

BMWG
Internet-Draft
Intended status: Informational
Expires: 6 December 2026

F. Calabria
Cisco
C. Pignataro
Blue Fern Consulting
Q. Wu
G. Fioccola
Huawei
S. Reddy
Apple
4 June 2026

Benchmarking Methodology for AI Training Network Fabrics
draft-calabria-bmwg-ai-fabric-training-bench-02

Abstract

This document defines benchmarking terminology, methodologies, and Key Performance Indicators (KPIs) for evaluating Ethernet-based AI training network fabrics.

As large-scale distributed Artificial Intelligence / Machine Learning (AI/ML) training clusters grow to tens of thousands of accelerators (GPUs or generic accelerator processing units (XPU)), the backend network fabric becomes the critical bottleneck determining Job Completion Time (JCT), training throughput, and accelerator utilization.

This document establishes vendor-independent, reproducible test procedures for benchmarking fabric-level performance under realistic AI training workloads, covering Remote Direct Memory Access (RDMA) over Converged Ethernet version 2 (RoCEv2) transport, the Ultra Ethernet Transport (UET) protocol defined by the Ultra Ethernet Consortium (UEC) Specification 1.0 [UEC-1.0], congestion management (Priority Flow Control (PFC), Explicit Congestion Notification (ECN), Data Center Quantized Congestion Notification (DCQCN), Credit-Based Flow Control (CBFC)), load balancing strategies (Equal-Cost Multi-Path (ECMP), Dynamic Load Balancing (DLB), packet spraying), collective communication patterns (AllReduce, AllToAll, AllGather), and scale/soak testing.

The methodology enables direct, reproducible comparison across different switch ASICs, vendor implementations, NIC transport stacks (RoCEv2 vs. UET), and fabric architectures (2-tier Clos, 3-tier Clos, rail-optimized).

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://fcalabri.github.io/bmwg-ai-fabric-training-bench/draft-calabria-bmwg-ai-fabric-training-bench.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-calabria-bmwg-ai-fabric-training-bench/>.

Discussion of this document takes place on the BMWG Working Group mailing list (<mailto:bmwg@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/bmwg/>. Subscribe at <https://www.ietf.org/mailman/listinfo/bmwg/>.

Source for this draft and an issue tracker can be found at <https://github.com/fcalabri/bmwg-ai-fabric-training-bench>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 December 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	4
1.1. Requirements Language	5
1.2. Scope and Applicability	5
1.3. Relationship to Existing BMWG Work	6
2. Terminology	7
2.1. Acronyms	7
3. Test Topology and Architecture	7
3.1. Reference Fabric Topologies	8
3.1.1. Topology A: 2-Tier Clos (Leaf-Spine)	8
3.1.2. Topology B: 3-Tier Clos (Leaf-Spine-Superspine)	8
3.1.3. Topology C: Rail-Optimized	8
3.2. Device Under Test (DUT) Identification	9
3.3. Traffic Generator Requirements	10
3.3.1. Mandatory Functional Capabilities	10
3.3.2. Minimum Measurement Accuracy Requirements	10
3.3.3. Acceptable Implementations	11
4. KPI Framework and Metrics Taxonomy	11
4.1. Primary KPIs	12
4.2. Secondary KPIs	12
4.3. Fabric Health Indicators	13
5. Test Category 1: RDMA Transport Benchmarks	14
5.1. Baseline Throughput	14
5.2. Latency Characterization	14
5.3. Back-to-Back Burst Absorption	15
6. Test Category 1A: UEC Transport Protocol Benchmarks	15
6.1. UET Throughput by Transport Service	16
6.2. UET Latency Characterization	16
6.3. Packet Spray Efficacy Under UET RUD	17
6.4. UET Congestion Control Benchmarks	18
6.5. Link Layer Enhancement Benchmarks	18
6.6. UET Collective Communication Performance	19
6.7. UET PDC Scalability and Connection Setup Rate	20
7. Test Category 2: Congestion Management	20
7.1. ECN Marking Accuracy and Threshold	20
7.2. PFC Behavior Under Incast	21
7.3. DCQCN Convergence Time	21
7.4. PFC Storm and Deadlock Resilience	21
8. Test Category 3: Load Balancing Efficacy	21
8.1. ECMP Entropy and Polarization	21
8.2. Dynamic Load Balancing (Flowlet)	22
8.3. Packet Spraying	22
8.4. Jain Fairness Index Measurement	22
9. Test Category 4: Collective Communication Benchmarks	23
9.1. AllReduce Benchmark	23
9.2. AlltoAll Benchmark	23
9.3. AllGather Benchmark	24

9.4. Collective Communication Library Bus Bandwidth Summary	25
10. Test Category 5: Job Completion Time (JCT) Benchmarks	25
10.1. Synthetic JCT Under Controlled Conditions	25
10.2. MLPerf-Aligned JCT	27
10.3. Multi-Tenant JCT Interference	27
11. Test Category 6: Scale and Convergence	28
11.1. Fabric Scale Limits	28
11.2. Link Failure Convergence	28
11.3. Zero-Impact Failover Measurement	28
12. Test Category 7: Soak and Stability	29
12.1. 24-Hour Sustained Load	29
12.2. Resource Leak Detection	29
13. Reporting Format	29
14. Security Considerations	30
15. IANA Considerations	32
16. References	32
16.1. Normative References	32
16.2. Informative References	33
Appendix A. KPI-to-Test Mapping Summary	34
Appendix B. Indicative Reference Values (Non-Normative)	36
Appendix C. ASIC Feature Categories (Informational)	37
Appendix D. RoCEv2 Test Frame Format	39
Appendix E. UET (Ultra Ethernet Transport) Frame Format	40
E.1. Key Differences from RoCEv2	42
Acknowledgments	43
Authors' Addresses	43

1. Introduction

The rapid growth of distributed AI/ML training workloads has fundamentally changed the performance requirements for data center network fabrics. Unlike traditional data center traffic characterized by diverse flow sizes and protocols, AI training workloads generate highly synchronized, bandwidth-intensive, east-west traffic patterns dominated by collective communication operations (AllReduce, AlltoAll, AllGather). These workloads impose unique demands: lossless transport (via RoCEv2 over RDMA), ultra-low tail latency, near-perfect load balancing across all fabric paths, and the ability to absorb coordinated micro-bursts from thousands of accelerators simultaneously.

Existing BMWG methodologies, while foundational, do not adequately address the characteristics of AI training fabrics. [RFC2544] defines benchmarking for general network interconnect devices but does not account for RDMA transport semantics, collective communication patterns, or the unique congestion dynamics of GPU-to-GPU traffic. [RFC8238] and [RFC8239] establish data center benchmarking terminology and methodology but predate the AI fabric

paradigm and do not address RoCEv2-specific behaviors such as Priority Flow Control (PFC) interactions, DCQCN congestion control convergence [DCQCN-PAPER], or the impact of load balancing strategies on Job Completion Time (JCT). Industry experience deploying RoCEv2 at scale [META-ROCE] further highlights the need for standardized benchmarking methodology.

The Ethernet Virtual Private Network (EVPN) benchmarking methodology [EVPN-BENCH] provides a structural template for service-oriented benchmarking but is scoped to L2VPN services rather than RDMA fabrics.

This document fills the gap by defining a comprehensive benchmarking methodology specifically designed for AI training network fabrics.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Scope and Applicability

This document applies to Ethernet-based AI training backend network fabrics employing RoCEv2 and/or UEC Ultra Ethernet Transport (UET) protocols. The scope includes leaf-spine (2-tier Clos) and leaf-spine-superspine (3-tier Clos) topologies.

InfiniBand fabrics are explicitly *out of scope*, though many KPIs defined herein may be adapted for IB benchmarking by future documents. The DUT is the network fabric itself (the collection of switches and interconnecting links), not individual accelerators or host NICs; host-side configuration is documented in the test report as it materially affects results.

