

CO suffices for the Dijkstra ring-token detector because any basic message can cross *at most one* wave; the colouring rule then guarantees safety.

3. Many rendezvous-based algorithms work under CO, avoiding blocking latency.
4. FIFO alone is *insufficient* for algorithms that assume “no messages in flight.”
5. Implementation cost rises sharply: RSC \rightarrow latency; CO \rightarrow meta-data; FIFO \rightarrow buffer space.

1.2 Link Fabrics: An Objective Comparison

This document provides a technical overview of five prominent link fabrics: NVLink, UALink, Scale-Up Ethernet, InfiniBand, and conventional Ethernet. We evaluate their architectural characteristics, header overheads, and implications for block-size efficiency, with particular focus on suitability for modern workloads such as AI, HPC, and disaggregated systems. We present side-by-side comparisons of header sizes relative to data payloads and reflect on architectural biases that favor or hinder scalability, latency, and composability.

1.2.1 Introduction

As computation and memory disaggregation evolve, the role of the interconnect becomes central. Whether linking GPUs in an AI training pod, scaling up a symmetric multiprocessor system, or tying together memory and compute pools across racks, the *link fabric* is the substrate on which system performance is built.

Each fabric comes with assumptions about packet size, latency, error recovery, congestion handling, and topology. In this review, we provide a comparative analysis of five contenders:

- **NVLink** (NVIDIA): High-bandwidth, low-latency GPU interconnect
- **UALink** (AMD, Broadcom et al.): Emerging open alternative to NVLink
- **Scale-Up Ethernet**: Evolving conventional Ethernet for tightly coupled systems
- **InfiniBand**: HPC-focused with credit-based flow control and low-latency verbs
- **Conventional Ethernet**: Ubiquitous best-effort packet transport

1.2.2 Architectural Overview

1.2.3 Topology and Purpose

- **NVLink**: Point-to-point or mesh GPU topologies with explicit scheduling and hardware-managed coherence domains.
- **UALink**: Targeted as a broader standard across vendors; switch-based; supports memory pooling and accelerator interconnect.

- **Scale-Up Ethernet:** Designed to bring reliability and ordered delivery to Ethernet via reduced headers, low-latency slicing, and potential for transaction-level acknowledgment.
- **InfiniBand:** Mature switch-based architecture, deeply integrated into RDMA stacks and MPI; emphasis on zero-copy and reliable transport.
- **Ethernet:** Best-effort delivery; scales via oversubscription and buffering; header-heavy; assumes software-managed retry.

1.2.4 Header Overhead vs. Block Size

Link fabrics differ significantly in header-to-payload ratio, especially at small block sizes. Header overhead penalizes small messages in conventional Ethernet, motivating larger minimum block sizes to maintain efficiency. For AI and tightly coupled compute, where atomicity and latency matter, small block sizes with low overhead are preferable.

1.2.5 Header Size vs Block Size Table

Fabric	Header	64B	128B	256B	512B	1024B
NVLink v3	16	25.0%	12.5%	6.3%	3.1%	1.6%
UALink (proj.)	20	31.3%	15.6%	7.8%	3.9%	2.0%
Scale-Up ETH	8–16	12.5%	6.3%	3.1%	1.6%	0.8%
InfiniBand HDR	64	100%	50.0%	25.0%	12.5%	6.3%
Ethernet + IP + TCP	76–92	143.8%	57.8%	28.9%	14.5%	7.1%

Does not Include Atomic Ethernet

We wanted to review and compare these systems before introducing OAE. Imagine what this table looks like when we add OAE’s 4-byte header, with fixed size: 64B, 256B, 1024B and 4096B transfers.

1.2.6 Latency and Atomicity Considerations

Atomic operations (e.g., tensor updates, semaphore-based locking) are increasingly being performed across links. The cost of round-trips, retries, or failed speculative execution due to packet drops grows non-linearly with header size and tail latency variance.

- **NVLink**: High atomicity; limited to GPU domain.
- **UALink**: Claims to enable coherent memory semantics across nodes.
- **InfiniBand**: Explicit verbs for atomic ops, requires RDMA semantics.
- **Ethernet**: Lacks atomic primitives; must emulate via protocols.
- **Scale-Up Ethernet**: Explicit focus on atomic packet slices, with transaction-layer feedback.

