spcl.ethz.ch

@spcl
@spcl_eth

ETH zürich

T. HOEFLER

# Ultra Ethernet for Next-Generation AI and HPC Workloads

Invited Talk at Hot Interconnects, August 2025

## Ultra Ethernet's Design Principles and Architectural Innovations


TORSTEN HOEFLER, ETH Zurich, Switzerland & Microsoft, USA
KAREN SCHRAMM, Broadcom, USA
ERIC SPADA, Broadcom, USA
KEITH UNDERWOOD, Hewlett Packard Enterprise, USA
CEDELL ALEXANDER, Broadcom, USA
BOB ALVERSON, Hewlett Packard Enterprise, USA
PAUL BOTTORFF, Hewlett Packard Enterprise, USA
ADRIAN CAULFIELD, OpenAI, USA
MARK HANDLEY, OpenAI, USA
CATHY HUANG, Intel, USA
COSTIN RAICIU, Broadcom, USA
ABDUL KABBANI, Microsoft, USA
EUGENE OPSASNICK, Broadcom, USA
RONG PAN, AMD, USA
ADEE RAN, Cisco, USA
RIP SOHAN, AMD, USA


https://arxiv.org/abs/2508.08906

arXiv:2508.08906v1 [cs.NI] 12 Aug 2025

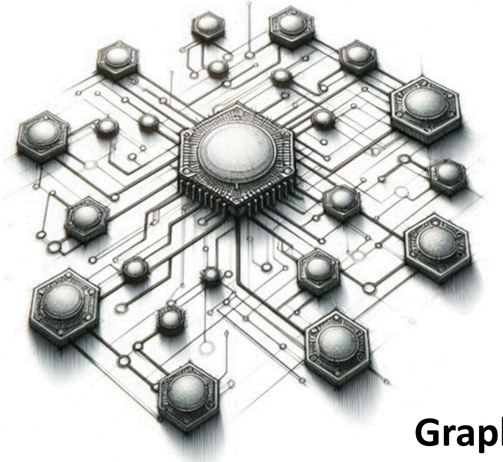

The recently released Ultra Ethernet (UE) 1.0 specification defines a transformative High-Performance Ethernet standard for future Artificial Intelligence (AI) and High-Performance Computing (HPC) systems. This paper, written by the specification's authors, provides a high-level overview of UE's design, offering crucial motivations and scientific context to understand its innovations. While UE introduces advancements across the entire Ethernet stack, its standout contribution is the novel Ultra Ethernet Transport (UET), a potentially fully hardware-accelerated protocol engineered for reliable, fast, and efficient communication in extreme-scale systems. Unlike InfiniBand, the last major standardization effort in high-performance networking over two decades ago, UE leverages the expansive Ethernet ecosystem and the 1,000x gains in computational efficiency per moved bit to deliver a new era of high-performance networking.


### 1 Introduction

Ultra Ethernet (UE) standardizes a new protocol to support high-performance Artificial Intelligence (AI) and High-Performance Computing (HPC) networking over Ethernet. This paper, written by UE's authors, supplements the full specification by highlighting historical and innovative technical aspects of our nearly 2.5-year journey. It is designed to be approachable to a general audience and, thus, abstracts many details while using intuitive wording and explanations. The final authority for questions regarding UE is the full 562-page specification [35].


https://arxiv.org/abs/2508.08906

# AI/HPC Network workload classes



GRAPH 500

Graph Computations

High

Latency Sensitivity

Weather/Climate Simulations

...

Strong scaling HPC workloads

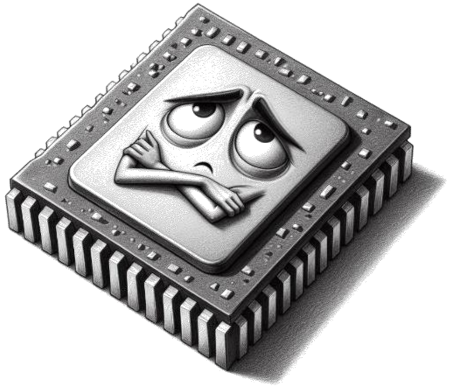
LLM inference

Low — Bandwidth Sensitivity (measure in flop/byte) — High

Bandwidth is simply a matter of area/cost

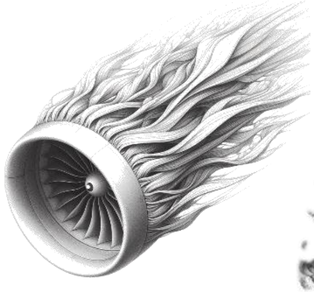
Very hard to measure

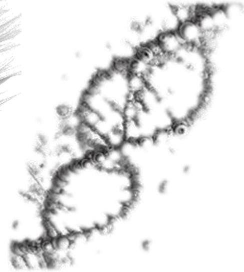
Very hard to improve
(Speed of Light)



Independent jobs
(not really large-scale HPC)

(but may be a major business)





...

Weak scaling HPC workloads

LLM/AI Training

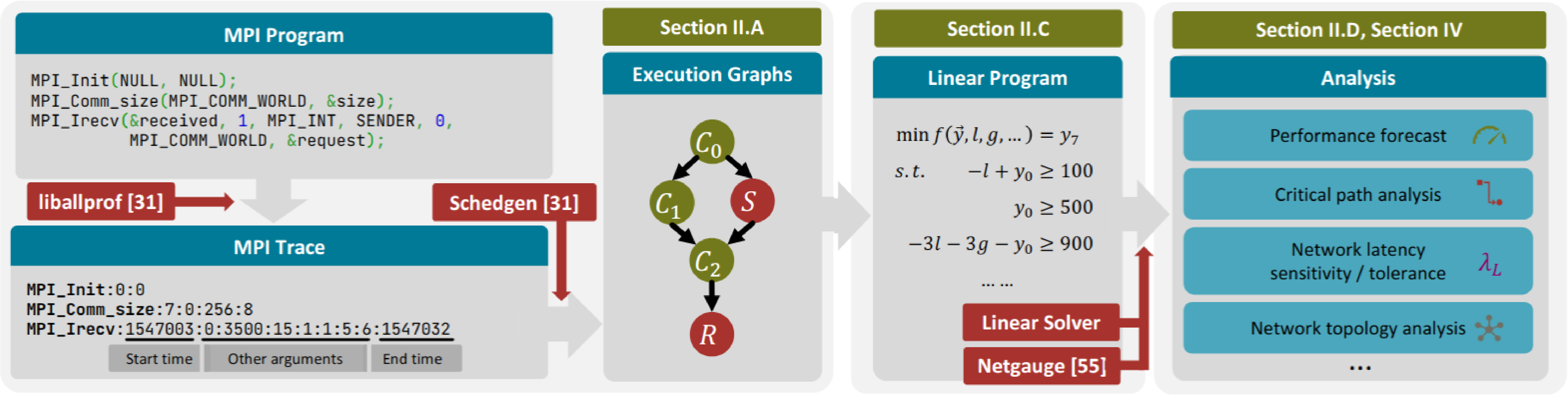
Low

# Measuring the Latency Sensitivity of Real Applications

**How can we determine the latency sensitivity of an application?**

1. Large messages are less latency sensitive than small messages
2. Pipelined/overlapped small messages are less latency sensitive than small messages on the critical path

**→ We need to analyze the whole execution DAG of a parallel (MPI, NCCL) application – message depth!**




S. Chen et al.: "LLAMP: Assessing Network Latency Tolerance of HPC Applications with Linear Programming", arXiv 2404.14193, SC24 Best Paper

# Latency sensitivity of applications varies widely!




## Latency Sensitivity Can be Accurately Measured Using Linear Programming

- <1% sensitivity **varies by 32x** between MILC and ICON -

## Watch for ATLAHS traces and simulator tool-chain at SC25!

https://arxiv.org/abs/2505.08936







S. Chen et al.: “LLAMP: Assessing Network Latency Tolerance of HPC Applications with Linear Programming”, arXiv 2404.14193, SC24 Best Paper

COVER FEATURE TECHNOLOGY PREDICTIONS



# The Convergence of

# Computing Networks

**Torsten Hoefler,** ETH Zurich
**Ariel Hendel,** Scala Computing
**Duncan Roweth,** Hewlett Packard Enterprise

*We discuss the differences and commonalities between network technologies used in supercomputers and data centers and outline a path to convergence at multiple layers. We predict that emerging smart networking solutions will accelerate that convergence.*

- **Design and Deployment**
  - One-off vs. incremental
  - ➢ Proprietary networks vs. Ethernet
  - ✓ AI supercomputers in the cloud
- **Operations philosophy**
  - Run-to-completion jobs vs. high-reliability services
  - ➢ Checkpoint/restart vs. replicated instances
  - ✓ Most will be AI-driven to serve LLMs
- **Protocol stacks and layers**
  - Proprietary vs. task-adapted flow control
  - ➢ Simple protocols vs. multi-traffic protocols
  - ➢ Lossless vs. lossy
- **Utilization and applications**
  - High peak low noise vs. low peak high noise
  - ➢ High bandwidth low latency vs. normal bandwidth high latency
  - ✓ AI demands highest bandwidths and reasonable latency