The DUT boundary for all measurements in this document is the NIC-to-NIC Ethernet fabric segment. Intra-node communication (proprietary accelerator interconnects, e.g., NVLink, Infinity Fabric/xGMI, or PCIe) and individual GPU/accelerator performance are explicitly out of scope. Collective operation measurements (AllReduce, AllGather, AllToAll) are measured at the Ethernet fabric boundary; intra-node accelerator-interconnect contributions are reported separately when characterizing wide Expert Parallelism (wide-EP) or multi-node configurations.

The methodology is designed for controlled laboratory environments per the BMWG charter; it is NOT intended for production network measurement.

1.3. Relationship to Existing BMWG Work

Document	Relationship
[RFC1242]	Base terminology for network benchmarking; terms reused herein
[RFC2544]	Base methodology; throughput/latency/loss tests adapted for RDMA
[RFC2889]	LAN switching methodology; MAC learning concepts adapted for Address Resolution Protocol (ARP) / Neighbor Discovery (ND) scale
[RFC8238]	Data center terminology; buffer, congestion, and microburst terms extended
[RFC8239]	Data center methodology; line-rate and buffer tests adapted for RoCEv2
[RFC9004]	Back-to-back frame updates; burst absorption methodology referenced
[LLM-BENCH]	Complementary document benchmarking the inference serving stack. Treats the network as opaque SUT. This document benchmarks the fabric itself. The two documents MAY be used together but MUST NOT be combined in a single benchmarking report without explicit section demarcation.
[UEC-1.0]	UET protocol specification; transport services, congestion control, and link-layer enhancements benchmarked in Section 6

Table 1: Relationship to Existing BMWG Work

2. Terminology

Terminology used in this document is defined in [TERMINOLOGY]. Readers should consult that document before applying the methodology defined here. Where a term overlaps with [RFC1242] or [RFC8238], the terminology document provides AI fabric context extensions; the foundational definitions in those RFCs remain authoritative for general network benchmarking.

All terminology used in this document — including the AI fabric, RoCEv2, UET, RDMA transport, congestion control (PFC, DCQCN, ECN, CBFC), load balancing (ECMP, Packet Spray, DLB/Flowlet), collective communication, and KPI vocabulary (JCT, JCT Ratio, BusBW, MMR, etc.) — is defined normatively in [TERMINOLOGY] and is not redefined here. The following table lists the single bench-specific extension introduced by this document:

Term	Definition
PFC Pause Event	A single PFC PAUSE frame transmitted on a priority class. Used in this document as the unit of count for PFC event-rate metrics (events/sec, cumulative duration) reported by the methodology in Section 7.

Table 2: Bench-Specific Terminology Extensions

In addition to the BusBW reporting requirements specified in [TERMINOLOGY], the runtime algorithm selected by the collective library MUST be verified via library tracing and documented as part of the test conditions for any AllReduce, AllGather, ReduceScatter, or AllToAll benchmark in this document.

The scope of the DUT for the tests defined in this document is the set of leaf switches, spine switches, superspine switches (if applicable), and interconnecting links forming the AI training fabric, consistent with the Fabric DUT Boundary defined in [TERMINOLOGY].

2.1. Acronyms

Acronyms used in this document are expanded in the Acronyms appendix of [TERMINOLOGY]. Acronyms unique to the methodology defined herein are expanded on first use in the body of this document.

3. Test Topology and Architecture

3.1. Reference Fabric Topologies

Three reference topologies are defined. Every test report identifies which topology was used. Results obtained under different topologies are not directly comparable without normalization.

3.1.1. Topology A: 2-Tier Clos (Leaf-Spine)

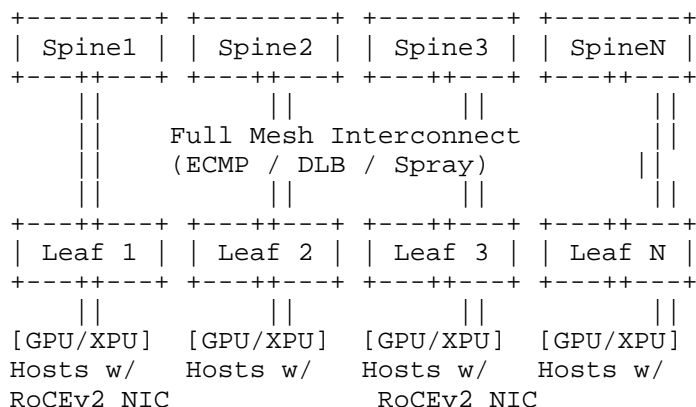


Figure 1: Topology A: 2-Tier Clos (Leaf-Spine)

The DUT boundary encompasses all leaf and spine switches and their interconnecting links. Traffic generators or actual GPU hosts connect at the leaf layer.

3.1.2. Topology B: 3-Tier Clos (Leaf-Spine-Superspine)

For clusters exceeding thousands of accelerators, a superspine layer is added. Each pod consists of a leaf-spine fabric; pods interconnect via superspine switches. This topology scales to 32,000+ accelerators at 800GbE with current-generation ASICs. The DUT boundary encompasses all three tiers.

3.1.3. Topology C: Rail-Optimized

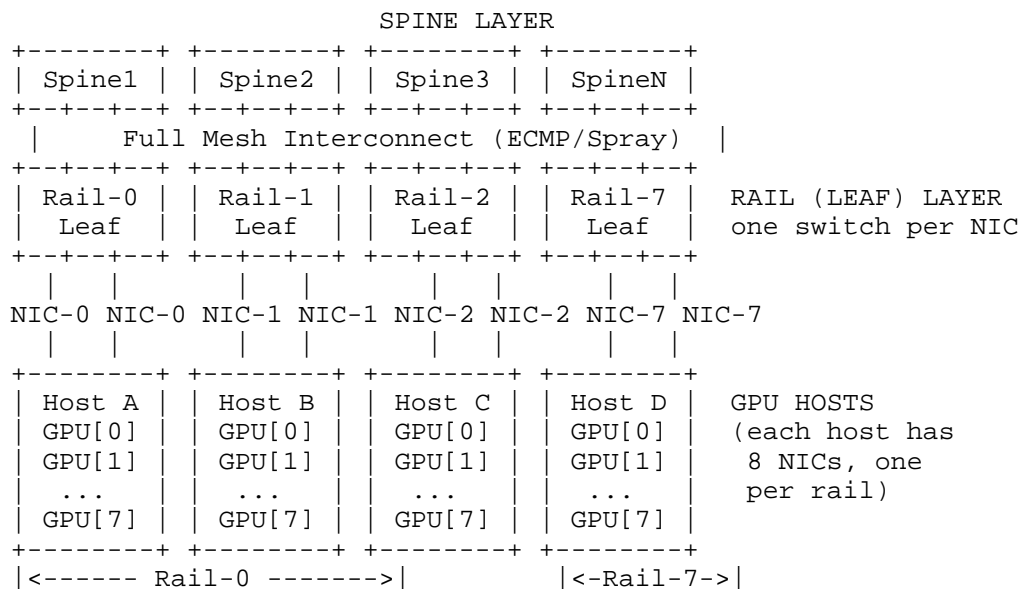


Figure 2: Topology C: Rail-Optimized

In rail-optimized topologies, each NIC on a multi-NIC host connects to a dedicated leaf switch ("rail"), co-optimizing network locality with the collective communications library (CCL) in use (e.g., NCCL, RCCL, oneCCL). The DUT boundary and rail mapping are fully documented in the test report.

3.2. Device Under Test (DUT) Identification

Parameter	Description	Example
Switch Vendor/Model	Vendor name, product family, model number	Vendor Family Model
Switch ASIC	Silicon vendor, ASIC family, revision	Silicon Vendor ASIC Family Rev
NOS Version	Network operating system name and version	NOS Name Version
Port Speed	Per-port line rate	400GbE, 800GbE

Buffer Architecture	Shared/dedicated, total buffer per ASIC/port	32MB shared + 16MB VOQ per port
Optics/Cables	Transceiver type, cable type and length	Octal Small Form-factor Pluggable (OSFP) 400G-DR4, Direct Attach Copper (DAC) 3m cable
NIC Vendor/Model	RDMA NIC vendor, model, firmware	NIC Vendor Model Speed
NIC Firmware	NIC firmware version	Firmware Version
Host Config	OS, CCL lib version, driver, BIOS settings	OS Version, CCL Version, OFED Version

Table 3: DUT Identification Parameters

3.3. Traffic Generator Requirements

3.3.1. Mandatory Functional Capabilities

The traffic generator supports: RoCEv2 transport emulation (QP establishment, RDMA Write/Read, ECN processing, DCQCN rate control); configurable QP scaling (1-256 QPs per source-destination pair); programmable collective communication patterns (AllReduce, AllToAll, AllGather with configurable message sizes); and nanosecond-precision timestamping.