1.2.7 Bias Toward Large Packets: A Critical View

Switch-centric fabrics such as Ethernet and InfiniBand often exhibit biases toward large block sizes, due to:

1. **Header amortization:** Larger blocks reduce relative overhead.
2. **Switch buffer economics:** Designed for long flows, not short atomic ops.
3. **Congestion avoidance:** Larger packets allow queue shaping, but penalize small-ops latency.

This introduces architectural bias that disfavors emerging patterns such as sparse updates, fine-grain load/store traffic between heterogeneous xPUs, or distributed execution of transactional graphs.

1.2.8 Conclusion

While Ethernet and InfiniBand continue to evolve, they remain biased toward packetization strategies that penalize small atomic units of work. NVLink and UALink challenge this by focusing on coherence and bandwidth at short distances, but they remain vendor-specific or in flux.

Scale-Up Ethernet presents a new opportunity: to build an atomic-capable, low-latency, congestion-aware transport that preserves Ethernet compatibility while shedding unnecessary biases—especially those that require applications to batch operations just to amortize protocol overhead.

Future fabrics should not merely transmit data, but should transmit *meaningful, recoverable state*—one slice at a time.

1.3 Ultra Ethernet

1.3.1 Executive overview

Ultra Ethernet Consortium (UEC) is producing an **open, Ethernet-based full-stack specification** targeted at AI and HPC fabrics with one million endpoints, terabit links and sub-microsecond tail latency. Its clean-slate *Ultra Ethernet Transport (UET)* replaces legacy RoCE, adds packet-spray multipath, flexible ordering, congestion-adaptive spraying and hardware collectives.

Compared with InfiniBand, UET keeps the broad Ethernet ecosystem while approaching supercomputer-interconnect performance.

1.3.2 Scope and objectives

- **Scale:** up to 1e6 endpoints and 800;1600 Gbps links.
- **Latency target:** keep 99.9 % of collectives within one-digit μ s

- **Profiles:** *AI Base*, *AI Full*, and *HPC*; same wire protocol, differing verb sets.
- **Layer coverage:** Physical, Link, Transport, Software API, Security, Telemetry/Management.

1.3.3 Architecture at a glance

Transport layer – UET

- **Ephemeral connections:** no handshake; state cached and reclaimed per transaction (memory-efficient at scale).
- **Packet spraying** across all equal-cost paths with per-packet sequence numbers; ordering enforced on message completion only.
- **Reliability** uses selective retry plus optional *packet trimming* (header delivered, payload discarded) to signal incipient congestion without head-of-line blocking.

Congestion control

Two complementary schemes:

- (1) **Sender-adaptive AI/MD** window (~RTT-speed reaction, ECN-driven).
- (2) **Credit-grant** receiver pacing for extreme incast.

Link layer

- **Link-Layer Retry (LLR)** negotiated via extended LLDP; hop-by-hop NACK keeps BER-induced loss from bubbling up.
- **Two traffic classes** prevent deadlock and prioritise control traffic even on PFC-free fabrics.

Software API

Based on libfabric 2.0 with new verbs: *Deferrable Send*, optimistic rendezvous, expanded atomics and tagged collectives.

Security

Group-keyed AES-GCM; leverages IPSec/PSP primitives yet preserves ephemeral-connection model—important for accelerator offload.

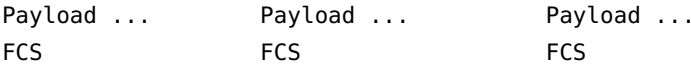
In-network collectives (INC)

Lightweight header extensions allow switches to execute *reduce*, *broadcast* and *all-reduce* without CPU round-trips, standardising what was previously vendor-proprietary.

