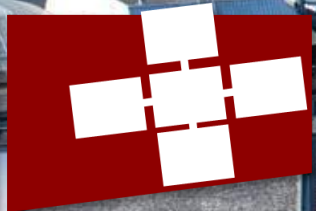


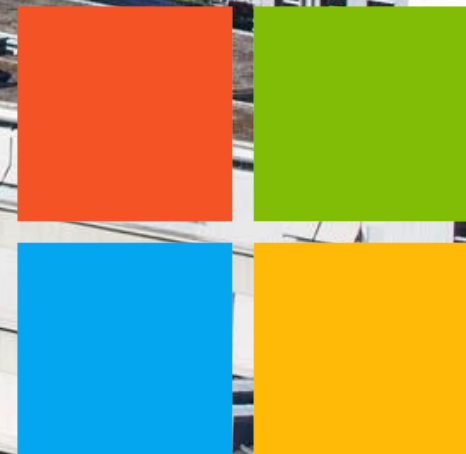
T. HOEFLER

Ultra Ethernet: An HPC and AI Interconnection Network Specification to Empower the Ethernet Ecosystem

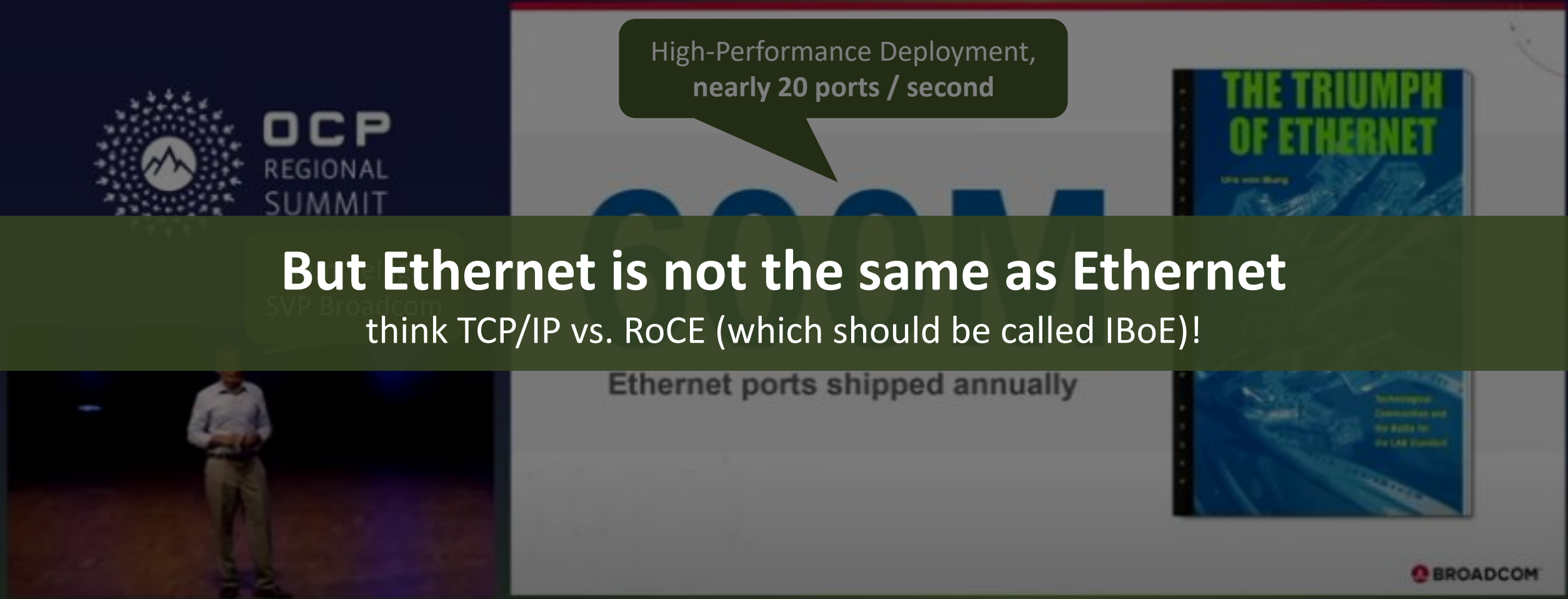
Invited talk at SOS, Engelberg, Switzerland



Institute of
Science and
Technology
Austria



The Ethernet Ecosystem – Is the **right one**!



