Bandwidth Works in Practice, not in Theory

Sahas Munamala, Andy Helland, Varun Datta, Dean Gladish, Paul Borrill, DÆDÆLUS Research

June 8, 2025

Ethernet, for the last 5 decades, has operated under the assumption that reliability begins with TCP. This has allowed network hardware to design failure modes of packet transmission into normal operation and sell it as bandwidth. There are more dimensions of throughput than raw link capacity, and more to reliability than patching over losses with idempotent APIs, nonce tokens, and retry logic. There is a simple fact: once the epistemic knowledge of an event is lost in the network, it is unrecoverable by either sender or receiver. No amount of timeout-and-retry or fail-fast design principles can recover the exactly-once event after it is lost. Distributed systems rely on exactly-once for correctness, but commodity hardware built for modern Ethernet consistently violate token integrity. Only a global, open, free standard can truly bridge the gap between Ethernet and proprietary reliable networks like Infiniband and Fibre Channel.

Introduction

The computer networking industry has been marking its progress by the theoretical maximum bandwidth of links. The exponential increase of bits that fit on a 3 m copper cable make a latency argument by claiming the frame will arrive earlier because it transmits faster. On point-to-point connections, this is true, based on the current definitions of bandwidth and latency. However, there is more to bandwidth than # of bits/second that makes it useful, and there's more nuance to latency that will stall distributed applications no matter how fast the hardware gets.

Round-trip interactions are a fundamental unit of computation, particularly in distributed systems, networked computing, and interactive AI workflows. However, conventional networking thinking has placed this as a feature of Layer 4, as an optional transport layer, and not a fundamental method of networked communication. The result of this is TCP, where a Layer 4 protocol between endpoints negotiates the bidirectional transmission of variable length messages on an unreliable network.

This architecture has led to a widespread fallacy: that faster links mean faster systems. In practice, modern distributed applications are not limited by raw link capacity but by round trip latency, queuing behavior, and contention across multiple hops. A 400 Gbps link does not eliminate the propagation delay, backpressure effects, or failures that cascade through switches, NIC buffers, and software stacks. As a result, the marginal gains in link speed are often masked by tail-latency outliers, especially in systems requiring synchronized state or sequential coordination. The assumption that we can “outrun” these delays with bandwidth ignores their non-linearity and compounding effect in real systems.

Hidden cost of Bandwidth-First

Raw, one-way Bandwidth metrics alone fail to account for the crucial aspect of round-trip reliability – the guaranteed and verifiable transfer of packets between nodes. Such guarantees require explicit handshakes by the hardware to properly transfer ownership and responsibility of each packet with the lowest possible latency. Without these mechanisms, software interfaces cannot trust intermediate nodes to handle their tokens responsibly.

In contrast, bandwidth-maximizing designs focus primarily on pushing bit streams at peak throughput. Their impressive bandwidth benchmarks are typically achieved by sending large, uninterrupted byte sequences that minimize overhead. These systems are engineered to drop packets during congestion, prioritizing throughput numbers over the integrity or reliability of tokens.

SENDER
INFINITE BANDWIDTH PIPE
RECEIVER

Figure 1: A infinite bandwidth pipe with packet loss can still limit throughput of TCP flows.

$$\mathrm{BW} = \frac{\mathrm{MSS}}{\mathrm{RTT}} \cdot \frac{C}{\sqrt{p}}$$

MSS: max segment size
RTT: round-trip time
C: constant (1.22)
p: packet loss probability

Consider a lossy link with infinite bandwidth, as illustrated in Figure 1. In this hypothetical, the bottleneck is not raw capacity but the twin constraints of latency and packet loss. Since TCP relies on acknowledgments to regulate its sending rate, the time it takes to complete a round trip –RTT – becomes a limiting factor on throughput. As shown by the Mathis equation[1], throughput degrades proportionally to the inverse of RTT and the square root of the loss probability. In such regimes, increasing bandwidth alone does not improve performance; if anything, the absence of reliable round-trip feedback renders the network incapable of sustaining high-throughput flows. Even in a system of perfect raw transmission capacity, epistemic uncertainty introduced by loss and latency can strangle performance. Round trips are essential to any communication that requires certainty, ordering, or acknowledgment.