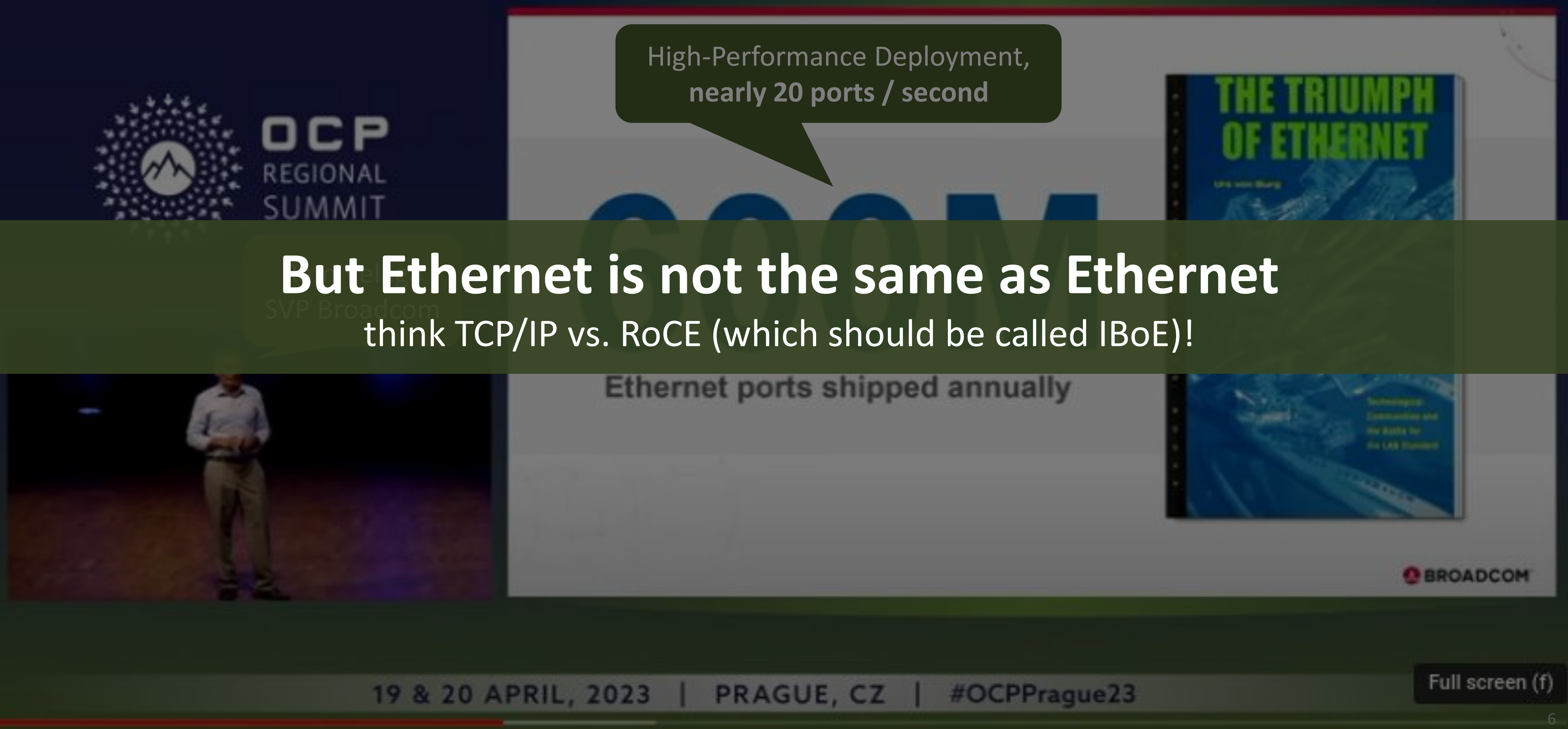
**The network is the cardiovascular system of the datacenter**

**Convergence must happen around <u>Ethernet</u> !**

IEEE Computer, June 2022 (10.1109/MC.2022.3158437)

# The Ethernet Ecosystem – Is the right one!

High-Performance Deployment,
**nearly 20 ports / second**




**But Ethernet is not the same as Ethernet**

think TCP/IP vs. RoCE (which should be called IBoE)!

# Converging our HPC Networking Mess into a Unified Ethernet Standard

COVER FEATURE **TECHNOLOGY PREDICTIONS**


## Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

**Torsten Hoefler**, ETH Zürich
**Duncan Roweth, Keith Underwood, and Robert Alverson,** Hewlett Packard Enterprise
**Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu,** Broadcom
**Siyuan Shen,** ETH Zürich
**Moray McLaren,** Google
**Abdul Kabbani and Steve Scott,** Microsoft

*Remote direct memory access (RDMA) over converged Ethernet (RoCE) was an attempt to adopt modern RDMA features into existing Ethernet installations. We revisit RoCE's design points and conclude that several of its shortcomings must be addressed to fulfill the demands of hyperscale data centers.*

IEEE Computer, June 2023

Ultra Ethernet Consortium

Founding Members

AMD, ARISTA, BROADCOM, CISCO, EVIDEN an atos business, Hewlett Packard Enterprise, intel, Meta, Microsoft

Ultra Ethernet Consortium

**Ultra Ethernet™**
**Specification v1.0**
**June 11, 2025**

https://ultraethernet.org/uec-1-0-spec

**Ultra Ethernet's Design Principles and Architectural Innovations**

TORSTEN HOEFLER, ETH Zurich, Switzerland & Microsoft, USA
KAREN SCHRAMM, Broadcom, USA
ERIC SPADA, Broadcom, USA
KEITH UNDERWOOD, Hewlett Packard Enterprise, USA
CEDELL ALEXANDER, Broadcom, USA
BOB ALVERSON, Hewlett Packard Enterprise, USA
PAUL BOTTORFF, Hewlett Packard Enterprise, USA
ADRIAN CAULFIELD, OpenAI, USA
MARK HANDLEY, OpenAI, USA
CATHY HUANG, Intel, USA
COSTIN RAICIU, Broadcom, USA
ABDUL KABBANI, Microsoft, USA
EUGENE OPSASNICK, Broadcom, USA
RONG PAN, AMD, USA
ADEE RAN, Cisco, USA
RIP SOHAN, AMD, USA

.08906v1 [cs.NI] 12 Aug 2025

The recently released Ultra Ethernet (UE) 1.0 specification defines a transformative High-Performance Ethernet standard for future Artificial Intelligence (AI) and High-Performance Computing (HPC) systems. This paper, written by the specification's authors, provides a high-level overview of UE's design, offering crucial motivations and scientific context to understand its innovations. While UE introduces advancements across the entire Ethernet stack, its standout contribution is the novel Ultra Ethernet Transport (UET), a potentially fully hard ... -scale systems. ... r two decades ago, UE leverages the expansive Ethernet ecosystem and the 1,000x gains in computational efficiency per moved bit to deliver a new era of high-performance networking.

https://arxiv.org/abs/2508.08906

# Ecosystem is quicky growing

Ultra Ethernet Consortium
https://arxiv.org/abs/2508.08906

**Today 10 steering companies, 17 general member companies, 51 contributor members**

Chair's view of the Transport WG Meeting in March'24 (60+ members on site, 1,300+ total)

# Ultra Ethernet Members – Join our Journey!

Ultra Ethernet Consortium
https://arxiv.org/abs/2508.08906

*not all members listed

**75+ member companies**
**1,300+ individual participants**

https://arxiv.org/abs/2508.08906

# Modernizing RDMA for HPC and AI

| Classic RDMA | Ultra Ethernet Consortium |
|---|---|
| **Lossless (PFC or CBFC) operation** | **Lossy (& lossless) operation** |
| **In-order transport and delivery**<br>Inefficient go-back-n | **Out-of-order data and message delivery**<br>(Un)Reliable (Un)Ordered - ROD, RUD/RUDI, and UUD |
| **Proprietary congestion control (e.g., DCQCN)** | **Open, configurable, and flexible CC** |
| **Single-path routing**<br>No load balancing and "link polarization" | **Per-packet multipathing and load balancing**<br>Including (close-to) zero state REPS |
| **Large state per queue pair**<br>kb NIC memory per peer | **Connection-less API**<br>Ephemeral zero-RTT reliability state |
| **Security added at higher layers**<br>IPSec, $N^2$ contexts, known attacks | **Built-in security**<br>Cluster-wide keying, zero state replay protection |