OCP REGIONAL SUMMIT

The Ethernet Ecosystem

High-Performance Deployment,
nearly 20 ports / second

150M

Ethernet ports shipped annually

THE TRIUMPH OF ETHERNET
John W. Barry

BROADCOM

But Ethernet is not the same as Ethernet
think TCP/IP vs. RoCE (which should be called IBoE)!

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)

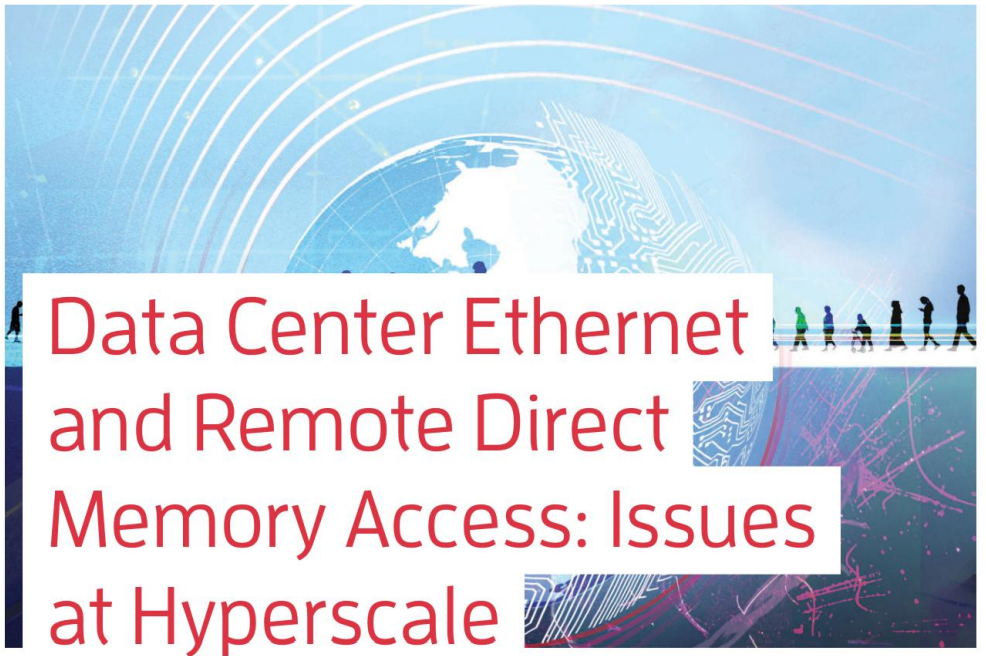


- Extreme bandwidth requirements at endpoint
- No tags, in-order delivery though
- Connecting to few (<1k) endpoints
- Regular (oblivious) patterns (pre-plannable)

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

Converging this HPC Networking Mess into a Unified Ethernet-based Standard

COVER FEATURE **TECHNOLOGY PREDICTIONS**



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

Torsten Hoefler^{ETH}, 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.



Founding Members



Ultra Ethernet Consortium

white Paper on ultraethernet.org

Overview of and Motivation for the Forthcoming Ultra Ethernet Consortium Specification