In practical networks where congestion is a real and dynamic force, TCP flows must adapt their behavior to avoid collapse. This adaptation is governed by Additive Increase/Multiplicative Decrease (AIMD) – a deceptively simple algorithm that allows each sender to probe the available capacity of the network while reacting swiftly to congestion signals. Each flow increases its sending window linearly over time, but upon detecting loss (interpreted as a sign of congestion), it slashes the rate multiplicatively. This feedback loop produces a sawtooth pattern in throughput, enabling multiple flows to converge to a fair, stable sharing of the underlying path. Crucially, AIMD is fully decentralized and stateless beyond the endpoints. Yet, this

[1] Mathis, M., Semke, J., Mahdavi, J., and Ott, T. The macroscopic behavior of the tcp congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review* 27, 3 (July 1997), 67–82

elegant self-regulation only functions when loss reflects congestion, and when RTT remains a trustworthy signal of delay. In networks where loss is stochastic or induced by buffer mismanagement, AIMD underperforms or misbehaves [2] – but in its ideal regime, it is a marvel of distributed equilibrium: each sender, optimizing selfishly, contributes to global stability.

Packet loss and network congestion represent more than inefficiency, they threaten the epistemic state of distributed systems. When a packet is dropped due to congestion, the information it carried vanishes completely, erasing knowledge of the event it represented. This loss isn't just temporary. It's fundamental and irreversible. Without this information, no node or application can know if the packet is coming late, or not at all. Exactly-once semantics rely entirely upon preserving and transferring this epistemic state across nodes. Failures in handling epistemic state leads to inconsistency and grey failures[3]. Thus, the hidden peril of the bandwidth-first approach emerges clearly: by optimizing purely for throughput at the expense of reliable delivery, one risks catastrophic losses in epistemic certainty, fundamentally undermining the correctness and reliability of distributed computation.

Claude Shannon

Shannon showed us that redundancy is necessary to overcome noise, but it doesn't have to come in the form of retransmissions. It can emerge from structure, timing, flow control, and error-detecting codes. In his landmark 1948 paper[4], Shannon formalized the limits of communication under uncertainty, showing that any noisy channel could achieve reliable transmission — not by eliminating noise, but by understanding and quantifying it, then embedding just enough redundancy to overcome it. This gave rise to the idea of channel capacity: the maximum rate at which information can be sent over a noisy medium with arbitrarily low error.

However, Shannon's theory is asymptotic. It assumes infinite message lengths and probabilistic decoding, which don't cleanly map onto real-time systems with finite payloads, strict latency bounds, and diverse failure modes. In practice, the design of reliable communication must balance redundancy with delay, error handling with timing, and correctness with structure.

Crucially, redundancy does not imply retransmission. A reliable system can suppress entropy not just by encoding messages, but by constraining the channel. Deterministic protocols — like InfiniBand and Fibre Channel — ensure losslessness not through statistical error correction, but through strict flow control, bounded buffering, and

[2] Gettys, J., and Nichols, K. Bufferbloat: dark buffers in the internet. *Commun. ACM 55*, 1 (Jan. 2012), 57–65

[3] Grey failures represent a class of failure where events are partially known or suspected, but never fully provable

[4] Shannon, C. E. A mathematical theory of communication, 1948

physical acknowledgment paths. By guaranteeing that every bit is either delivered or backpressured, they narrow the range of possible outcomes. This reduces uncertainty without increasing retries — a form of reliability that aligns with Shannon's vision but adapts it to interactive, bounded-latency environments.

True reliability, then, need not come at the cost of throughput. According to information theory, maximum capacity is achieved when redundancy is sufficient but minimal — just enough to defeat uncertainty. Lossless fabrics accomplish this through design, not brute force. They do not fight entropy after the fact; they prevent it from entering the system at all. The result is not just efficiency — it's epistemic fidelity: a network where knowledge, once sent, is preserved without compromise.

