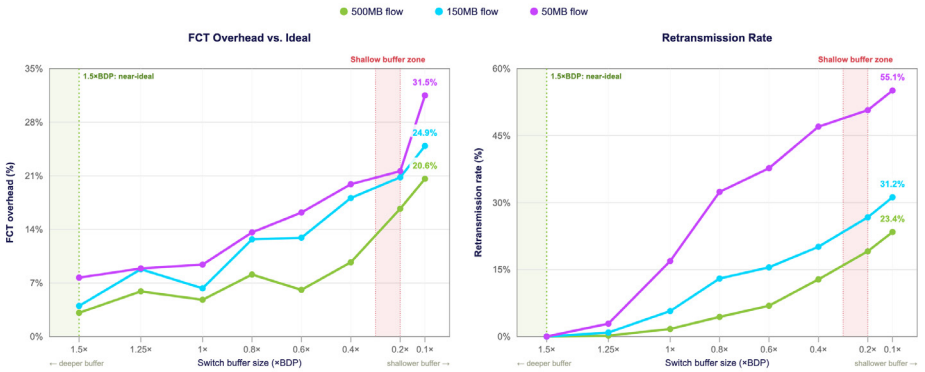


Figure 1 shows FCT overhead and retransmission rate as the switch buffer size is swept from 1.5× BDP down to 0.1× BDP, at 10 km under NSCC. Above the 1.5× BDP line, both metrics are flat and near zero for every message size. Below it, retransmission climbs first, and completion-time overhead follows sharply, once buffer size falls into the shallow-buffer zone.

Two things stand out. First, the curves are well-behaved above threshold, with near-ideal completion time and essentially no retransmission for all three flow sizes. Second, below threshold, the smallest flow (50 MB) degrades earliest and hardest: at 0.1× BDP it is losing 31.5% of its ideal completion time to overhead, with a 55.1% retransmission rate. Larger flows are more forgiving of a shallow buffer, simply because they have more packets to average the pain over, but they're not immune: even the 500 MB flow loses over 20% of its completion time at the shallowest buffer size tested.

The Threshold Holds Across the Full Message-Size Range

AT 1.5× BDP, RETRANSMISSION WAS 0% FOR EVERY MESSAGE SIZE TESTED

The message-size sweep was extended further, from 50 MB up to 500 MB in 25 MB steps, at 10 km under NSCC, to see whether the outcome changed for larger messages. It didn't—it got cleaner. Across all 152 combinations of message size and buffer point tested, retransmission decreased monotonically as message size increased, with zero exceptions. It also plateaus: most of the improvement from a larger message happens early, and the curve flattens out by roughly 300 MB to 500 MB. Most importantly, at 1.5× BDP, retransmission was 0% for every single message size in the range.

The following tables show both metrics across the full sweep. For both tables, the rows represent message size, from 500 MB down to 50 MB. The columns represent the switch buffer size, in × BDP (BDP = 10 MB at 10 km).

Table 2: Retransmission Rate (%)[b]

Message size	0.1×	0.2×	0.4×	0.6×	0.8×	1.0×	1.25×	1.5×
500MB	23.4%	19.1%	12.8%	6.9%	4.4%	1.7%	0.2%	0%
475MB	23.5%	19.3%	12.9%	7.1%	4.6%	1.8%	0.3%	0%
450MB	23.7%	19.4%	13.1%	7.2%	4.8%	1.9%	0.3%	0%
425MB	23.9%	19.6%	13.3%	7.3%	5.1%	2%	0.3%	0%
400MB	24.2%	19.8%	13.5%	7.4%	5.4%	2.1%	0.3%	0%
375MB	24.6%	20%	13.7%	7.6%	5.7%	2.2%	0.3%	0%
350MB	24.9%	20.4%	14%	7.9%	6%	2.4%	0.4%	0%
325MB	25.1%	20.7%	14.4%	8.3%	6.4%	2.6%	0.4%	0%
300MB	25.6%	21.2%	14.7%	8.8%	6.9%	2.8%	0.4%	0%
275MB	26.1%	21.5%	15.2%	9.4%	7.4%	3.1%	0.5%	0%
250MB	26.9%	22.2%	15.8%	10.1%	8.1%	3.4%	0.5%	0%
225MB	27.7%	23%	16.4%	11.1%	9%	3.8%	0.6%	0%
200MB	28.4%	23.9%	17.3%	12.2%	10%	4.3%	0.7%	0%
175MB	29.5%	25.1%	18.5%	13.6%	11.3%	4.9%	0.8%	0%
150MB	31.2%	26.7%	20.1%	15.5%	13%	5.7%	0.9%	0%
125MB	33.6%	29.2%	22.3%	18.1%	15.5%	6.8%	1.1%	0%
100MB	36.9%	32.8%	25.9%	21.9%	19.4%	8.6%	1.4%	0%
75MB	42.6%	37.6%	32.5%	28.4%	25.3%	11.4%	1.9%	0%
50MB	55.1%	50.7%	47%	37.7%	32.4%	16.9%	2.9%	0%

b. Based on Broadcom internal data.