1.3.4 Key innovations vs. legacy RDMA

1.3.5 Performance and scalability

UEC's internal simulations on Clos-4 fabrics show >95 % link utilisation and <5 µs 99.9-percentile all-reduce latency on 4096-GPU clusters



Ethernet II places the EtherType immediately after the source MAC and is the framing used by practically all modern IP traffic.¹ With 802.3 + LLC, the receiver must parse the LLC header (DSAP / SSAP) to learn which upper layer service is carried. SNAP keeps the 802 structure while still transporting traditional EtherType values.

¹ See IEEE Std 802.3-2022, Section 3.1.

1.4.4 Why LLC Faded on Ethernet

1. **Simplicity and cost.** Early NICs already spoke Ethernet II, so adding LLC parsing logic gave little benefit.
2. **IP dominance.** Most traffic needed only EtherType 0x0800 (IPv4) or 0x86DD (IPv6), so DSAP / SSAP were redundant.
3. **Redundant reliability.** LLC Type 2 and 3 capabilities overlapped with TCP end to end guarantees.
4. **VLAN tagging.** 802.1Q inserts its own 4 byte shim but preserves EtherType demultiplexing.

As a result, modern NICs understand LLC/SNAP for legacy frames (for example STP, LLDP), yet more than 99 percent of everyday traffic uses Ethernet II framing.

1.4.5 LLC Versus Other Ethernet Related Layers

Aspect	LLC (802.2)	Ethernet II	MAC Control
Purpose	Uniform service, optional reliability	Minimal frame wrapper	Flow control, security
Media scope	Any IEEE 802 medium	Ethernet only	Ethernet only
Header size	3 B (8 B with SNAP)	none beyond MAC + EtherType	Control frames are separate 64 B PDUs
Error handling	Optional ACK / REJ at L2	None	Pause frames stop the transmitter; no ARQ
Typical use today	STP, LLDP, some industrial stacks	IP, ARP, VLANs, nearly all data traffic	Datacenter congestion control, MACsec

Table 1.4.5: Comparative position of LLC and other Ethernet related layers.

1.4.6 Take Aways for New Protocol Design

- If you need to multiplex a new L3 protocol, registering an EtherType or using OUI + SNAP is simpler than reviving DSAP / SSAP values.
- Link layer reliability costs latency. Modern reversible or causal ordering schemes are better placed above the MAC, much as RoCEv2 rides over UDP/IP.
- For designs that must traverse Wi Fi or other IEEE 802 media, SNAP framing keeps you inside the standard while preserving familiar EtherType semantics.

1.5 Infiniband vs Ethernet

From `./AE-Specifications-ETH/Infiniband.tex`

1.5.1 High-Level Overview

At a high level, **Ethernet** and **InfiniBand** both provide packet-switched networking, yet they originate from very different design philosophies and ecosystems. The widespread perception that **InfiniBand is more reliable** arises from the architectural and operational differences examined below.

1.5.2 Origins and Design Philosophy

- **Ethernet** evolved as a best-effort LAN. Reliability, ordering, and latency guarantees were relegated to higher layers (TCP/IP), Quality of Service, or RDMA over Converged Ethernet (RoCE).
- **InfiniBand** was conceived for high-performance computing (HPC) and modern data-center fabrics, prioritising low latency, high throughput, lossless delivery, and *built-in* reliability.

1.5.3 Transport and Reliability Model

- **Ethernet (traditional)** is best-effort; packets may be dropped, duplicated, or reordered. Higher layers must restore reliability.
- **InfiniBand** embeds a reliable transport protocol in hardware, handling acknowledgments, retransmissions, and flow control with minimal host-CPU involvement.

1.5.4 Congestion and Flow Control

- **Ethernet:** early 802.3 had none. Data-Center Bridging (DCB) and Priority Flow Control (PFC) improve matters but add complexity and are not always enabled end-to-end.
- **InfiniBand:** credit-based flow control and end-to-end congestion management prevent buffer overruns without dropping traffic.

1.5.5 Packet-Loss Behaviour

- **Ethernet:** under load or misconfiguration, packets drop and TCP must recover, causing latency spikes.
- **InfiniBand:** packets are rarely dropped; congestion produces back-pressure instead.