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

**sRDMA – Efficient NIC-based Authentication and Encryption for Remote Direct Memory Access**

Konstantin Taranov, Benjamin Rothenberger, Adrian Perrig, and Torsten Hoefler, *ETH Zurich*

https://www.usenix.org/conference/atc20/presentation/taranov

usenix THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

**ReDMArk: Bypassing RDMA Security Mechanisms**

Benjamin Rothenberger, Konstantin Taranov, Adrian Perrig, and Torsten Hoefler, *ETH Zurich*

https://www.usenix.org/conference/usenixsecurity21/presentation/rothenberger

MPI

Open SHMEM

*CCL

# Remote Memory Access and Send/Receive with libfabric
## Matching, Connectionless, Lightweight

MPI *CCL Open SHMEM

**Application Layer** (*CCL, MPI, OpenSHMEM)

libfabric

**Transport Layer**

- **Semantics Sublayer (SES)** — UE addressing, Send/Recv, Deferrable Send, RMA Read/Write
- **Packet Delivery Sublayer (PDS)** — Zero-RTT PD Context Build, Req, Resp, Ctrl Packets, Flexible Loss Detection
- **Congestion Mgmt Sublayer (CMS)** — Window-based Sender + opt. Receiver Based CC components, Flexible LB
- **Transport Security Sublayer (TSS)** — O(1) key state per FEP, KDF for rekeying and per-client keys, replay protection

**Network Layer (IP)**

**Data Link Layer**

Credit-Based FC | Link Level Retry

Media Access Control (MAC)

**Ethernet Physical Layer**

UE 100G / lane | UE 200G / lane

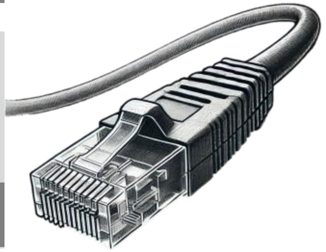
Physical Medium Attachment / Dependent (PMD/PMA)

| IP | Opt. UDP | Opt. TSS | PDS | SES | UE Payload | Opt. TSS |
|---|---|---|---|---|---|---|
| IPv4 (20B + opt) or IPv6 (40B) | (8B) | 12-16B | [if no UDP – 4B entropy] (2B + 12-20B ROD/RUD or 8B RUDI, or 4B UUD + opt. 8B CRC) | (44B standard, 32B up to 8kiB msgs, 20B min non-matching) | | ICV (16B) |

Port 4793

| preamble | sfd | dest addr | src addr | type | data | fcs |
|---|---|---|---|---|---|---|
| 7B | 1B | 6B | 6B | 2B | >46B | 4B |

**IEEE 802.3 – with optional Link Level Retry and Credit Based FC Extensions**



**Follows developing Ethernet specifications**

- **802.3 100G per lane signaling today**
- **802.3 200G per lane signaling upcoming**

Note the **Inter Frame Gap (IFG)** for efficiency estimates

https://arxiv.org/abs/2508.08906

## Application Layer (*CCL, MPI, OpenSHMEM)
libfabric

## Transport Layer
**Semantics Sublayer (SES)**

**Packet Delivery Sublayer (PDS)**

**Congestion Mgmt Sublayer (CMS)**

**Transport Security Sublayer (TSS)**

## Network Layer (IP)

## Data Link Layer
Credit-Based FC | Link Level Retry

Media Access Control (MAC)

## Ethernet Physical Layer
UE 100G / lane | UE 200G / lane

Physical Medium Attachment / Dependent (PMD/PMA)

MPI *CCL Open SHMEM

# Remote Memory Access and Send/Receive with libfabric
# Matching, Connectionless, Lightweight

- UE addressing, Send/Recv, Deferrable Send, RMA Read/Write
- Zero-RTT PD Context Build, Req, Resp, Ctrl Packets, Flexible Loss Detection
- Window-based Sender + opt. Receiver Based CC components, Flexible LB
- O(1) key state per FEP, KDF for rekeying and per-client keys, replay protection

| IP | Opt. UDP | Opt. TSS | PDS | SES | UE Payload | Opt. TSS |
|---|---|---|---|---|---|---|
| IPv4 (20B + opt) or IPv6 (40B) | (8B) | 12-16B | [if no UDP – 4B entropy] (2B + 12-20B ROD/RUD or 8B RUDI, or 4B UUD + opt. 8B CRC) | (44B standard, 32B up to 8kiB msgs, 20B min non-matching) | | ICV (16B) |

Port 4793

| preamble | sfd | dest addr | src addr | type | data | fcs |
|---|---|---|---|---|---|---|
| 7B | 1B | 6B | 6B | 2B | >46B | 4B |

**IEEE 802.3 – with optional Link Level Retry and Credit Based FC Extensions**

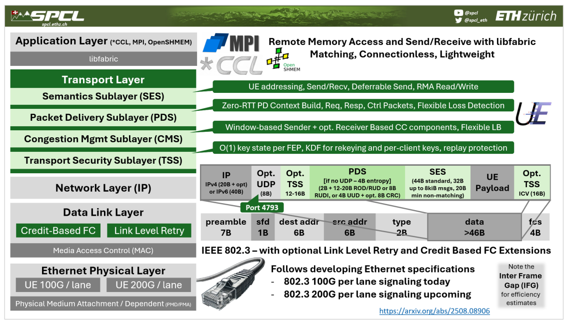
**Follows developing Ethernet specifications**
- **802.3 100G per lane signaling today**
- **802.3 200G per lane signaling upcoming**

Note the **Inter Frame Gap (IFG)** for efficiency estimates

https://arxiv.org/abs/2508.08906

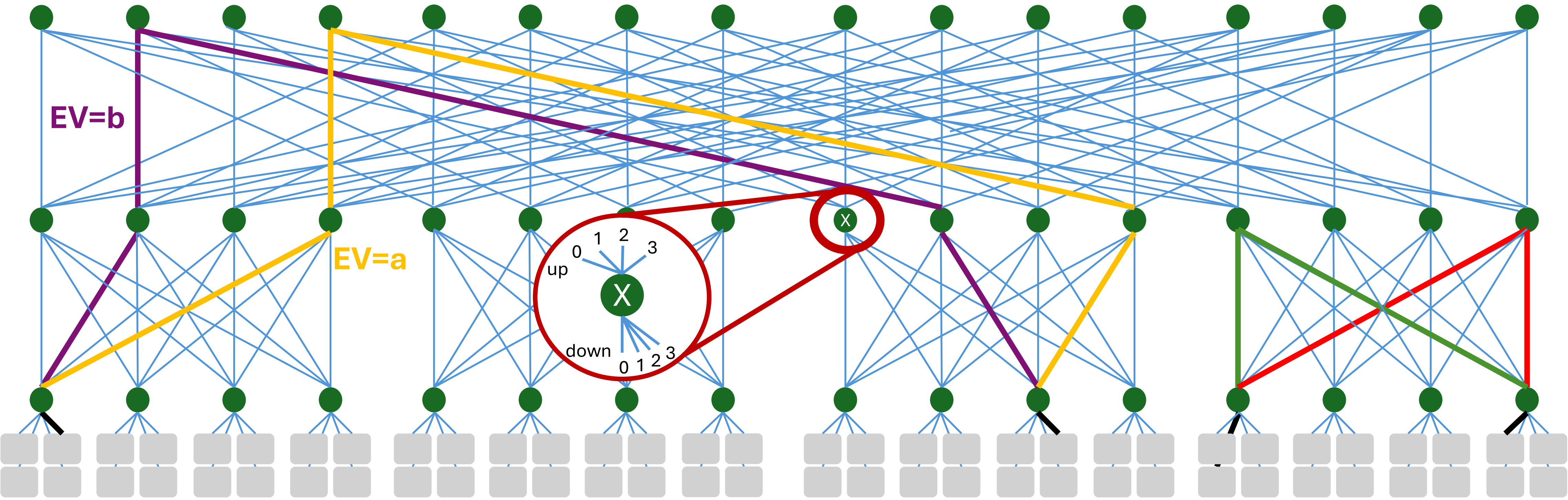
# Ultra Ethernet Transport - Load Balancing Philosophy

Ultra Ethernet Consortium




Based on standard Ethernet Equal Cost Multi Pathing (ECMP)

- Uses an "entropy value" (EV) to select from a set of output ports (encoded as UDP source port)
- Each EV selects a path (not necessarily unique)
- Same EV means same path (without failures)



## Remote Memory Access and Send/Receive with libfabric
## Matching, Connectionless, Lightweight

MPI *CCL OpenSHMEM

**Application Layer** (*CCL, MPI, OpenSHMEM)

libfabric

**Transport Layer**

- **Semantics Sublayer (SES)** — UE addressing, Send/Recv, Deferrable Send, RMA Read/Write
- **Packet Delivery Sublayer (PDS)** — Zero-RTT PD Context Build, Req, Resp, Ctrl Packets, Flexible Loss Detection
- **Congestion Mgmt Sublayer (CMS)** — Window-based Sender + opt. Receiver Based CC components, Flexible LB
- **Transport Security Sublayer (TSS)** — O(1) key state per FEP, KDF for rekeying and per-client keys, replay protection

UE

**Network Layer (IP)**

| IP | Opt. UDP | Opt. TSS | PDS | SES | UE Payload | Opt. TSS |
|---|---|---|---|---|---|---|
| IPv4 (20B + opt) or IPv6 (40B) | (8B) | 12-16B | [if no UDP – 4B entropy] (2B + 12-20B ROD/RUD or 8B RUDI, or 4B UUD + opt. 8B CRC) | (44B standard, 32B up to 8kiB msgs, 20B min non-matching) | | ICV (16B) |