Fibre Channel

Fibre Channel was designed as a highly reliable, lossless transmission protocol, with robust flow control and acknowledgment mechanisms fundamentally built in. It uses a credit-based system where each receiver controls the transmission rate through buffer credits, so the sender can only transmit frames when the receiver is ready, eliminating overruns and ensuring lossless delivery. Acknowledgments, both implicit and explicit, confirm receipt of frames, while any lost or corrupted frames can be detected via CRC checks and retransmitted using control frames. These mechanisms work end-to-end and at every switch, guaranteeing in-order, reliable delivery across the entire fabric. As a result, Fibre Channel became the backbone of enterprise storage area networks (SAN), bringing mainframe-class reliability and flow control to open systems and enabling storage area networks to scale without the risk of dropped or unordered data.

Fibre Channel hardware was much more expensive than Ethernet equivalents, sometimes by an order of magnitude. Even mid-tier SAN switches could cost tens of thousands of dollars. Fiber Channel also required dedicated expertise. Configuration, zoning, and troubleshooting were very different from the familiar world of Ethernet networking, making skilled staff scarce and expensive.

As Ethernet bandwidth increased from 1/10/25/40/100Gbps and beyond, its price dropped and its reliability improved. Features like iSCSI, FCoE (Fibre Channel over Ethernet), and later, NVMe-over-Fabrics let storage ride over ordinary Ethernet with performance and reliability approaching Fibre Channel for most workloads. Ethernet was already everywhere: data, management, and now storage could be unified onto one network—saving hardware, cabling, and operational overhead.

InfiniBand

InfiniBand[5], by fixing the message structure and introducing deterministic flow control, turns a noisy, uncertain channel into a nearly noiseless, deterministic pipeline, analogous to a physical circuit.

Fixed-size packets enable efficient buffer management, deterministic flow control, and cut-through switching— critical for maintaining lossless transmission. From a Shannon theory perspective, fixed packet sizes simplify the encoding and decoding process by reducing entropy per symbol and minimizing variance in transmission time, which stabilizes throughput near channel capacity.

InfiniBand offered clear technical advantages, including guaranteed lossless delivery, extremely low latency through cut-through switching, and efficient memory transfers via RDMA. However, it failed to achieve mainstream adoption beyond high-performance computing and certain enterprise deployments. This outcome resulted from a combination of economic, architectural, and ecosystem realities. InfiniBand required specialized hardware, tight coupling, and strict credit-based flow control, making it difficult to scale and integrate in diverse environments. While these features served scientific and tightly synchronized workloads well, they were overengineered and prohibitively expensive for the broader datacenter and cloud markets.

The acquisition of Mellanox[6] by NVIDIA in 2020 was a significant move that sent ripples through Silicon Valley, though it did not cause a dramatic upheaval in the traditional sense. It was less of a shockwave and more of a strategic signal—one that made clear NVIDIA's ambitions to move beyond GPUs and into the heart of data center infrastructure.

AI training benefits enormously from lossless networking, but it's not because the algorithms require it for correctness. Rather, it's about efficiency, scale, and determinism in distributed computation. Reliable networking seamlessly, and near-losslessly, extends GPU memory space across multiple GPUs, allowing compute pods to share memory like a single fast memory pool.

Ethernet Workarounds

Ultimately, Fibre Channel and InfiniBand are specialized solutions in a world that prioritizes openness, cost efficiency, and broad adoption. Rather than merge with a proprietary technology, the Ethernet ecosystem chose to adapt and absorb its most useful features. The result was not technical defeat but strategic obsolescence.

Ethernet embraces simplicity, flexibility, and continuous evolution.

[5] InfiniBand guarantees lossless transmission even under congestion by using credit-based flow control and hardware backpressure. Instead of dropping packets, it prevents senders from overrunning buffers, ensuring reliable delivery without retries.

[6] Mellanox was the steward and primary driver of InfiniBand technology. It played a central role in both the development and commercialization of the InfiniBand standard, acting as the lead implementer and key evangelist within the HPC and low-latency networking community.