3.3.2. Minimum Measurement Accuracy Requirements

Parameter	Minimum Requirement
Timestamp accuracy	<= 100 nanoseconds
Frame rate accuracy	+/- 0.1% of specified rate
QP scaling range	1 to 256 QPs per src-dst pair
Message size range	1 KB to 8 GB
Flow counter resolution	Per-flow byte and packet counts

Loss measurement	0 ppm resolution	
Burst generation	1-1000 frames at line rate	

Table 4: Minimum Measurement Accuracy Requirements

3.3.3. Acceptable Implementations

The platform used is identified in all test reports.

(a) Hardware Traffic Generator — dedicated hardware capable of line-rate RDMA emulation meeting the Measurement Accuracy Requirements specified in this document. Suitable for point-to-point RDMA tests (Section 5 and Section 6). For collective tests (Section 9), the following limitations are documented: whether synchronization barriers are reproduced, whether flow patterns are schedule-driven or gradient-driven, and whether straggler behavior is modeled.

(b) Accelerator Cluster — cluster running an actual collective communication library with RDMA tooling. Preferred for the collective benchmarks in Section 9. Host configuration (accelerator model, collective library name and version, PCIe topology, BIOS power management settings) is documented. Any non-fabric overhead in timing measurements is quantified and reported separately.

When a hardware generator is used for collective benchmarks, results should be cross-validated against an accelerator cluster at one or more overlapping (message_size, N) configurations.

Discrepancies exceeding 10% in BusBW or JCT Ratio are investigated and reported.

4. KPI Framework and Metrics Taxonomy

NOTE: Per BMWG charter, the definition of acceptance criteria or performance requirements is explicitly outside the scope of this Working Group. The KPI tables in this section define what is measured and how it is reported; they do not set thresholds. Indicative non-normative reference values reflecting current industry observations are provided in Appendix B; those values MUST NOT be used as pass/fail thresholds in vendor evaluations.

4.1. Primary KPIs

KPI	Unit	Definition
Job Completion Time (JCT)	seconds	Wall-clock time for benchmark iteration (compute + communication)
JCT Ratio	dimensionless	Measured JCT / Roofline JCT
Bus Bandwidth (BusBW)	Gbps/accelerator	Effective per-accelerator throughput during collective. See the BusBW definition in [TERMINOLOGY]
Aggregate Throughput	Tbps	Total fabric goodput during collective phase
Packet Drop Rate	ppm	Frames lost end-to-end not retransmitted
Tail Latency (P99/P99.9)	us	99th/99.9th percentile one-way fabric latency

Table 5: Primary KPIs

4.2. Secondary KPIs

KPI	Unit	Definition
ECN Marking Ratio	%	Percentage of packets marked CE over measurement interval
PFC Pause Count	events/sec	Rate of PFC PAUSE frames per priority per port
PFC Pause Duration	us	Cumulative time a port is in PFC-paused state per interval
RDMA Retransmission	retx/sec	NIC-level

Rate		retransmissions due to timeouts or NAKs
ECMP Imbalance (MMR)	dimensionless	Max-Mean Ratio of flow counts across parallel uplinks
Jain Fairness Index (JFI)	0.0-1.0	Fairness of traffic distribution; 1.0 = perfect
Queue Depth (P95/Max)	bytes or cells	95th percentile and maximum egress queue occupancy per port
Congestion Control Convergence	us	Time from congestion onset to DCQCN rate stabilization
Out-of-Order Packet Rate	pkt/sec	Packets delivered out of sequence (relevant for packet spray)
Clear-to-Send (CTS) / Acknowledgment (ACK) Delay	us	Delay for control messages (Clear-to-Send, ACKs)

Table 6: Secondary KPIs

4.3. Fabric Health Indicators

Indicator	Unit	Definition
Switch CPU Utilization	%	Average and peak CPU usage on DUT control plane during test
Switch Memory Utilization	%	Average and peak memory usage, including FIB/MAC table occupancy
Forwarding Information Base (FIB) / Route Convergence Time	ms	Time to converge routing after topology change
Link Flap Count	events	Spurious link state changes during test period

CRC/FCS Error Rate	errors/ sec	Physical layer errors indicating cable or optics issues
Power Consumption	Watts	Per-switch and per-port power draw under test load

Table 7: Fabric Health Indicators

5. Test Category 1: RDMA Transport Benchmarks

These tests establish baseline fabric performance for RDMA traffic independent of collective communication patterns. They extend [RFC2544] and [RFC8239] methodology for RoCEv2 semantics.

5.1. Baseline Throughput

Objective: Determine the maximum sustainable RDMA Write throughput through the DUT fabric at each tested message size.

Procedure:

- * Configure N host pairs, each establishing Q Queue Pairs per pair
- * Initiate RDMA Write operations and measure aggregate goodput
- * Each test runs for at least 60 seconds at each rate
- * Binary search per [RFC2544] Section 26.1 is used
- * Message sizes: 64B, 256B, 1KB, 4KB, 64KB, 256KB, 1MB, 4MB
- * QP counts: 1, 4, 16, 32 per src-dst pair
- * Test both unidirectional and bidirectional traffic

Reporting: Report aggregate throughput (Tbps), per-port utilization (%), and throughput efficiency (measured/theoretical). Present as table indexed by message size x QP count, and as graph (message size on X-axis).

5.2. Latency Characterization

Objective: Determine one-way and round-trip RDMA latency distribution at the throughput rate from Section 5.1.

Procedure:

- * Inject tagged frames at 60s into a 120s stream (per [RFC2544] Section 26.2)
- * Nanosecond-precision timestamping
- * Reported statistics: min, mean, P50, P95, P99, P99.9, max
- * Repeat at least 20 times; report averages
- * Test under both zero-load (single QP) and loaded (full fabric utilization) conditions

Reporting: Tabulate latency statistics per message size. Provide histogram and CDF plot. Report latency increase factor (loaded/unloaded).

5.3. Back-to-Back Burst Absorption

Objective: Characterize the DUT fabric's ability to absorb back-to-back RDMA bursts without loss, extending [RFC9004] methodology for RoCEv2.

Procedure:

- * Transmit bursts at line rate with minimum inter-frame gap
- * Increase burst length until first frame loss is detected
- * Test incast ratios: 2:1, 4:1, 8:1, 16:1, 32:1
- * Repeat at least 50 times per burst length

Reporting: Report burst absorption capacity (frames and bytes) for each message size and incast ratio. Plot burst capacity vs. incast ratio.

6. Test Category 1A: UEC Transport Protocol Benchmarks

The Ultra Ethernet Consortium (UEC) Specification 1.0 [UEC-1.0] defines UET, a connectionless RDMA transport designed to replace RoCEv2 for AI/HPC workloads. All UET tests use the libfabric API [LIBFABRIC] and run on UEC 1.0-compliant NICs.

The UEC compliance profile (AI Base, AI Full, or HPC) used during testing is documented in the test report.

6.1. UET Throughput by Transport Service

Objective: Determine maximum sustainable throughput under each UET transport service (ROD, RUD, RUDI, UUD) and compare to RoCEv2 Reliable Connected (RC) / Unreliable Connected (UC) on the same DUT fabric.

Procedure: Use UEC 1.0-compliant NICs; establish PDCs; use libfabric fi_write. Apply binary search ([RFC2544] Section 26.1). Vary PDC counts: 1, 4, 16, 32. A parallel RoCEv2 test series is executed for comparison. Both unidirectional and bidirectional configurations are tested.