1.5.6 Latency and Jitter

- **Ethernet:** latency and jitter depend on traffic, buffering, and TCP recovery.
- **InfiniBand:** microsecond-scale latency and very low jitter via lightweight stack, hardware offload, and zero-copy RDMA.

1.5.7 RDMA Support

- **Ethernet:** RoCE provides RDMA but demands a tuned, lossless fabric (PFC, ECN, buffer sizing), often fragile and vendor-specific.
- **InfiniBand:** native RDMA with hardware reliability and no special tuning.

1.5.8 CPU Overhead

- **Ethernet:** full TCP/IP stack consumes CPU unless offloaded by SmartNICs.
- **InfiniBand:** host-channel adapters offload packetisation, ordering, and reliability, yielding lower CPU utilisation.

1.5.9 Ecosystem and Deployment

- **Ethernet:** ubiquitous, inexpensive, multi-vendor, and scaling to 800 Gb/s.
- **InfiniBand:** dominant in latency-sensitive HPC/AI but niche, costlier, and largely single-vendor (NVIDIA/Mellanox).

1.5.10 Summary Table

Feature	Ethernet	InfiniBand
Reliability	Best-effort (TCP)	Hardware-enforced
Latency	Milli-second typical	Micro-second
Jitter	High	Very low
Congestion control	Optional (DCB/PFC)	Built-in
Packet loss	Possible	Avoided by design
RDMA	RoCE (complex)	Native
CPU overhead	High (software stack)	Low (offload)
Primary use	General networking	HPC / AI clusters

Table 1.4: High-level comparison of Ethernet and InfiniBand.

1.5.11 RoCE: RDMA over Converged Ethernet

1.5.12 Variants

- **RoCE v1:** Layer 2 only, not routable.
- **RoCE v2:** UDP/Layer 3, IP-routable.

1.5.13 Why Reliable Deployment is Hard

1. **Lossless Fabric Requirement:** depends on PFC; mis-tunes cause deadlocks and head-of-line blocking.
2. **UDP Transport:** inherits best-effort IP semantics unless the fabric is tightly managed.
3. **Ecosystem Coordination:** NICs, drivers, libraries (`libibverbs`), and applications must align or fall back to TCP.

1.5.14 Why Ethernet Remains Dominant

- **Cost & Ubiquity:** every datacenter already runs Ethernet; hardware is commoditized.
- **Interoperability:** multi-vendor openness avoids lock-in.
- **Performance Road-map:** speeds have risen from 10 Gb/s to 800 Gb/s.
- **Software Ecosystem:** the global Internet stack assumes Ethernet/TCP.
- **RoCE as Bridge:** hyperscalers deploy RoCE successfully by exerting strict control over their fabrics.

1.5.15 Conclusion

InfiniBand remains the gold standard for ultra-low-latency, highly reliable HPC workloads. Ethernet is closing the gap through RoCE, SmartNIC offloads, DCB/PFC, and ever-faster links. Its dominance stems from universality and cost, not intrinsic technical superiority.

1.5.16 Design Goals for a Next-Generation Fabric

1. Sub-microsecond latency with deterministic throughput.
2. Hardware-enforced reliability (acknowledgment & retransmission in silicon).
3. *RDMA-first* semantics: zero-copy PUT, GET, and atomic operations.
4. Programmability: P4/eBPF pipelines in NICs and switches.
5. Security by design: cryptographic authN/authZ and fine-grained access control.
6. Clock-agnostic operation: causal or reversible timing models.
7. Composable transports: reliable/unreliable, ordered/unordered as required.
8. Multi-tenant virtual fabrics on shared hardware.

1.5.17 Key Building Blocks

- **SmartNICs:** onboard CPUs or FPGAs for protocol state, reversibility, and EPI/ONT registers.
- **Flow-Aware Non-Switch Fabric:** dynamic, congestion-aware path scheduling.
- **Unified Declarative Transport:** intent-based API replacing TCP/IP or InfiniBand verbs.
- **Fabric-Wide Memory Space:** global RDMA address space with capability-based security.