Port 4793

**Data Link Layer**

Credit-Based FC | Link Level Retry

Media Access Control (MAC)

| preamble | sfd | dest addr | src addr | type | data | fcs |
|---|---|---|---|---|---|---|
| 7B | 1B | 6B | 6B | 2B | >46B | 4B |

**IEEE 802.3 – with optional Link Level Retry and Credit Based FC Extensions**

**Ethernet Physical Layer**

UE 100G / lane | UE 200G / lane

Physical Medium Attachment / Dependent (PMD/PMA)

**Follows developing Ethernet specifications**

- **802.3 100G per lane signaling today**
- **802.3 200G per lane signaling upcoming**

Note the **Inter Frame Gap (IFG)** for efficiency estimates

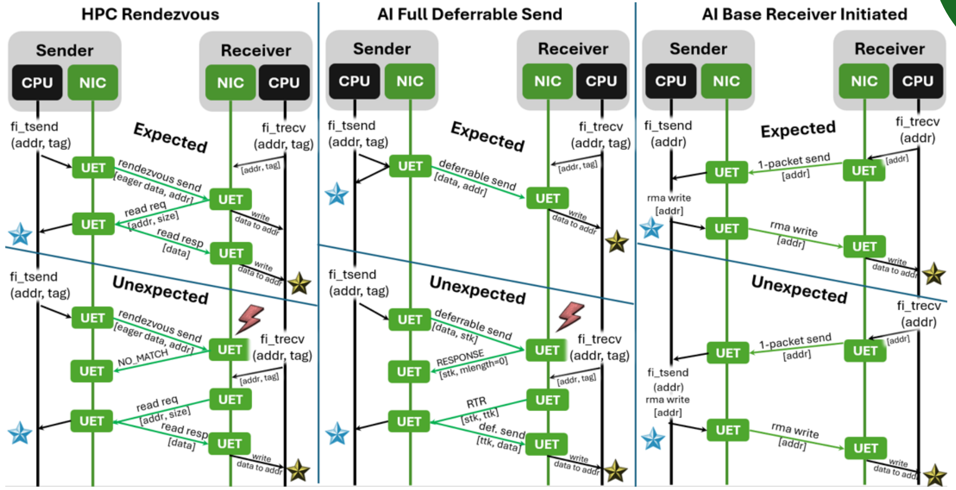
https://arxiv.org/abs/2508.08906

# Transport layer - sublayers

Ultra Ethernet Consortium

**Transport Layer**
- **Semantics Sublayer (SES)**
- **Packet Delivery Sublayer (PDS)**
- **Congestion Mgmt Sublayer (CMS)**
- **Transport Security Sublayer (TSS)**

- Compatible with existing applications (libfabric) – no change!
- RDMA services: Send/Recv + RMA (Write, Read, Atomics)
  - Focus on MPI and *CCL semantics
- Scalable addressing to millions of endpoints
- Optimized extensions:
  - Deferrable Send for optimized HW (aimed at AI)
  - Rendezvous using Send/Read (aimed at HPC)
  - Exact match tags for HW offload of ordering between endpoints using shared receive queues