Reporting template:

Metric	ROD	RUD	RUDI	UUD	RoCEv2 RC	RoCEv2 UC
Throughput @ 1MB (Gbps)	(meas)	(meas)	(meas)	(meas)	(meas)	(meas)
Throughput @ 4MB (Gbps)	(meas)	(meas)	(meas)	(meas)	(meas)	(meas)
Efficiency (% line rate)	(meas)	(meas)	(meas)	(meas)	(meas)	(meas)
PDC/QP Setup Time (us)	(meas)	(meas)	(meas)	(meas)	(meas)	(meas)
Max Sustained PDC/QP Count	(meas)	(meas)	(meas)	(meas)	(meas)	(meas)

Table 8: UET Throughput by Transport Service

6.2. UET Latency Characterization

Objective: Measure latency distribution for UET transport services; quantify differential vs. RoCEv2, with particular attention to connectionless PDC establishment overhead.

***Procedure:** Measure latency for: (a) steady-state PDC transfers; (b) first-packet latency (PDC + first data packet, measuring "data before handshake"); (c) zero-load baseline. Test ROD and RUD separately to isolate reordering-related latency.

***Reporting:** Tabulate latency statistics per (transport_service, message_size, load_condition) tuple. Plot latency CDF for UET ROD, UET RUD, and RoCEv2 RC side-by-side.

6.3. Packet Spray Efficacy Under UET RUD

***Objective:** Quantify the load balancing improvement achieved by UET's native per-packet spray with RUD, which eliminates the receiver reorder buffer constraint.

***Procedure:** Test four configurations:

- * UET RUD + packet spray
- * UET ROD + packet spray
- * RoCEv2 RC + packet spray
- * RoCEv2 RC + standard ECMP (baseline)

Measure MMR, JFI, out-of-order delivery rate, retransmission rate, and effective goodput. Vary ECMP paths: 4, 8, 16, 32.

***Reporting template:**

Load Balancing Config	MMR	JFI	OOO Rate	Retx Rate	Effective Goodput (%)
UET RUD + Packet Spray	(meas)	(meas)	(meas)	(meas)	(meas)
UET ROD + Packet Spray	(meas)	(meas)	(meas)	(meas)	(meas)
RoCEv2 RC + Packet Spray	(meas)	(meas)	(meas)	(meas)	(meas)
RoCEv2 RC + ECMP (baseline)	(meas)	(meas)	(meas)	(meas)	(meas)
UET RUD + DLB/ Flowlet	(meas)	(meas)	(meas)	(meas)	(meas)

Table 9: Packet Spray Efficacy Under UET RUD

UET RUD is expected to achieve zero host-visible reordering despite per-packet spray because the transport layer natively tolerates unordered delivery.

6.4. UET Congestion Control Benchmarks

Objective: Evaluate UET's dual-sided (sender + receiver) congestion control under N:1 incast conditions vs. RoCEv2 DCQCN.

Procedure: Measure: (a) incast throughput at $N = \{2, 4, 8, 16, 32, 64\}$; (b) convergence time after doubling active senders (until all flows within 10% of fair share); (c) PFC avoidance with PFC disabled on the DUT; (d) receiver credit utilization.

Reporting: Tabulate incast throughput, convergence time, peak queue depth, PFC event count, and packet drop rate for UET vs. DCQCN per incast ratio. ***Critical differentiator:*** report whether UET achieves zero application-visible loss without PFC.

6.5. Link Layer Enhancement Benchmarks

Objective: Measure performance impact of optional link-layer enhancements: LLR, Packet Trimming (PT), and CBFC.

Procedure:

- * ***(a) LLR Retry Latency:** inject controlled bit errors; measure LLR retry latency (expected sub-microsecond per hop) vs. transport-layer retransmission (~10-100us RTT). Run with 80% background load.
 - * ***(b) Packet Trimming Effectiveness:** configure 2:1 oversubscription bottleneck; measure time from congestion onset to first retransmission request, bandwidth saved vs. full-packet drops.
 - * ***(c) CBFC vs. PFC:** identical N:1 (N=32) incast scenarios; measure head-of-line blocking duration (CBFC is per-destination, PFC is per-priority), pause propagation hops, and throughput of non-congested flows.
- *Reporting:** Before/after comparison table for each enhancement. Note which features are hardware-supported vs. software-emulated.

6.6. UET Collective Communication Performance

***Objective:** Measure collective communication (AllReduce, AllToAll, AllGather) performance over UET and compare directly to RoCEv2, isolating the transport protocol contribution to collective efficiency.

***Procedure:** Execute the collective benchmark suite from Section 9 over UET RUD transport using a UEC-compliant collective library. The same accelerator count (N), message sizes, and fabric topology are used for both UET and RoCEv2 runs to ensure a valid comparison. Run UET RUD + packet spray as the primary configuration and UET ROD + ECMP as the secondary baseline.

For AllReduce, if UET group keying (transport-layer reduction support per UEC Spec 1.0) is active on the DUT NIC, this is noted explicitly in the test report.

When UET group keying is active during testing, report the observed BusBW computed from measured bytes transferred. The algo_factor defined in [TERMINOLOGY] (fixed per collective type) still applies to the formula; the observed transfer volume reflects group keying behavior.

Document the group keying state (active / inactive) as a required result field. The runtime algorithm in use is reported per message-size bucket. See [TERMINOLOGY] for the BusBW definition and algo_factor values.

***Reporting:** Report the percentage improvement in BusBW and JCT attributable to UET native packet spray and congestion control.

***Reporting template:**

Collective	Msg Size	N Accels	UET RUD BusBW	UET ROD BusBW	RoCEv2 RC BusBW	Delta UET/ RoCEv2
AllReduce	1GB	128	(meas)	(meas)	(meas)	(meas)
AllReduce	1GB	512	(meas)	(meas)	(meas)	(meas)
AlltoAll	1GB	128	(meas)	(meas)	(meas)	(meas)
AllGather	1GB	128	(meas)	(meas)	(meas)	(meas)

Table 10: UET Collective Communication Performance

6.7. UET PDC Scalability and Connection Setup Rate

***Objective:** Measure PDC establishment rate and maximum concurrent PDC count vs. RoCEv2 QP-based connections.

***Procedure:** (a) PDC establishment rate: initiate PDC creation to $M = \{100, 1000, 10000, 100000\}$ remote endpoints. (b) Data-before-handshake: measure first-byte latency for UET vs. RoCEv2 RDMA Write. (c) Maximum concurrent PDC count: scale until per-PDC throughput drops below 90% of single-PDC rate. The UEC specification [UEC-1.0] targets up to 1 million endpoints.

7. Test Category 2: Congestion Management

AI training workloads generate repetitive micro-congestion during the back-propagation gradient synchronization phase.

7.1. ECN Marking Accuracy and Threshold

***Objective:** Verify that the DUT marks packets with ECN CE at the configured threshold with correct granularity.

***Procedure:** Configure threshold T on DUT egress queue. Verify: (a) no packets marked below T ; (b) 100% marked above maximum threshold; (c) appropriate Weighted Random Early Detection (WRED) / Random Early Detection (RED) probability ramp between thresholds. Test thresholds: low (~100KB), medium (~1MB), high (~5MB).

Reporting: Plot ECN marking probability vs. instantaneous queue depth. Report measured threshold accuracy (deviation from configured).

7.2. PFC Behavior Under Incast

Objective: Characterize DUT's PFC generation behavior under N:1 incast conditions.

Procedure: Generate N:1 incast at 100% line rate, $N = \{2, 4, 8, 16, 32, 64\}$. Measure PFC PAUSE frame count/sec per hop, PFC PAUSE duration per port, PFC storm onset, and end-to-end throughput. The test characterizes headroom sizing and PFC watchdog effectiveness.

7.3. DCQCN Convergence Time

Objective: Measure time for DCQCN to converge to fair-share rate after congestion onset.

Procedure: Establish M flows through a common bottleneck. At T₀, inject additional M flows (creating 2:1 oversubscription). Measure time until all 2M flows achieve rates within 10% of fair share. Repeat for $M = \{4, 16, 64, 256\}$. Vary DCQCN parameters and report sensitivity.

7.4. PFC Storm and Deadlock Resilience

Objective: Verify the DUT does not enter PFC deadlock or sustained PFC storm under adversarial traffic.