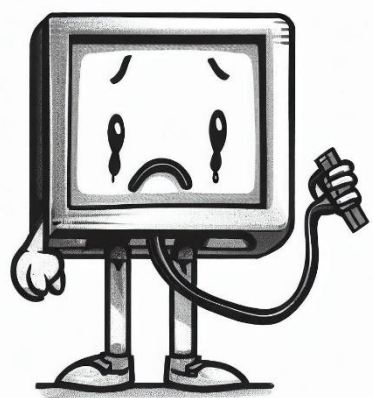
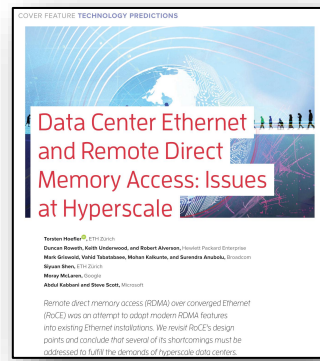
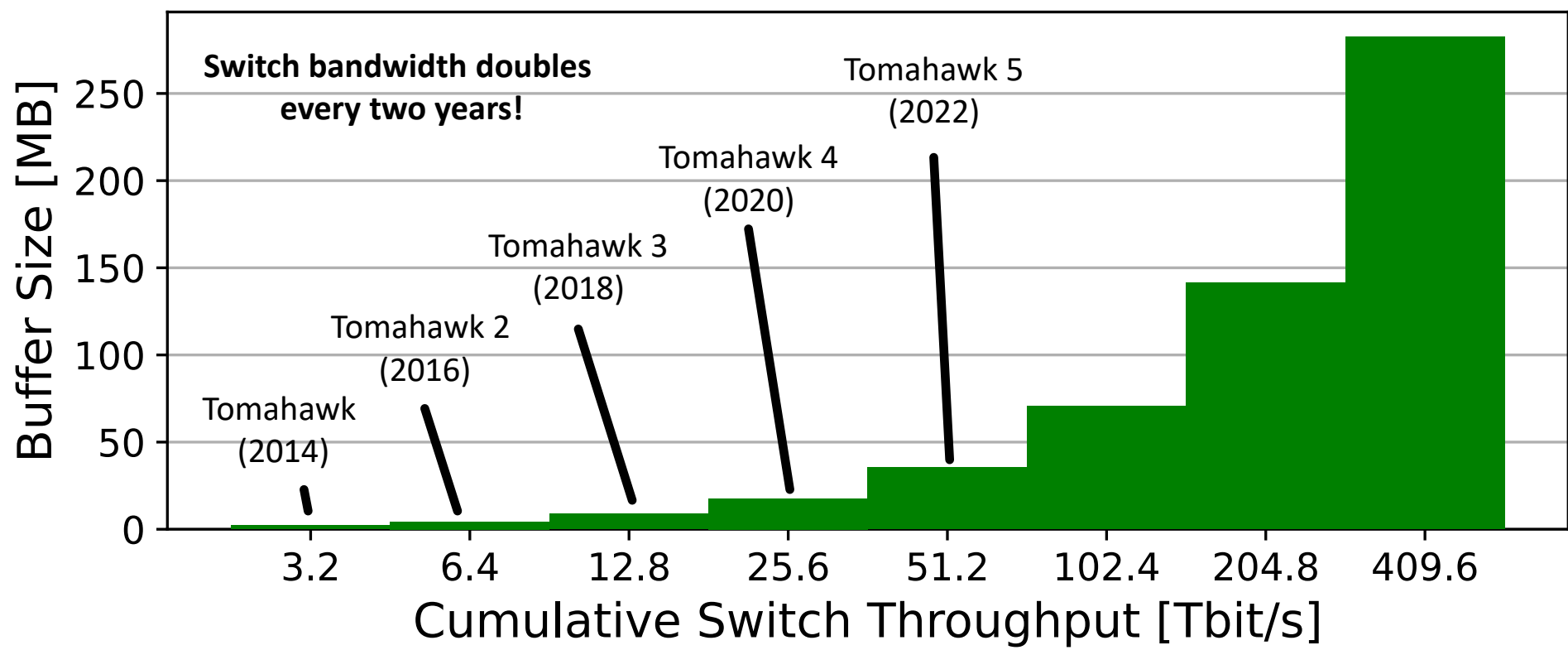
Networking Demands of Modern AI Jobs

Networking is increasingly important for efficient and cost-effective training of AI models. Large Language Models (LLMs) such as GPT-3, Chinchilla, and PALM, as well as recommendation systems like DLRM and DHEN, are trained on clusters of thousands of GPUs.