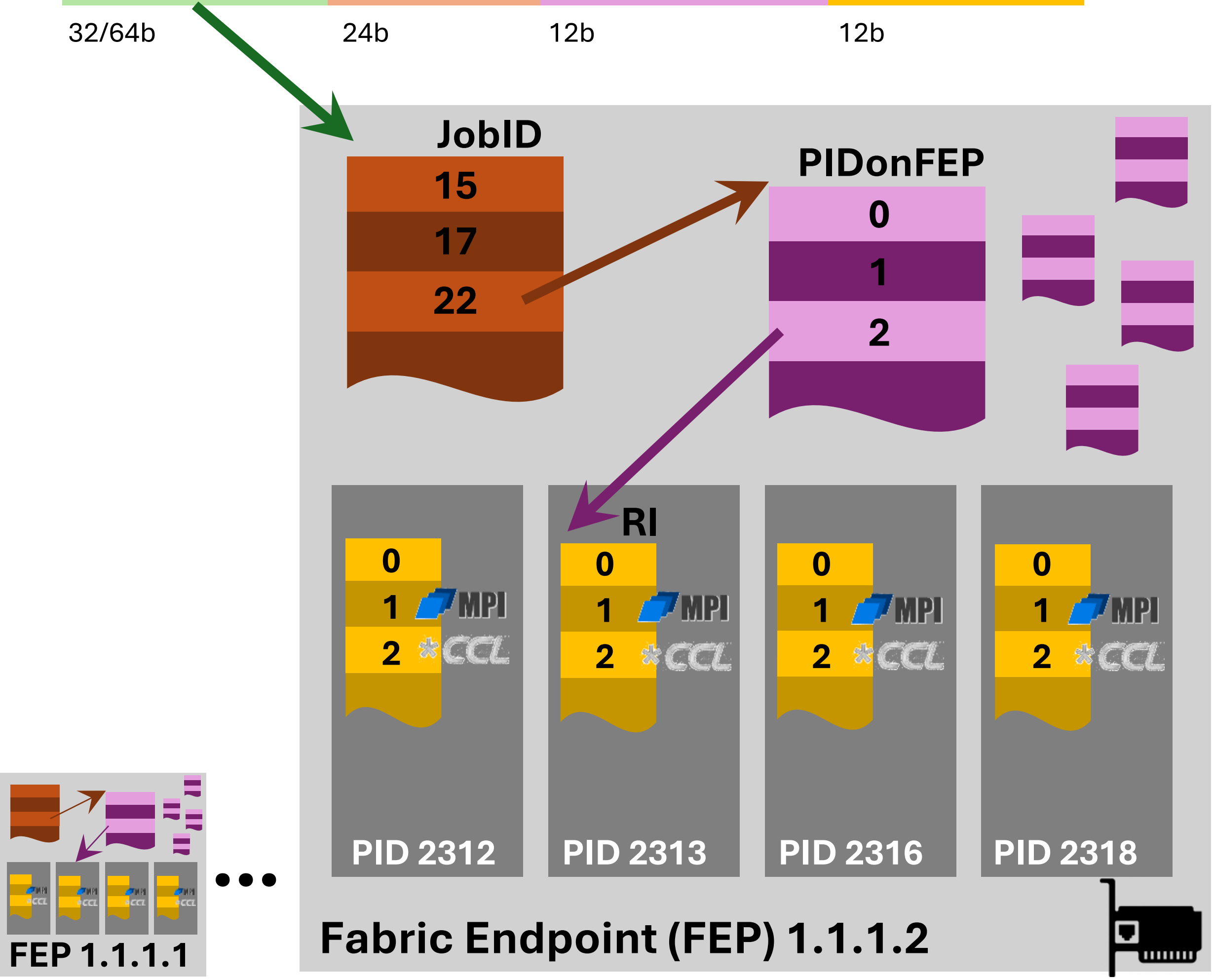

**Use-case optimized communication profiles (AI Base, AI Full, HPC)**

https://arxiv.org/abs/2508.08906

# Ultra Ethernet Transport Scalable Addressing

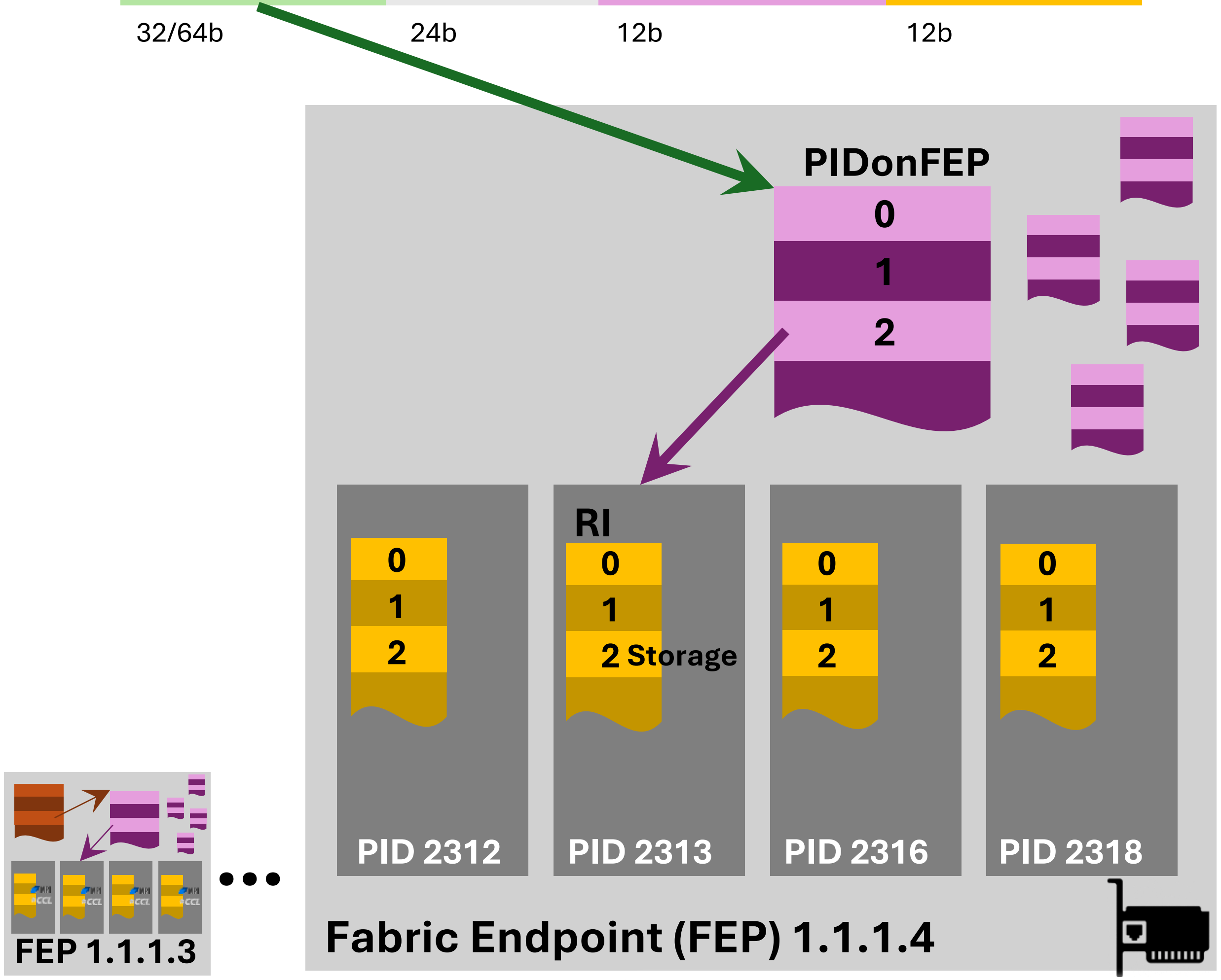
## Relative (parallel job) Addressing

| FA 1.1.1.2 | JobID 22 | PIDonFEP 2 | RI 1 |
|---|---|---|---|
| 32/64b | 24b | 12b | 12b |



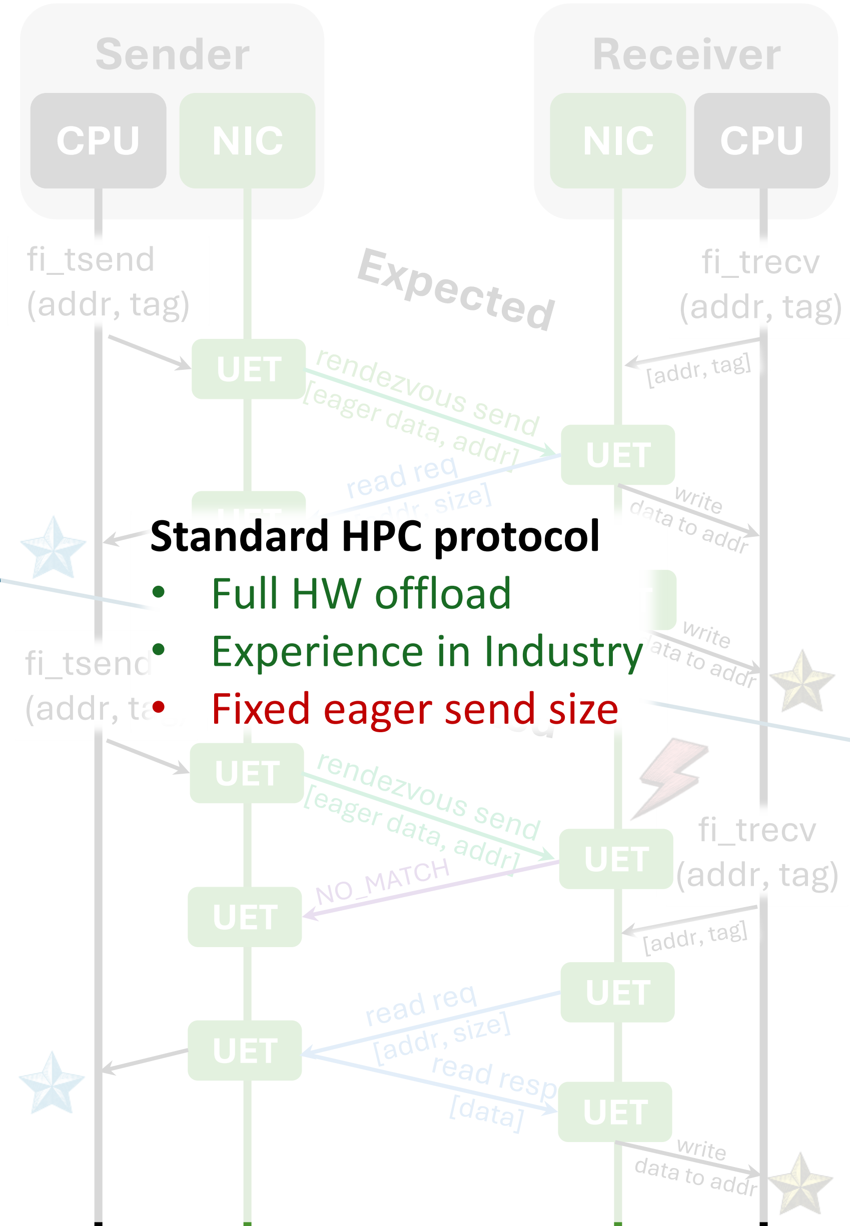

## Absolute (client/server) Addressing

| FA 1.1.1.4 | JobID xx | PIDonFEP 2 | RI 2 |
|---|---|---|---|
| 32/64b | 24b | 12b | 12b |



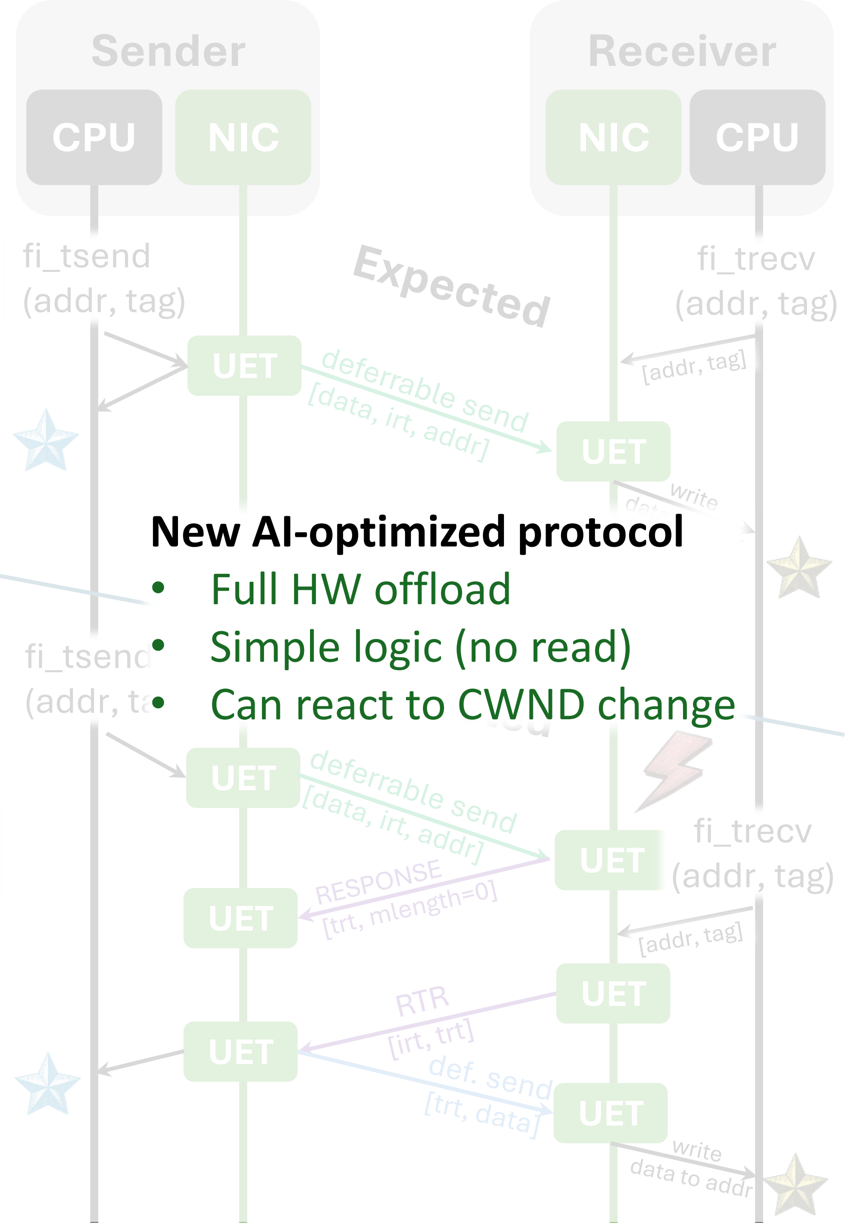
# Ultra Ethernet Transport Large Message Delivery Options and Profiles