1.5.18 Design Framework

Something Old Knowledge == captured information

Something New RED == Information surprisal (the answer to a yes/no question)

Something Borrowed == Tie-in to Rust model (for RPC - Alice owns but Bob borrows)

Something Blue Semantics (Meaning) – The SmartNIC/IPU understands the context

Something Green OCP Green for Open Syntax – goes over PCIe.

From [./AE-Specifications-ETH/Infiniband.tex](#)

At a high level, Ethernet and InfiniBand serve similar roles as network technologies, but they come from very different design philosophies and ecosystems. The perception (and often the reality) that InfiniBand is more reliable than Ethernet stems from multiple architectural and operational differences:

Origins and Design Philosophy Ethernet evolved from a best-effort, general-purpose LAN protocol for office use. Reliability, ordering, and latency guarantees were added later (e.g., via TCP/IP, QoS, or RDMA over Converged Ethernet—RoCE). InfiniBand was designed from the start for high-performance computing (HPC) and data centers, prioritizing low latency, high throughput, lossless transmission, and reliability at the transport layer.

Transport and Reliability Model Ethernet (traditional) is best-effort. Packets may be dropped, duplicated, or arrive out of order. It's the responsibility of higher layers (e.g., TCP) to ensure reliability. InfiniBand includes a reliable transport protocol in hardware, supporting retransmissions, acknowledgments, and flow control natively, without software stack involvement.

Congestion and Flow Control Ethernet historically lacked built-in congestion control (early 802.3 Ethernet had none). Modern enhancements like Data Center Bridging (DCB) and Priority Flow Control (PFC) try to fix this, but they're complex and not universally deployed. InfiniBand uses credit-based flow control and end-to-end congestion control, ensuring no buffer overruns and graceful behavior under load.

Packet Loss Behavior Ethernet (especially under load or with improper configuration) will drop packets, which TCP must recover from. In large-scale systems, dropped packets can cause latency spikes or retries. InfiniBand virtually never drops packets under normal operation. When congestion happens, packets are backpressured rather than discarded.

Latency and Jitter Ethernet is not inherently low-latency. Latency and jitter vary depending on traffic conditions, switch buffering, and TCP behavior. InfiniBand achieves microsecond-scale latencies with

very low jitter, due to lightweight protocol stack, hardware offloads, and zero-copy RDMA support.

RDMA Support Ethernet supports RDMA via RoCE, but it requires very careful network tuning (lossless fabric, PFC, ECN, etc.), which is brittle and often vendor-specific. InfiniBand natively supports RDMA with hardware reliability, requiring no tuning or lossless Ethernet tricks.

CPU Overhead Ethernet (via TCP/IP stack) incurs significant CPU overhead unless you're using SmartNICs or offload engines. InfiniBand offloads nearly everything—packetization, reliability, ordering—to the HCA (Host Channel Adapter), allowing much lower CPU utilization.

Ecosystem and Deployment Ethernet is everywhere—ubiquitous, cheap, interoperable, and increasingly faster (100/200/400/800 Gbps). InfiniBand is dominant in HPC and AI clusters where latency and throughput are critical, but it's niche, expensive, and vendor-constrained (mainly NVIDIA/Mellanox).

Let's start by exploring RoCE (RDMA over Converged Ethernet), then dig into why Ethernet dominates, despite its technical inferiority in some contexts.

1.5.19 1. What is RoCE?

RoCE stands for RDMA over Converged Ethernet, and it attempts to bring InfiniBand-style performance (especially RDMA) to the Ethernet world. It's driven largely by:

The widespread dominance of Ethernet hardware. The desire to reduce CPU overhead using zero-copy RDMA. Avoiding the proprietary nature and cost of InfiniBand gear. Variants:

RoCE v1: Works at Layer 2. Not routable—same Ethernet broadcast domain.

RoCE v2: Works at Layer 3. IP routable (UDP-based).

1.5.20 2. Why RoCE is Hard to Deploy Reliably

RoCE does not bring InfiniBand-level reliability automatically. Here's why:

- a. Requires Lossless Ethernet Fabric
RoCE depends on PFC (Priority Flow Control) to avoid packet loss, because it assumes a reliable transport. But PFC is fragile and can cause deadlocks, head-of-line blocking, or cascading failures if mis-