Procedure: Generate cyclic traffic patterns known to cause PFC deadlocks. Run for 300 seconds. The test characterizes whether the DUT demonstrates resilience via PFC watchdog or architectural immunity (e.g., VOQ-based scheduling); the mechanism observed is reported.

8. Test Category 3: Load Balancing Efficacy

Load balancing across parallel fabric paths is critical for AI training fabrics because the traffic consists of a small number of high-bandwidth, long-lived elephant flows.

8.1. ECMP Entropy and Polarization

Objective: Quantify traffic polarization under standard ECMP hashing for AI training flow patterns.

***Procedure:** Configure standard 5-tuple ECMP. Generate traffic with $Q = \{1, 4, 8, 16, 32\}$ QPs per src-dst pair. Measure per-link utilization, MMR, and JFI. Test with and without BTH-aware hashing. Repeat for fabric sizes of 8, 16, 32, and 64 leaf switches.

8.2. Dynamic Load Balancing (Flowlet)

***Objective:** Evaluate DUT's flowlet-based DLB performance and compare to baseline ECMP.

***Procedure:** Configure vendor-specific DLB (document algorithm type). Generate traffic with $Q=4$ QPs. Measure MMR, JFI, per-link utilization, out-of-order rate. Vary flowlet gap timer and report sensitivity.

8.3. Packet Spraying

***Objective:** Evaluate DUT's per-packet spraying performance and quantify the utilization vs. reordering tradeoff.

***Procedure:** Configure per-packet load balancing. Measure MMR (expected ~1.0), JFI (expected ~1.0), out-of-order rate, and RDMA retransmission impact. If the DUT provides an in-fabric reorder buffer, document per Appendix C.

8.4. Jain Fairness Index Measurement

***Objective:** Single-number summary of load balancing quality comparable across all strategies.

***Formula:**

$$JFI = (\text{Sum LinkTx}_i)^2 / (N * \text{Sum LinkTx}_i^2)$$

Figure 3: Jain Fairness Index Formula

where LinkTx_i = transmitted traffic on fabric link i , N = total parallel links. Range: $1/N$ (worst) to 1.0 (perfect).

***Reporting:** Report JFI for each load balancing strategy. Provide bar chart comparing ECMP, DLB, and packet spray.

9. Test Category 4: Collective Communication Benchmarks

These tests evaluate the fabric's performance under realistic collective communication patterns. Unlike synthetic RDMA tests in Section 5 and Section 6, these exercise the full stack including the collective communications library (CCL) in use (e.g., NCCL, RCCL, oneCCL).

9.1. AllReduce Benchmark

Objective: Measure fabric performance during AllReduce operations, the dominant collective for gradient synchronization in data-parallel training.

Procedure: Using N accelerators connected through the DUT fabric, execute AllReduce (sum) operations using a collective communications library benchmark suite (e.g., nccl-tests, rccl-tests, or equivalent).

Test parameters:

- * Message sizes: 1 MB, 8 MB, 64 MB, 256 MB, 1 GB, 4 GB
- * Accelerator counts (N): 8, 16, 32, 64, 128, 256, 512, 1024
- * Minimum iterations per (message_size, N) pair: 100
- * Load balancing strategies: ECMP, DLB, packet spray

For each (message_size, N) pair, record average, P50, P95, and P99 BusBW, ECN marking ratio, PFC pause count, and per-link utilization. BusBW is computed per the BusBW definition in [TERMINOLOGY]; algo_factor is fixed per collective type and does not vary with the algorithm the library selects at runtime. The runtime algorithm selected by the library for each message-size bucket is verified via library tracing and documented as part of the test conditions.

Reporting: Tabulate BusBW for each (message_size, N, LB_strategy, Algorithm (verified)) combination. The "Algorithm (verified)" column is required; results without it are incomplete. Plot BusBW vs. N for each message size. Report BusBW efficiency = BusBW / NIC_line_rate.

9.2. AlltoAll Benchmark

Objective: Measure fabric performance during AllToAll operations, the dominant collective for Mixture-of-Experts (MoE) expert parallelism dispatch and pipeline-parallel communication.

***Procedure:** Using the same message sizes, accelerator counts, iteration count, and load balancing strategies as Section 9.1, execute AllToAll operations via the collective communication library.

AllToAll generates the worst-case fabric stress pattern: every accelerator simultaneously sends a unique payload to every other accelerator in the group, creating maximum entropy and N-to-N incast at every fabric link. This makes AllToAll JCT the most sensitive single indicator of fabric congestion management quality.

BusBW is computed per the BusBW definition in [TERMINOLOGY]; algo_factor is fixed per collective type and does not depend on topology or library implementation. The runtime algorithm in use is verified via library tracing and documented as part of the test conditions.

***Measurement:** Report BusBW (average, P50, P95, P99), JCT per iteration, ECN marking ratio, PFC pause count, and per-link utilization for each (message_size, N, LB_strategy) combination.

***Reporting:** Same table format as Section 9.1, with the "Algorithm (verified)" column required. Additionally report JCT for each configuration; JCT degradation relative to the ECMP baseline is highlighted as the primary congestion sensitivity indicator.

9.3. AllGather Benchmark

***Objective:** Measure fabric performance during AllGather operations, the dominant collective for weight and activation distribution in tensor-parallel and pipeline-parallel training.

***Procedure:** Using the same message sizes, accelerator counts, iteration count, and load balancing strategies as Section 9.1, execute AllGather operations via the collective communication library.

AllGather consists of a gather phase only — each accelerator contributes a shard and receives the full concatenated tensor. There is no reduce phase, which produces lower peak fabric load than AllReduce at equivalent message size and N. This makes AllGather a useful baseline for isolating the gather-path fabric contribution from the combined send-and-reduce cost.

BusBW is computed per the BusBW definition in [TERMINOLOGY]; algo_factor is fixed per collective type and does not depend on the library's algorithm selection. The runtime algorithm in use is verified via library tracing and documented as part of the test conditions.

***Measurement:** Report BusBW (average, P50, P95, P99), JCT per iteration, ECN marking ratio, PFC pause count, and per-link utilization for each (message_size, N, LB_strategy) combination.

***Reporting:** Same table format as Section 9.1, with the "Algorithm (verified)" column required. Report BusBW efficiency = BusBW / NIC_line_rate. Where results are compared to AllReduce under identical parameters, the BusBW ratio (AllGather / AllReduce) quantifies the fabric overhead attributable to the reduce phase.

9.4. Collective Communication Library Bus Bandwidth Summary

***Reporting template:**

Collective	Msg Size	N Accels	ECMP BusBW (Gbps/accel)	DLB BusBW (Gbps/accel)	Spray BusBW (Gbps/accel)
AllReduce	1GB	128	(meas)	(meas)	(meas)
AllReduce	1GB	512	(meas)	(meas)	(meas)
AlltoAll	1GB	128	(meas)	(meas)	(meas)
AlltoAll	1GB	512	(meas)	(meas)	(meas)
AllGather	1GB	128	(meas)	(meas)	(meas)
AllGather	1GB	512	(meas)	(meas)	(meas)

Table 11: Collective Communication Bus Bandwidth Summary

10. Test Category 5: Job Completion Time (JCT) Benchmarks

JCT is the single most important user-facing KPI for AI training fabrics, directly determining accelerator utilization and training cost.

10.1. Synthetic JCT Under Controlled Conditions

***Objective:** Measure JCT for a defined synthetic workload with a known computation-to-communication ratio to isolate fabric-induced overhead.

***Procedure:** Define a synthetic training iteration as a strictly sequential model:

1. Computation phase of C milliseconds (simulated sleep or GPU compute kernel)
2. Communication phase: AllReduce of S bytes across N accelerators

Parameter	Values
Computation time C	10 ms, 50 ms, 100 ms, 500 ms
Message size S	256 MB, 1 GB, 4 GB
Accelerator count N	64, 128, 256, 512, 1024
Iterations	1000

Table 12: Synthetic JCT Test Parameters

Execute 1000 iterations and measure total wall-clock JCT.

$$\text{Roofline_seq} = \text{Iterations} \times (\text{C} + (8 \times \text{S} \times \text{algo_factor}) / \text{B_acc})$$
$$\text{JCT Ratio} = \text{Measured_JCT} / \text{Roofline_seq}$$

where:

C

= compute time per iteration (seconds)

S

= message size per iteration (bytes)

algo_factor

= fixed normalization constant per collective type;
see the BusBW definition in {{TERMINOLOGY}}

B_acc

= aggregate per-accelerator NIC line rate (bits/second);
sum across all NICs serving the accelerator (e.g., in
rail-optimised topologies, the sum of all rail NIC speeds)

Iterations

= number of synthetic iterations executed

The factor of 8 converts S from bytes to bits to match the units of B_acc.