Table 3: FCT Overhead vs. Ideal (%)[c]

Message size	0.1×	0.2×	0.4×	0.6×	0.8×	1.0×	1.25×	1.5×
500MB	20.6%	16.7%	9.7%	6.1%	8.1%	4.8%	5.9%	3.1%
475MB	21%	16.2%	9.8%	6.5%	8.3%	4.8%	6.2%	3.1%
450MB	20.6%	15.8%	10.4%	6%	8.9%	5%	6.4%	3.1%
425MB	19.8%	15.3%	10%	6.5%	9.1%	5%	6.5%	3.1%
400MB	20.7%	15.2%	10.4%	6.6%	9.5%	5.1%	6.8%	3.1%
375MB	21.8%	15.5%	10.4%	7%	10.1%	5.2%	6.9%	3%
350MB	22.4%	15.7%	10.4%	7.7%	10.3%	5%	7%	2.9%
325MB	22.8%	15.6%	10.7%	7.4%	10.6%	5%	7.3%	2.9%
300MB	22.2%	16.3%	10.8%	7.8%	11.1%	5%	7.6%	2.8%
275MB	23%	17.4%	11.4%	8.6%	11.6%	5%	7.9%	2.9%
250MB	22.9%	17.6%	11.5%	9.2%	11.7%	4.9%	8.2%	3.1%
225MB	24.2%	18%	12.3%	10.4%	11.9%	5%	8.3%	3.3%
200MB	24.4%	19.4%	14.1%	10.1%	14%	7.6%	8.5%	3.4%
175MB	26.8%	20.8%	14.2%	11.2%	12.8%	5.7%	8.5%	3.6%
150MB	24.9%	20.8%	18.1%	12.9%	12.7%	6.3%	8.8%	4%
125MB	25.4%	23.2%	16.5%	11.8%	15.2%	7.1%	8.7%	4.5%
100MB	29.8%	23.6%	18.5%	14.4%	16%	8.3%	9.3%	5.2%
75MB	28.6%	21.2%	19.7%	12.3%	14.8%	10.4%	8.6%	6.2%
50MB	31.5%	21.6%	19.9%	16.2%	13.6%	9.4%	8.9%	7.7%

c. Based on Broadcom internal data.

BELOW 1.5× BDP, RETRANSMISSION CLIMBS FIRST, THEN COMPLETION TIME

In Table 3, note that the 1.5× BDP column stays low and flat across every message size—the same threshold, holding again.

One pattern is worth flagging: unlike the retransmission table, which decreases monotonically as buffer size grows, the FCT-overhead table is noisier in the middle range (0.6× to 1.25× BDP). Completion time depends on the retransmit timer and scheduling interactions that don't move in perfect lockstep with the retransmission rate. The overall trend is unambiguous: overhead is high in the shallow buffer zone and negligible at 1.5× BDP, but the exact ordering of intermediate points should be read as noise around that trend, not as a separate signal.

This section provides three powerful takeaways:

Near-Ideal FCT

A buffer ≥1.5× BDP lands within a few percent of ideal completion time at every distance tested.

Retransmission First

Below threshold, retransmission climbs before completion time does, and then they both grow sharply together.

CC Agnostic

The threshold holds under NSCC and RCCC alike. It's a buffer-sizing property, not a protocol-tuning trick.

Engineering the System, Not Just a Switch

A buffer-sizing threshold is only useful if the rest of the system can act on it. Getting from *we know the right buffer depth* to *the network behaves that way in production* requires orchestration, oversubscription handling, and utilization working together end to end: sender, interconnect, and receiver.

Sender: Thor NIC

- RCCC+ rate modulation tuned for long round-trip times.
- Speculative transmission that hides handshake latency rather than waiting it out.
- Hardware tracking of in-flight packets, enabling wire-speed retransmission when it's needed.

Interconnect: Hierarchical Switch

- Shallow or hierarchical buffering, chosen per deployment rather than fixed to one answer.
- HyperPort consolidation, improving hashing and ECMP entropy for elephant flows.
- Bursts trimmed and re-injected rather than handled with PFC pause frames.

Receiver: Unified Stack

- Telemetry-driven modulation (INT/CSIG) responding to real-time queue depth.
- Integration with customer orchestration layers.
- A closed loop back to sender-side rate control.

Conclusion: A Foundation for the Path to AGI

PROVISION DEEP BUFFERING AT 10 KM AND BEYOND, AND THE SYSTEM HOLDS

The data is unambiguous: once distance crosses 10 km, deep buffers aren't optional, and this is true regardless of congestion-control protocol or message size, with zero exceptions in the simulation. That threshold gives network architects a simple, reliable design rule as multi-site AI clusters move toward gigawatt-scale deployments: provision deep buffering at 10 km and beyond, and the system holds.

For further information, read the Broadcom article: What Is Scale-Across Networking for AI Clusters, and the Arista article: The Many Facets of AI Fabrics.

BROADCOM®

For more information, visit our website at: www.broadcom.com

Copyright © 2026 Broadcom. All Rights Reserved. The term "Broadcom" refers to Broadcom Inc. and/or its subsidiaries.
All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.
This white paper is for informational purposes only and does not constitute a commitment, promise, or obligation to deliver any product, feature, or functionality. Performance figures reflect simulation results under the specific conditions described and may vary in other environments.
CSG-ScaleAI-WP100 October 9, 2026