## HPC Rendezvous



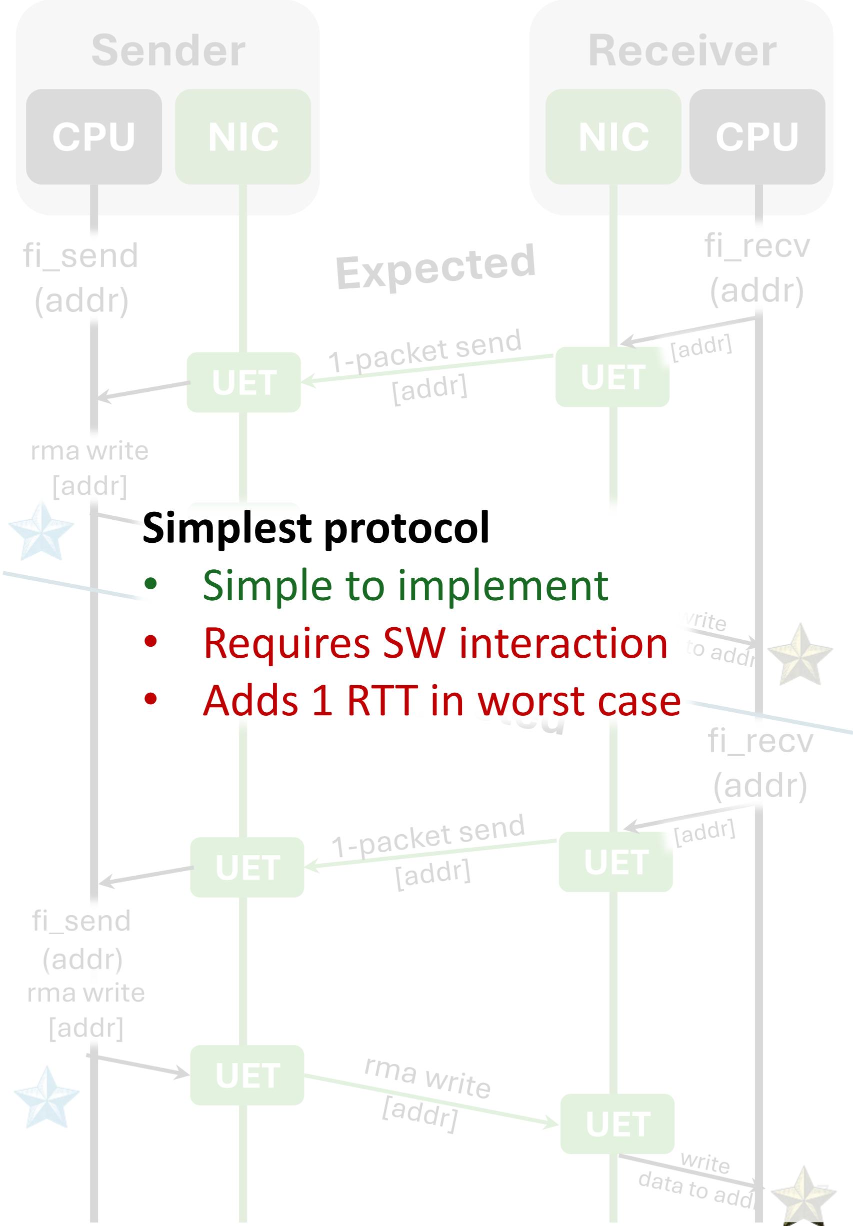

**Standard HPC protocol**
- Full HW offload
- Experience in Industry
- Fixed eager send size

## AI Deferrable Send




**New AI-optimized protocol**
- Full HW offload
- Simple logic (no read)
- Can react to CWND change

## AI Base Receiver Initiated




**Simplest protocol**
- Simple to implement
- Requires SW interaction
- Adds 1 RTT in worst case

# Transport layer - sublayers

**Transport Layer**

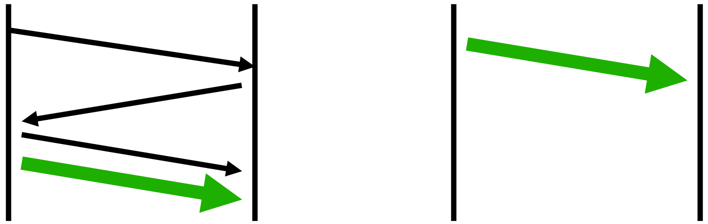
**Semantics Sublayer (SES)**

**Packet Delivery Sublayer (PDS)**

**Congestion Mgmt Sublayer (CMS)**

**Transport Security Sublayer (TSS)**

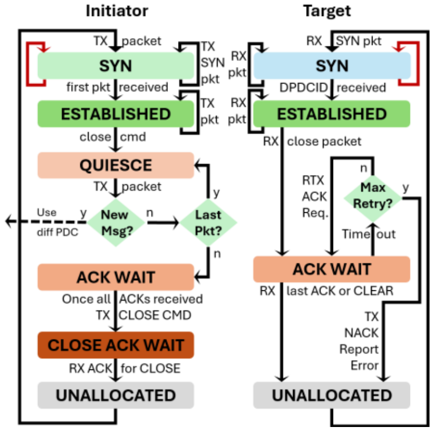
## Zero-RTT Startup

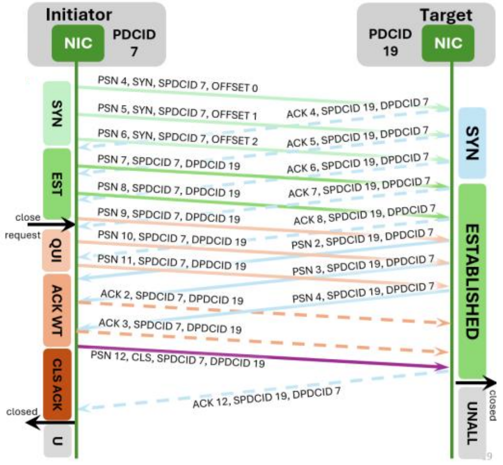


slow
(e.g., TCP)

fast
(UET)

- Dynamic, ephemeral connections
  - Zero start up time, 1-RTT close
- 4 delivery services
  - ROD – Reliable, ordered
  - RUD – Reliable, unordered
  - RUDI – Reliable, unordered, idempotent (Write/Read)
  - UUD – Unreliable, unordered
- Shared receive queues
- Out-of-order packet arrival
- Selective acknowledgement and retransmission for RUD
  - ROD uses Go-Back-N




**Fastest startup, drop state when convenient, rebuild it quickly!**

https://arxiv.org/abs/2508.08906

# Transport layer - sublayers

Ultra Ethernet Consortium

**Transport Layer**

- **Semantics Sublayer (SES)**
- **Packet Delivery Sublayer (PDS)**
- **Congestion Mgmt Sublayer (CMS)**
- **Transport Security Sublayer (TSS)**

- Multipath with congestion avoidance
  - Leveraging ECMP
- Trimming with NACK signal
- Network Signaled CC (NSCC)
  - Window based at sender using RTT and ECN
- Receiver Controlled CC (RCCC)
  - Credit based at receiver

Network Signal Based CC
(Sender-controlled)

- Available in all UE products
- Can be disabled
- Flexible for most deployments

Receiver Controlled CC

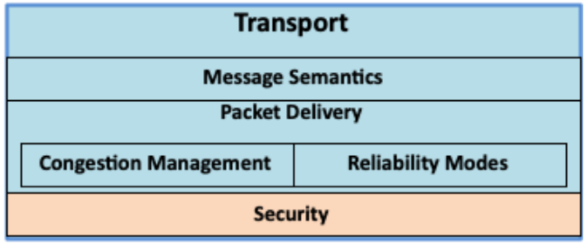
- Available in some UE products
- Receiver hands out credits
- Ideal for incast patterns