Figure 4: JCT Ratio Calculation

This model assumes strictly sequential compute and communication phases and represents a conservative upper bound on communication overhead. Many frameworks overlap these phases via gradient bucketing or asynchronous collectives, reducing the effective communication overhead visible in wall-clock JCT.

Implementations using overlapped execution additionally report:

$$\text{Overlap_Fraction} = 1 - (\text{Measured_JCT} - C_{\text{total}}) / \text{Comm_time}$$

where:

$$C_{\text{total}} = \text{Iterations} \times C$$

$$\text{Comm_time} = \text{Iterations} \times (8 \times S \times \text{algo_factor}) / B_{\text{acc}}$$

S , algo_factor , B_{acc} as defined for `Roofline_seq` above.

Figure 5: Overlap Fraction Calculation

An `Overlap_Fraction` of 0 indicates fully sequential execution; 1.0 indicates communication is perfectly hidden behind compute.

When overlap is present, the residual fabric overhead is reported as:

$$\text{Effective_Comm_Overhead} = \text{Measured_JCT} - C_{\text{total}}$$

The `Overlap_Fraction` and communication-library overlap configuration (e.g., bucket size, number of async streams) are documented as part of the test configuration when this optional measurement is reported.

***Reporting:** Tabulate JCT Ratio for each (C , S , N , LB_strategy) combination. Plot JCT Ratio vs. N to characterize fabric scalability.

NOTE: JCT Ratio < 1.05 indicates excellent fabric performance; 1.15 indicates significant fabric-induced overhead. These are non-normative illustrative reference values only.

10.2. MLPerf-Aligned JCT

***Objective:** Measure JCT using MLPerf Training benchmark workloads [MLPERF] to enable comparison with published industry results.

***Procedure:** Execute MLPerf Training closed-division workloads (e.g., BERT, ResNet, GPT-3 175B) per MLPerf submission rules. Simultaneously capture all fabric KPIs from Section 4. Report time-to-train and/or tokens-per-second.

10.3. Multi-Tenant JCT Interference

***Objective:** Quantify JCT impact when multiple training jobs share the same fabric.

***Procedure:** Configure two or more independent training jobs. Jobs are configured to overlap in spine-layer link usage. Measure baseline JCT (isolated) and contention JCT (simultaneous).

$$\text{JCT Interference Factor} = \text{Contention_JCT} / \text{Baseline_JCT}$$

Figure 6: JCT Interference Factor

Test with spine link overlap: 0%, 25%, 50%, 75%.

11. Test Category 6: Scale and Convergence

11.1. Fabric Scale Limits

***Objective:** Determine the maximum fabric scale at which the DUT maintains acceptable KPI performance.

***Procedure:** Progressively increase active accelerator endpoints from N=64 to maximum topology support while running AllReduce (Section 9.1, S=1GB). At each scale point record JCT Ratio, BusBW, ECN ratio, PFC count, CPU and memory utilization. Also measure BGP/routing convergence time after clearing all adjacencies (analogous to [EVPN-BENCH] Sections 3.10, 3.11, 4.9, 4.10).

11.2. Link Failure Convergence

***Objective:** Measure traffic disruption and JCT impact when a fabric link fails during active training.

***Procedure:** With the fabric fully loaded (AllReduce, N=128, S=1GB), administratively fail a spine uplink. Measure:

- * Duration of packet loss
- * Packets lost
- * JCT overhead for the failure iteration vs. steady state
- * Time for load balancing mechanism to redistribute flows

Repeat for: leaf uplink failure, spine switch failure, superspine link failure (if applicable). Test under each load balancing strategy.

11.3. Zero-Impact Failover Measurement

***Objective:** Verify vendor claims of zero-impact or sub-microsecond failover.

***Procedure:** Execute Section 11.2 with nanosecond-precision measurement. A failure is considered "zero-impact" if the measured JCT for the failure iteration is within the P99 JCT of steady-state iterations.

12. Test Category 7: Soak and Stability

12.1. 24-Hour Sustained Load

***Objective:** Verify DUT fabric stability under sustained AI training load over an extended period, following the methodology pattern from [EVPN-BENCH] Sections 3.12, 4.11.

***Procedure:** Configure DUT at maximum validated scale from Section 11.1. Generate bidirectional collective communication traffic (alternating AllReduce and AlltoAll). Run continuously for 24 hours. Sample all KPIs from Section 4 every 60 seconds.

The DUT is expected to exhibit no memory leaks, crashes, or CPU spikes; any anomaly is reported with timestamp and duration.

***Reporting:** Time-series plots of JCT Ratio, BusBW, ECN ratio, PFC count, CPU, and memory over the 24-hour period. Report standard deviation of JCT Ratio (stability metric).

12.2. Resource Leak Detection

***Objective:** Detect memory leaks, handle exhaustion, or gradual performance degradation in DUT software.

***Procedure:** Record per-process memory usage at T=0, T=1h, T=6h, T=12h, T=24h. Compute linear regression slope of memory usage over time. A slope exceeding *1 MB/hour* for any process indicates a potential memory leak and is reported. Also monitor forwarding-plane counter wraparounds and hardware table occupancy trends.

13. Reporting Format

Test reports include the following sections:

1. ***DUT Identification:** Complete parameters from Section 3.2 for all fabric components.
2. ***Test Topology:** Diagram and description per Section 3.1, including physical cabling.

3. ***Test Configuration:** All DUT configuration parameters: QoS policies (ECN thresholds, PFC headroom, DCQCN parameters), load balancing mode, buffer allocation, and vendor-specific tuning.
4. ***Host Configuration:** Complete host stack description per Section 3.2 including NIC firmware, driver, collective library version, and any tuning. For UET tests, additionally report: UEC compliance profile, libfabric provider version, NIC UEC firmware version, and enabled optional link-layer features (LLR, Packet Trimming, PRI, CBFC).
5. ***Test Results:** For each test from Section 5 through Section 12, provide specified tables, graphs, and statistical summaries. For Section 6 tests, results include side-by-side UET vs. RoCEv2 comparison data on the identical DUT fabric.
6. ***Anomalies:** Any deviations from specified procedures, test failures, or unexpected behaviors are documented.
7. ***Repeatability Statement:** Report iteration count and coefficient of variation (std deviation / mean) for each test's primary metric. A CV below 5% is recommended for test validity.

14. Security Considerations

This document defines benchmarking methodology for controlled laboratory environments and does not specify any protocol mechanism. It therefore introduces no new protocol-level security considerations beyond those of the underlying technologies it references. The considerations below follow the BMWG convention established in [RFC8238] and align with the companion terminology document [TERMINOLOGY].

Benchmarking activities as described in this document are limited to technology characterization of AI training fabrics using controlled stimuli in a laboratory environment, with dedicated address space and the constraints specified herein.

The benchmarking network topology will be an independent test setup and MUST NOT be connected to devices that may forward the test traffic into a production network or misroute traffic to the test management network. This isolation requirement is particularly important for AI fabric benchmarking because the lossless transport modes referenced in this document (PFC, DCQCN, CBFC) propagate congestion hop-by-hop and can extend the blast radius of a misconfigured test beyond the immediate DUT.

Benchmarking is performed on a "black-box" basis, relying solely on measurements observable external to the DUT as defined in [TERMINOLOGY].

Special capabilities SHOULD NOT exist in the DUT specifically for benchmarking purposes. Any implications for network security arising from the DUT SHOULD be identical in the lab and in production networks. In particular, RDMA memory-region permissions are properties of the deployed configuration, not of the benchmarking methodology, and SHOULD reflect production posture during testing.