Getting there – Some RDMA Issues at Hyperscale



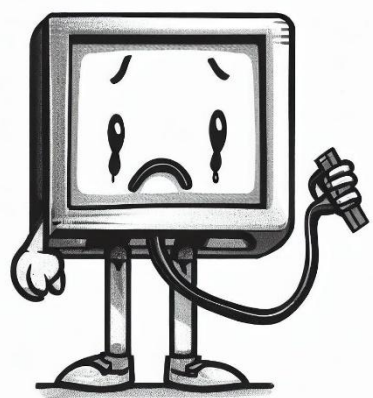
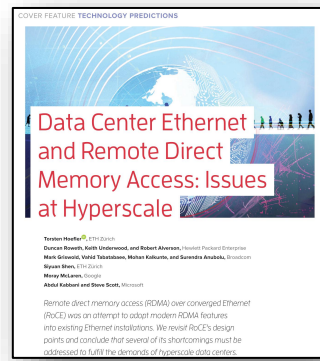
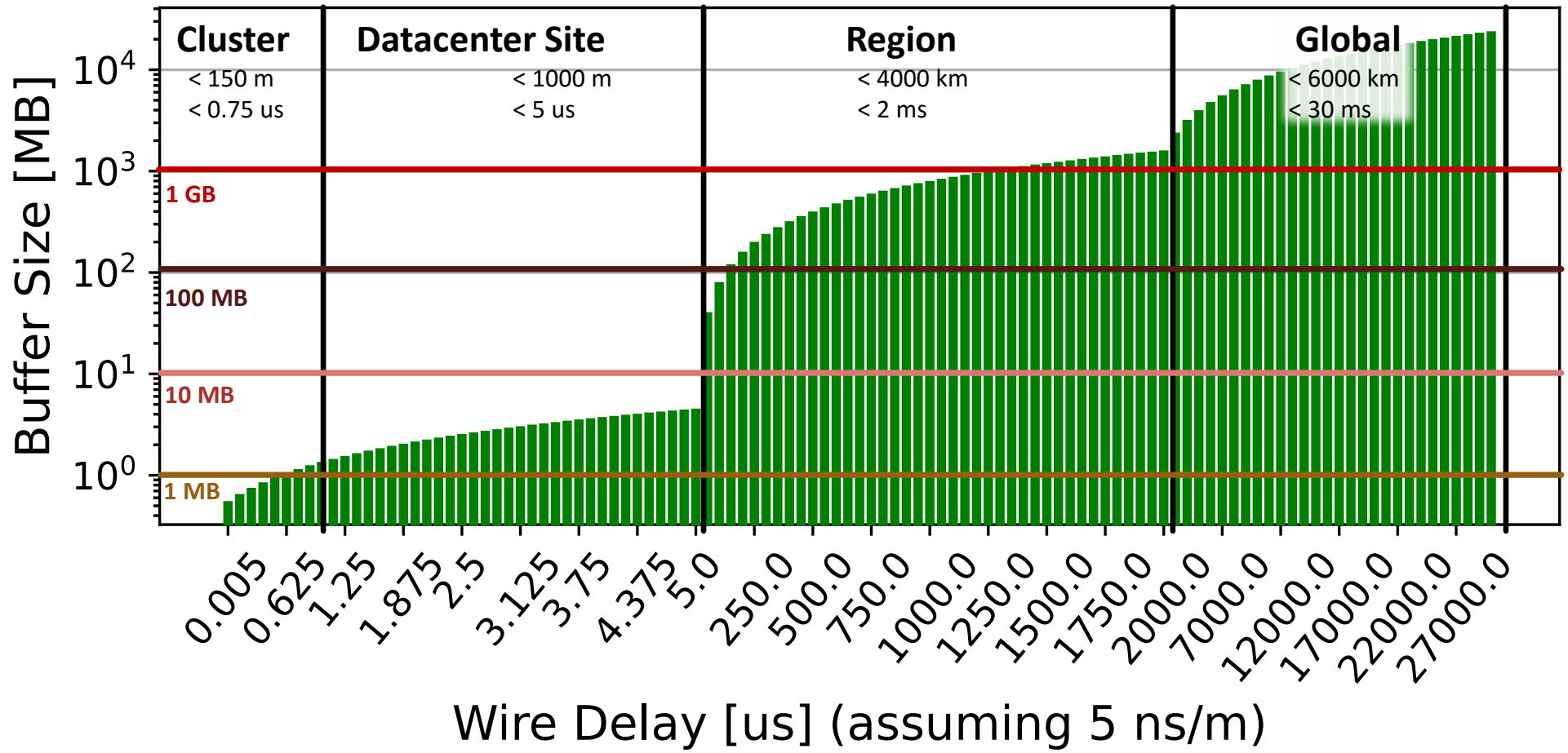
- 1) PFC requires excessive buffering for lossless transport – requires full $BDP = BW * RTT + MTU$ buffer!
 - Assuming 600ns traversal latency (FEC, arbitration, forwarding, wire delay), 9 kiB packets, 8 priorities



Getting there – Some RDMA Issues at Hyperscale



- 1) PFC requires excessive buffering for lossless transport – requires full $BW \cdot RTT + MTU$ buffer!
 - Per 800G port for longer distance links, BDP grows



[1] Hoefler et al.: “Datacenter Ethernet and RDMA: Issues at Hyperscale”, IEEE Computer June 2023, arXiv 2302.03337