**Work together for HPC+AI multi-pathing**

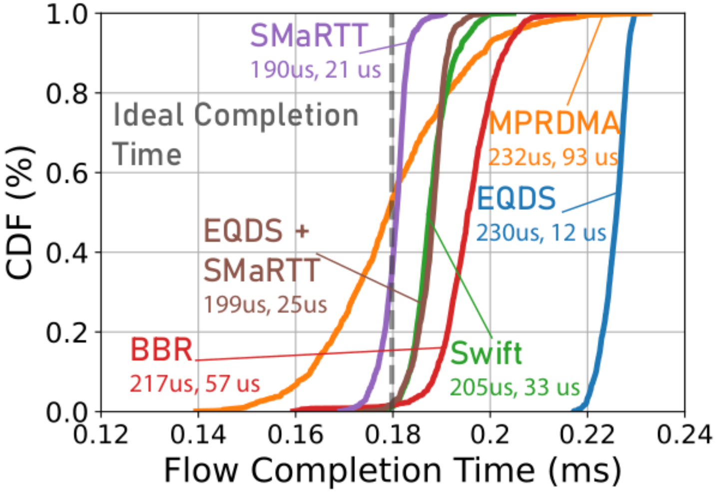
https://arxiv.org/abs/2508.08906

# SMaRTT-REPS enables Modern Packet Spraying

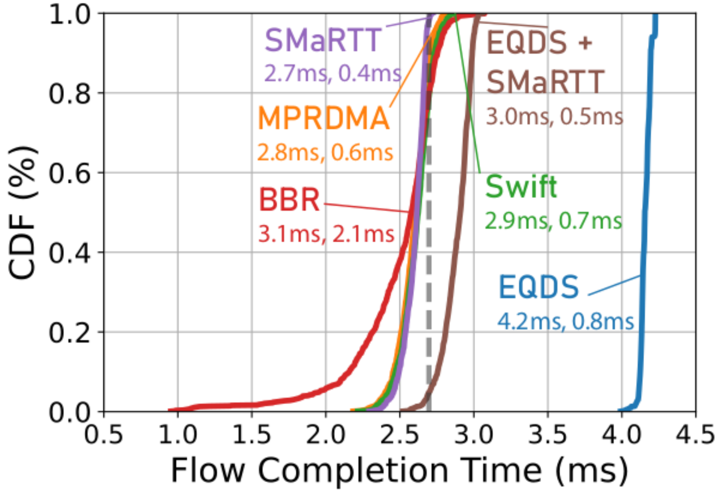
UltraEthernet Consortium



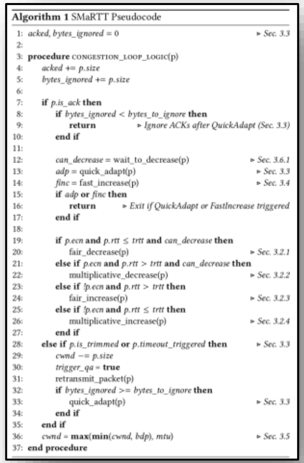

- **"State of the art" (2024), easily configured congestion control mechanisms**




2 MiB Flows




32 MiB Flows

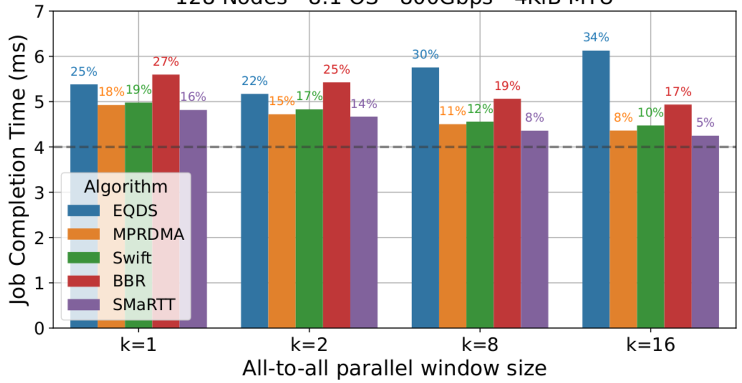


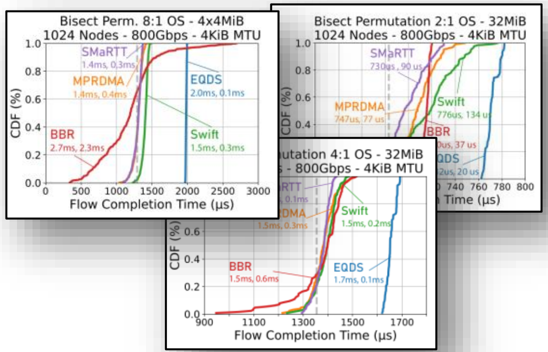
**Algorithm 1** SMaRTT Pseudocode

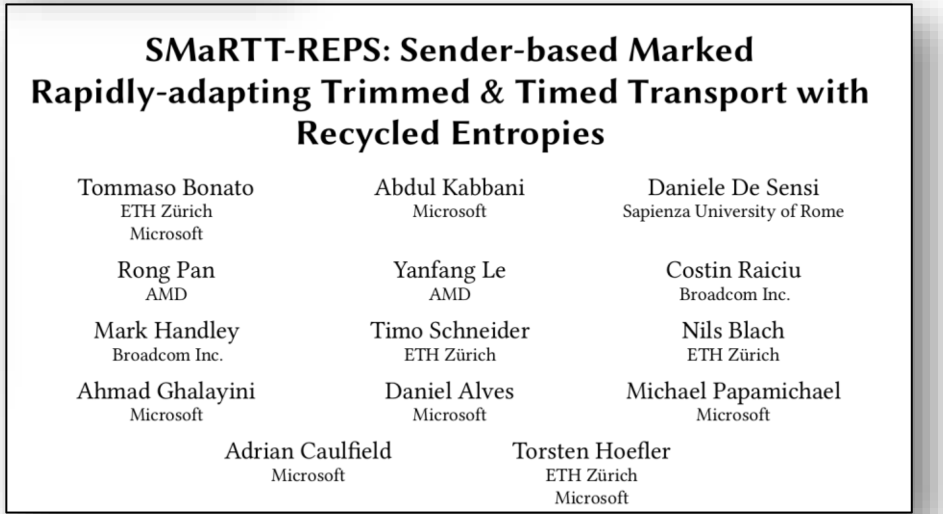
```
acked, bytes_ignored = 0                                        ▷ Sec. 3.3

procedure CONGESTION_LOOP_LOGIC(p)
    acked += p.size
    bytes_ignored += p.size

    if p.is_ack then
        if bytes_ignored < bytes_to_ignore then
            return                     ▷ Ignore ACKs after QuickAdapt (Sec. 3.3)
        end if

        can_decrease = wait_to_decrease(p)                      ▷ Sec. 3.6.1
        adp = quick_adapt(p)                                    ▷ Sec. 3.3
        finc = fast_increase(p)                                 ▷ Sec. 3.4
        if adp or finc then
            return          ▷ Exit if QuickAdapt or FastIncrease triggered
        end if

        if p.ecn and p.rtt ≤ trtt and can_decrease then
            fair_decrease(p)                                    ▷ Sec. 3.2.1
        else if p.ecn and p.rtt > trtt and can_decrease then
            multiplicative_decrease(p)                          ▷ Sec. 3.2.2
        else if !p.ecn and p.rtt > trtt then
            fair_increase(p)                                    ▷ Sec. 3.2.3
        else if !p.ecn and p.rtt ≤ trtt then
            multiplicative_increase(p)                          ▷ Sec. 3.2.4
        end if
    else if p.is_trimmed or p.timeout_triggered then            ▷ Sec. 3.3
        cwnd −= p.size
        trigger_qa = true
        retransmit_packet(p)
        if bytes_ignored >= bytes_to_ignore then
            quick_adapt(p)                                      ▷ Sec. 3.3
        end if
    end if
    cwnd = max(min(cwnd, bdp), mtu)                             ▷ Sec. 3.5
end procedure
```

37 lines simple pseudo-code

**Permutation traffic on 8:1 oversubscribed fat tree**



**SMaRTT-REPS: Sender-based Marked Rapidly-adapting Trimmed & Timed Transport with Recycled Entropies**

Tommaso Bonato, ETH Zürich, Microsoft
Abdul Kabbani, Microsoft
Daniele De Sensi, Sapienza University of Rome
Rong Pan, AMD
Yanfang Le, AMD
Costin Raiciu, Broadcom Inc.
Mark Handley, Broadcom Inc.
Timo Schneider, ETH Zürich
Nils Blach, ETH Zürich
Ahmad Ghalayini, Microsoft
Daniel Alves, Microsoft
Michael Papamichael, Microsoft
Adrian Caulfield, Microsoft
Torsten Hoefler, ETH Zürich, Microsoft

https://arxiv.org/abs/2508.08906



Bonato et al.: SMaRTT-REPS: Sender-based Marked Rapidly-adapting Trimmed & Timed Transport with Recycled Entropies, arXiv 2404.01630

# Transport layer features

Ultra Ethernet Consortium

**Transport Layer**

**Semantics Sublayer (SES)**

**Packet Delivery Sublayer (PDS)**

**Congestion Mgmt Sublayer (CMS)**

**Transport Security Sublayer (TSS)**

- End-to-end AES encryption
- Key derivation for additional security
- Replay protection
- Scalable security domains
- Optional within UET

- **Builds on state of the art of IPSec and PSP – fixes all known attacks on RDMA**
  - AES-GCM, KDFs, IVs, Key Rotation, Anti-Replay
  - Protect data, connection establishment, replay in all scenarios
- **High scalability**
  - Group (re)keying
  - Secure Domains
  - Strong isolation (also wrt. in-network computation)