Per [RFC6815], the tests defined herein MUST NOT be performed on production networks. The use of dedicated test IP address ranges per [RFC2544] Appendix C (198.18.0.0/15 for IPv4; 2001:db8::/32 per [RFC3849] for IPv6) is RECOMMENDED to prevent accidental interaction with production infrastructure.

The following considerations are specific to the methodology defined in this document:

- * *PFC leakage:* PFC PAUSE frames generated under incast or storm conditions (Section 7.2, Section 7.4) that escape the test environment can hang adjacent production switches sharing the same priority class. Physical or VLAN-based isolation of the test fabric is required.
- * *Line-rate RDMA traffic generators:* the equipment specified in Section 3.3 is capable of saturating production links at line rate; such generators MUST be confined to the test fabric.
- * *PFC disabled in Section 6.4:* the UET PFC-free incast test deliberately disables PFC on the DUT. In this configuration, traffic leaking to adjacent infrastructure cannot be backpressured and will be dropped on the adjacent device's queues. Isolation is mandatory.
- * *RDMA QP and PDC namespace isolation:* when RDMA/RoCEv2 traffic is used, the test environment SHOULD be isolated from production RDMA fabrics to prevent QP number space collisions or inadvertent PFC propagation. When UET traffic is used (Section 6), the test environment MUST ensure that UDP port 4793 traffic does not leak to production networks and that PDC identifier spaces are isolated.
- * *UET transport security sub-layer (TSS):* SHOULD NOT be enabled during performance benchmarking unless transport security overhead is explicitly being measured.

15. IANA Considerations

This document makes no request of IANA.

16. References

16.1. Normative References

- [RFC1242] Bradner, S., "Benchmarking Terminology for Network Interconnection Devices", RFC 1242, DOI 10.17487/RFC1242, July 1991, <<https://www.rfc-editor.org/rfc/rfc1242>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC2544] Bradner, S. and J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC 2544, DOI 10.17487/RFC2544, March 1999, <<https://www.rfc-editor.org/rfc/rfc2544>>.
- [RFC2889] Mandeville, R. and J. Perser, "Benchmarking Methodology for LAN Switching Devices", RFC 2889, DOI 10.17487/RFC2889, August 2000, <<https://www.rfc-editor.org/rfc/rfc2889>>.
- [RFC6815] Bradner, S., Dubray, K., McQuaid, J., and A. Morton, "Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful", RFC 6815, DOI 10.17487/RFC6815, November 2012, <<https://www.rfc-editor.org/rfc/rfc6815>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8238] Avramov, L. and J. Rapp, "Data Center Benchmarking Terminology", RFC 8238, DOI 10.17487/RFC8238, August 2017, <<https://www.rfc-editor.org/rfc/rfc8238>>.
- [RFC8239] Avramov, L. and J. Rapp, "Data Center Benchmarking Methodology", RFC 8239, DOI 10.17487/RFC8239, August 2017, <<https://www.rfc-editor.org/rfc/rfc8239>>.
- [RFC9004] Morton, A., "Updates for the Back-to-Back Frame Benchmark in RFC 2544", RFC 9004, DOI 10.17487/RFC9004, May 2021, <<https://www.rfc-editor.org/rfc/rfc9004>>.

[TERMINOLOGY]

Calabria, F., Pignataro, C., Wu, Q., and G. Fioccola,
"Benchmarking Terminology for AI Network Fabrics", Work in
Progress, Internet-Draft, draft-calabria-bmwg-ai-fabric-
terminology-01, 21 April 2026,
<[https://datatracker.ietf.org/doc/html/draft-calabria-
bmwg-ai-fabric-terminology-01](https://datatracker.ietf.org/doc/html/draft-calabria-bmwg-ai-fabric-terminology-01)>.

[UEC-1.0] Ultra Ethernet Consortium, "Ultra Ethernet Transport (UET)
Specification 1.0", June 2025,
<<https://ultraethernet.org>>.

16.2. Informative References

[DCQCN-PAPER]

Zhu, Y., "Congestion Control for Large-Scale RDMA
Deployments", DOI 10.1145/2785956.2787484, 2015,
<<https://doi.org/10.1145/2785956.2787484>>.

[EVPN-BENCH]

Jacob, S. and K. Tiruveedhula, "Benchmarking Methodology
for EVPN and PBB-EVPN", Work in Progress, Internet-Draft,
draft-ietf-bmwg-evpntest-11, August 2023,
<[https://datatracker.ietf.org/doc/html/draft-ietf-bmwg-
evpntest-11](https://datatracker.ietf.org/doc/html/draft-ietf-bmwg-evpntest-11)>.

[LIBFABRIC]

OpenFabrics Interfaces Working Group, "libfabric: Open
Fabric Interfaces", n.d.,
<<https://ofiwg.github.io/libfabric/>>.

[LLM-BENCH]

Gaikwad, et al, "Benchmarking Methodology for Large
Language Model Serving", Work in Progress, Internet-Draft,
draft-gaikwad-llm-benchmarking-methodology-00, January
2026, <[https://datatracker.ietf.org/doc/html/draft-
gaikwad-llm-benchmarking-methodology-00](https://datatracker.ietf.org/doc/html/draft-gaikwad-llm-benchmarking-methodology-00)>.

[META-ROCE]

Gangidi, A., "RDMA over Ethernet for Distributed AI
Training at Meta Scale", DOI 10.1145/3651890.3672233,
2024, <<https://doi.org/10.1145/3651890.3672233>>.

[MLPERF]

MLCommons, "MLPerf Training Benchmark Suite", n.d.,
<<https://mlcommons.org>>.

[RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", RFC 3849, DOI 10.17487/RFC3849, July 2004, <<https://www.rfc-editor.org/rfc/rfc3849>>.

Appendix A. KPI-to-Test Mapping Summary

KPI	Test Section	Measurement Method	Reporting Unit
Throughput Rate	Section 5.1	Binary search, zero-loss	Tbps, % line rate
Latency (P99)	Section 5.2	Tagged frame, loaded / unloaded	us
Burst Absorption	Section 5.3	Max burst without loss	frames, bytes
ECN Accuracy	Section 7.1	Queue depth vs. marking	threshold deviation %
PFC Behavior	Section 7.2	Incast sweep N=2..64	PAUSE events/sec, duration
DCQCN Convergence	Section 7.3	Rate stabilization after onset	us
PFC Deadlock	Section 7.4	Cyclic adversarial traffic	observed/reported, watchdog events
ECMP Imbalance	Section 8.1	MMR, JFI per QP count	dimensionless ratios
DLB Efficacy	Section 8.2	Throughput delta vs. ECMP	%, out-of-order rate
Spray Efficacy	Section 8.3	JFI, retransmission rate	dimensionless, retx/sec
AllReduce BusBW	Section 9.1	CCL benchmark	Gbps per accelerator
AlltoAll JCT	Section 9.2	CCL benchmark	seconds per iteration

AllGather BusBW	Section 9.3	CCL benchmark	Gbps per accelerator
Synthetic JCT Ratio	Section 10.1	Measured / Roofline	dimensionless
MLPerf JCT	Section 10.2	Time-to-train	minutes, tokens/sec
Multi-Tenant Impact	Section 10.3	Contention / Baseline JCT	interference factor
Scale Limit	Section 11.1	Max N with JCT Ratio characterized	accelerator count
Failover Time	Section 11.2	Loss duration on link fail	us
24h Stability	Section 12.1	JCT Ratio std deviation	dimensionless
UET Throughput (RUD)	Section 6.1	Binary search per transport service	Gbps, % line rate
UET First- Packet Latency	Section 6.2	PDC establish + first data	us
UET Spray Efficacy	Section 6.3	JFI/MMR under RUD spray	dimensionless, 000 rate
UET PFC-Free Loss Rate	Section 6.4	Incast without PFC enabled	%, retx overhead
LLR Retry Latency	Section 6.5	Per-hop error recovery time	nanoseconds
Packet Trimming Savings	Section 6.5	BW saved during congestion	% bandwidth
CBFC vs PFC HOL Blocking	Section 6.5	Head-of-line blocking duration	us
UET	Section	AllReduce/AlltoAll	Gbps per

Collective BusBW	6.6	over UET	accelerator
PDC Establishment Rate	Section 6.7	Sustained PDC creation rate	PDCs/second
Max Concurrent PDCs	Section 6.7	Scale limit per NIC	count

Table 13: KPI-to-Test Mapping Summary

Appendix B. Indicative Reference Values (Non-Normative)

This appendix provides indicative reference values for the KPIs defined in Section 4, reflecting current industry observations for distributed AI training workloads as of 2025-2026. These values are NON-NORMATIVE and do not constitute benchmarking acceptance criteria or performance requirements. Per the BMWG charter, the definition of acceptance criteria or performance requirements is explicitly outside the scope of this Working Group. Implementers may use these values as contextual references when interpreting results; they MUST NOT be used as pass/fail thresholds in vendor evaluations. Deployment-specific targets will vary by topology, accelerator architecture, collective library, and operator requirements.