Its early limitations were addressed through layered improvements such as Data Center Bridging, Priority Flow Control, and RDMA over Converged Ethernet. These additions made Ethernet "good enough" for many high-throughput, low-latency applications without abandoning compatibility with legacy systems. Each of these enhancements represents a patch on top of a fundamentally best-effort, lossy protocol stack. While they reduce packet drops in controlled environments, their guarantees are neither universal nor absolute.

At their core, these Ethernet extensions still operate atop the IP protocol, whose very design assumes and accepts the possibility of loss, reordering, and duplication. Reliability is punted up the stack to transport protocols like TCP, or to the application layer, where error detection, retransmission, and exactly-once semantics are laboriously reconstructed. As a result, the burden of handling ambiguity, uncertainty, and failure is pushed to the endpoints and the software, rather than being enforced in the network fabric itself.

This architecture cannot escape its origins. Any guarantee of reliability or determinism is inherently probabilistic and contingent on network conditions, traffic patterns, and careful configuration. In the presence of congestion, misconfiguration, or adversarial workloads, these workarounds can break down, leading to unpredictable loss, jitter, and transient failures – exactly the failure modes that storage and distributed systems sought to avoid by adopting Fibre Channel or InfiniBand in the first place.

Thus, the quest to retrofit lossless, reliable behavior onto a protocol stack designed for "good enough" connectivity results in a system that is complex, fragile, and never truly deterministic. Until the foundational assumption of best-effort delivery is replaced with provable, in-fabric reliability, Ethernet's workarounds will always be at risk of failure, and the promise of robust, exactly-once distributed computation will remain out of reach.

Are Reliable Networks still Niche Today?

Applications require predictable and deterministic behavior from the networks they rely upon. Every layer – from API down to the physical transmission of bits on the wire and back – must preserve the semantics of exactly-once requests.

Today, applications are forced to tolerate all forms of network unreliability in their communication because lossless, deterministic networks are expensive, proprietary, and specialized. However, modern applications from distributed microservices, replicated databases, robotics, and automation, are fundamentally dependent on predictable, exactly-once semantics. These applications crave reliability because it vastly simplifies their internal logic, improves consistency guarantees, and significantly reduces the burden of handling retries, duplicates, and partial failures.

With easy and open access to a fully verified, open-source standard for reliable communication, applications can trust their infrastructure implicitly, dramatically simplifying their design and enhancing operational robustness. The availability of such open, reliable standards democratizes technology previously reserved for niche, high-budget environments, enabling widespread adoption and fundamentally reshaping expectations around distributed system reliability.

Local Only Control

- Software Defined Networking provides applications with network-awareness by dynamically through centralized controllers, however a centralized control system suffers from the same exactly-once semantic issues as any other application
- It is impossible to create a gods-eye-view entirely reactive system that avoids network congestion entirely. Instead, state of the art Software Defined Networking will always be one-step behind congestion
- Instead, fully verifiable algorithms like spanning trees, failure routing, and healing must be done with local-only information

New Metrics for Networks

- New Local-First metrics must be created to measure network performance. Reliance on one-way bandwidth performance does not characterize network reliability, resilience, or performance.
- Interaction Latency – time (ns) for a round trip acknowledgement (hop-by-hop)
- Round Trip Bandwidth – # payload bits per second of acknowledged data bits (ignores headers)
- Resilience Metric for Constrained-Valency Networks via Graph Laplacian. Classically Edge Connectivity (minimum number of edges whose removal disconnects G), and Vertex Connectivity (minimum number of vertices whose removal disconnects G) capture global connectivity

References

[1] Gettys, J., and Nichols, K. Bufferbloat: dark buffers in the internet. *Commun. ACM 55*, 1 (Jan. 2012), 57–65.

[2] Mathis, M., Semke, J., Mahdavi, J., and Ott, T. The macroscopic behavior of the tcp congestion avoidance algorithm. *ACM SIGCOMM Computer Communication Review 27*, 3 (July 1997), 67–82.

[3] Shannon, C. E. A mathematical theory of communication, 1948.