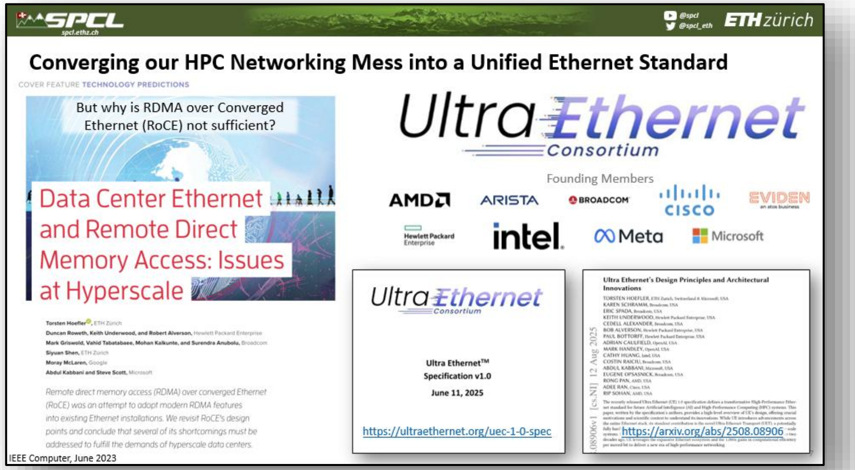
https://arxiv.org/abs/2508.08906

# Key Points and Conclusions

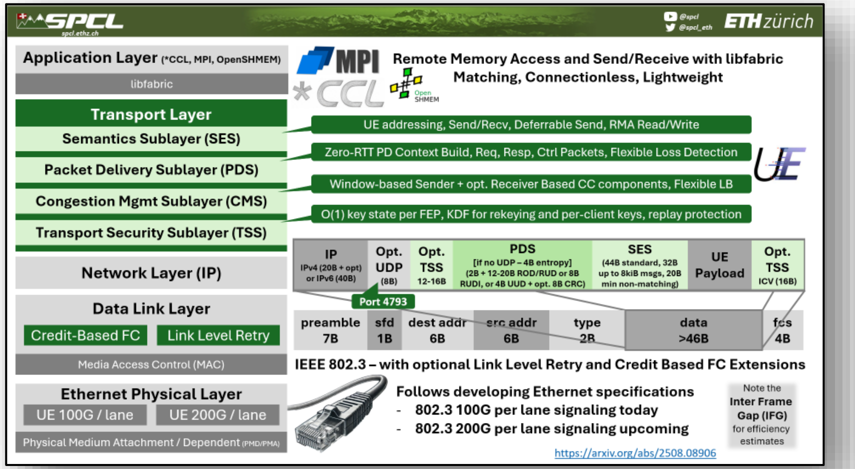
**More of SPCL's research:**

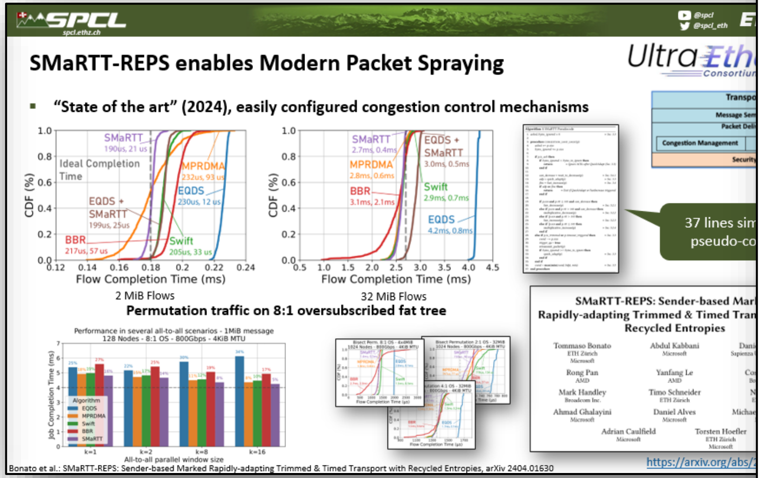
- youtube.com/@spcl — 210+ Talks
- twitter.com/spcl_eth — 1.4K+ Followers
- github.com/spcl — 2K+ Stars

… or spcl.ethz.ch








## Ultra Ethernet's Design Principles and Architectural Innovations

TORSTEN HOEFLER, ETH Zurich, Switzerland & Microsoft, USA
KAREN SCHRAMM, Broadcom, USA
ERIC SPADA, Broadcom, USA
KEITH UNDERWOOD, Hewlett Packard Enterprise, USA
CEDELL ALEXANDER, Broadcom, USA
BOB ALVERSON, Hewlett Packard Enterprise, USA
PAUL BOTTORFF, Hewlett Packard Enterprise, USA
ADRIAN CAULFIELD, OpenAI, USA
MARK HANDLEY, OpenAI, USA
CATHY HUANG, Intel, USA
COSTIN RAICIU, Broadcom, USA
ABDUL KABBANI, Microsoft, USA
EUGENE OPSASNICK, Broadcom, USA
RONG PAN, AMD, USA
ADEE RAN, Cisco, USA
RIP SOHAN, AMD, USA

https://arxiv.org/abs/2508.08906

arXiv:2508.08906v1 [cs.NI] 12 Aug 2025

The recently released Ultra Ethernet (UE) 1.0 specification defines a transformative High-Performance Ethernet standard for future Artificial Intelligence (AI) and High-Performance Computing (HPC) systems. This paper, written by the specification's authors, provides a high-level overview of UE's design, offering crucial motivations and scientific context to understand its innovations. While UE introduces advancements across the entire Ethernet stack, its standout contribution is the novel Ultra Ethernet Transport (UET), a potentially fully hardware-accelerated protocol engineered for reliable, fast, and efficient communication in extreme-scale systems. Unlike InfiniBand, the last major standardization effort in high-performance networking over two decades ago, UE leverages the expansive Ethernet ecosystem and the 1,000x gains in computational efficiency per moved bit to deliver a new era of high-performance networking.

### 1 Introduction

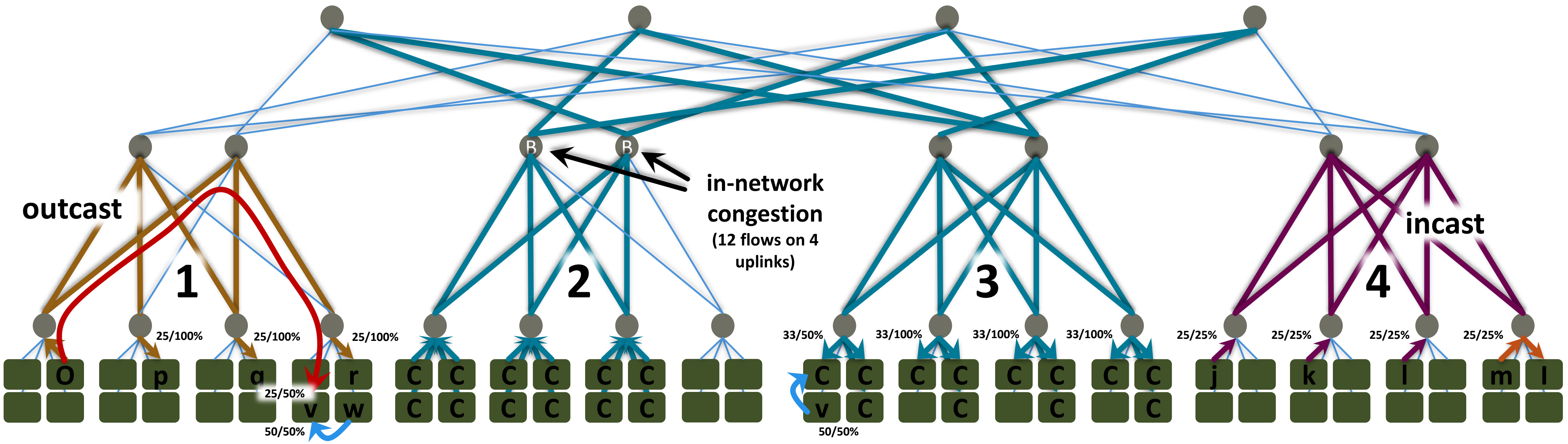
Ultra Ethernet (UE) standardizes a new protocol to support high-performance Artificial Intelligence (AI) and High-Performance Computing (HPC) networking over Ethernet. This paper, written by UE's authors, supplements the full specification by highlighting historical and innovative technical aspects of our nearly 2.5-year journey. It is designed to be approachable to a general audience and, thus, abstracts many details while using intuitive wording and explanations. The final authority for questions regarding UE is the full 562-page specification [35].

# More detailed requirements for HPC and AI networks


- Low latency / RTT
- Small message efficiency / message rate
- Tag matching (MPI, complex)
- Large # of connections (>10k for some apps)


- Extreme bandwidth requirements at endpoint
- No tags, in-order delivery though
- Connecting to few (<1k) endpoints
- Regular (oblivious) patterns (pre-plannable)

**Bulk Synchronous Application – Last Message / Flow that finishes determines performance!**


outcast
1
2
3
4
in-network congestion
(12 flows on 4 uplinks)
incast
B
B
25/100%
25/100%
25/100%
25/50%
50/50%
33/50%
33/100%
33/100%
33/100%
50/50%
25/25%
25/25%
25/25%
25/25%
O
p
q
r
v
w
C
j
k
l
m
l