KPI	Indicative Reference
JCT Ratio	<= 1.05 (<= 1.15 acceptable)
BusBW	>= 90% of NIC line rate (intra-pod)
Aggregate Throughput	>= 95% of bisection BW
Packet Drop Rate	0 ppm (lossless)

Table 14: Indicative Reference Values for Distributed AI Training Fabrics (Non- Normative)

Appendix C. ASIC Feature Categories (Informational)

This appendix identifies ASIC feature categories relevant to AI fabric performance. Implementers document which categories are present and enabled on the DUT. Specific vendor names are intentionally omitted.

Feature Category	Sub-types	Relevance to AI Fabric	What to Report
Aggregate Switching BW	ASIC-level capacity	Cluster scale, bisection BW	Total Tbps; per-port speed (400/800GbE)
Buffer Architecture	Shared, VOQ, Cut-through	Microburst absorption, PFC behavior, lossless operation	Buffer type; total bytes; shared vs. dedicated split; per-port/queue allocation
Packet Distribution	Per-flow, Per-packet, Flowlet	ECMP load balancing quality and reordering risk	Supported granularities; in-fabric reorder buffer (yes/no)
Congestion Control	ECN marking, PFC, DCQCN	DCQCN convergence and lossless behavior	ECN granularity (port/queue/VOQ); PFC priorities; DCQCN parameter range
Adaptive Routing	Flowlet, ECMP, Spray, Topology-aware	Load balancing quality under collective patterns	Algorithm type; flowlet gap timer range; topology-aware support
Telemetry	Per-port, Per-queue, Per-flow	Required for KPI measurement during benchmarking	Monitoring granularity; streaming interval; INT support
Cluster Scale Support	2-tier, 3-tier	Applicable topology scales	Max cluster size per topology; ASIC count

Table 15: ASIC Feature Categories

All values are reported based on vendor documentation or measured capability. Additional DUT capabilities affecting benchmark results are also documented.

Appendix D. RoCEv2 Test Frame Format

Offset	Field	Size	Value / Description
00	Ethernet Dst MAC	6B	DUT next-hop MAC
06	Ethernet Src MAC	6B	Test equipment MAC
12	EtherType / TPID	2B	0x0800 (IPv4) when untagged; 0x8100 (Tag Protocol Identifier — TPID) when 802.1Q-tagged
14	802.1Q Tag (optional)	4B	When tagged: Tag Control Information (TCI: Priority Code Point (PCP)=3 for RoCEv2 priority, VLAN Identifier (VID)) followed by inner EtherType 0x0800. Omit this row entirely when untagged and shift subsequent offsets back by 4B
18	IPv4 Header	20B	DSCP=26 (AF31, Assured Forwarding class 31), ECN=ECT(0) (ECN-Capable Transport), Proto=17 (UDP)
38	UDP Header	8B	DstPort=4791 (RoCEv2), SrcPort=var
46	BTH (Base Transport Header)	12B	OpCode, DstQP, PSN, P_Key
58	RDMA Extended Transport Header (RETH; if Write)	16B	Virtual Address (VA), R_Key, Direct Memory Access (DMA) Length
74	Payload	var	Test data (incrementing octets)
var	ICRC	4B	Invariant CRC

var+4	FCS	4B	Ethernet Frame Check Sequence	
+-----+	+-----+	+-----+	+-----+	+-----+

Table 16: RoCEv2 Test Frame Format

Appendix E. UET (Ultra Ethernet Transport) Frame Format

UET runs over UDP/IP using IANA-assigned destination port 4793.

Offset	Field	Size	Value / Description
00	Ethernet Dst MAC	6B	DUT next-hop MAC
06	Ethernet Src MAC	6B	Test equipment MAC
12	EtherType / TPID	2B	0x0800 (IPv4) when untagged; 0x8100 (TPID) when 802.1Q-tagged
14	802.1Q Tag (optional)	4B	When tagged: TCI (PCP=3 for UET priority class, VID) followed by inner EtherType 0x0800. Omit this row entirely when untagged and shift subsequent offsets back by 4B
18	IPv4 Header	20B	DSCP=26 (AF31), ECN=ECT(0), Proto=17 (UDP)
38	UDP Header	8B	DstPort=4793 (UET), SrcPort=entropy
46	UET Common Header	16B	Version, OpCode, PDC ID, PSN, Entropy Value, Flags
62	SES Header (Semantic)	var	Operation-specific (Write/Send/etc.)
var	PDS Header (Pkt Delivery)	var	Sequence, Credit, Ack fields
var	CMS Header (Cong. Mgmt)	var	ECN feedback, rate signals
var	Payload	var	Application data
var	ICRC	4B	Invariant CRC
var+4	FCS	4B	Ethernet Frame Check Sequence

Table 17: UET Frame Format

E.1. Key Differences from RoCEv2

Field	RoCEv2 Value	UET Value	Notes
UDP Dst Port	4791	4793	IANA-assigned for each protocol
Transport Endpoint	QP Number (24b)	PDC ID (variable)	Connectionless in UET
Sequence Number	PSN (24b)	PSN (extended)	Larger range for RUD 000 tolerance
Congestion Signal	ECN bits only	ECN + CMS sub-header	Sender + receiver signals in UET
Entropy Source	UDP src port	Explicit entropy field	Deterministic spray in UET
Ordering Guarantee	Always in-order (RC)	Per-service (ROD/RUD)	RUD allows 000 delivery
Min Header Overhead	~74B (Write)	~78B (est. Write)	Slight increase for sub-layer headers

Table 18: RoCEv2 vs. UET Comparison

1. ***UDP Destination Port:** UET uses port 4793 vs. RoCEv2 port 4791.
2. ***Entropy Value:** Explicit entropy field for ECMP path selection. Test equipment varies this field to achieve uniform path distribution.
3. ***Transport Service Indicator:** Header encodes transport service (ROD/RUD/RUDI/UUD). Tests set this to match the service being benchmarked.
4. ***PDC Identifier:** Connectionless PDC ID replaces RoCEv2's Destination QP. Test equipment tracks PDC lifecycle for accurate measurement.
5. ***Layered Sub-Headers:** UET uses four sub-layers (SES, PDS, CMS, TSS) with variable-length headers. Implementations MUST follow [UEC-1.0] Section 4 for wire format details.

6. *Optional Link Layer Headers:* When LLR, Packet Trimming, or PRI features are enabled, additional link-layer framing may be present. Test equipment is configured to recognize and parse these.

Acknowledgments

This work has benefited from the discussions that occurred during the joint IPPM and BMWG meeting and on the BMWG mailing list. Thanks to Carsten Rossenhoewel, Mohamed Boucadair, and Sowjanya Reddy for valuable review and comments.

Authors' Addresses

Fernando Calabria
Cisco
United States
Email: fcalabri@cisco.com

Carlos Pignataro
Blue Fern Consulting
United States
Email: carlos@bluefern.consulting

Qin Wu
Huawei
China
Email: bill.wu@huawei.com

Giuseppe Fioccola
Huawei
Italy
Email: giuseppe.fioccola@huawei.com

Sowjanya Reddy
Apple
United States
Email: sowjredd@gmail.com