Getting there – Some RDMA Issues at Hyperscale

- 2) Victim flows, congestion trees, PFC storms, and deadlocks

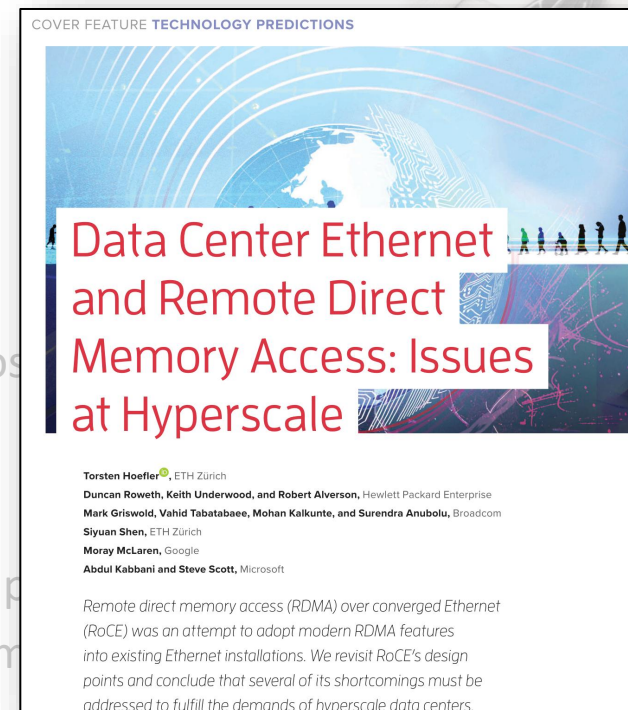
Many more such issues in the full paper!

S2

1/4

T2

- 3) Go-back-N retransmission
 - Simple recovery of lost packets (seq. number missing)
 - Yet, no real support for multi-pathing
 - Also retransmits full BDP on single loss (not a significant bandwidth loss)
- 4) Congestion control and colocated traffic
 - Interference with other traffic types, simple CC is not necessarily complete
 - Led to invention of DCQCN, TIMELY, HPCC, and likely many more – some



Ecosystem is quickly growing

Today 10 steering companies, 18 general member companies, 25 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!



*not all members listed

100+ member companies
1,300+ individual participants

Modernizing RDMA for HPC and AI

Classic RDMA

Lossless (PFC or CBFC) operation

In-order transport and delivery

Inefficient go-back-n

Proprietary congestion control (e.g., DCQCN)

Single-path routing

No load balancing and “link polarization”

Large state per queue pair

kb NIC memory per peer

Security added at higher layers

IPSec, N^2 contexts, known attacks

UltraEthernet
Consortium

Lossy (& lossless) operation

Out-of-order data and message delivery

(Un)Reliable (Un)Ordered - ROD, RUD/RUDI, and UUD

Open, configurable, and flexible CC

Per-packet multipathing and load balancing

Including (close-to) zero state REPS

Connection-less API

Ephemeral zero-RTT reliability state

Built-in security

Cluster-wide keying, zero state replay protection



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>



ReDMark: Bypassing RDMA Security Mechanisms

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

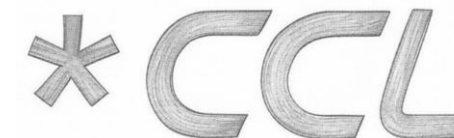
<https://www.usenix.org/conference/usenixsecurity21/presentation/rothenberger>



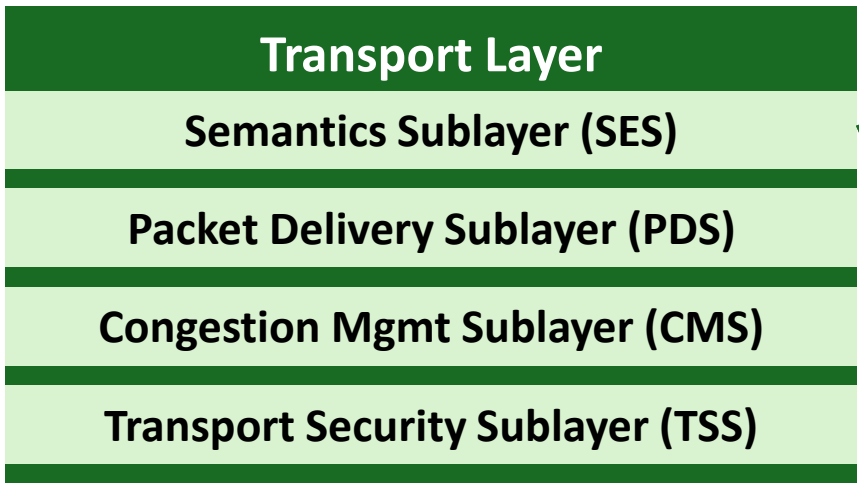
MPI



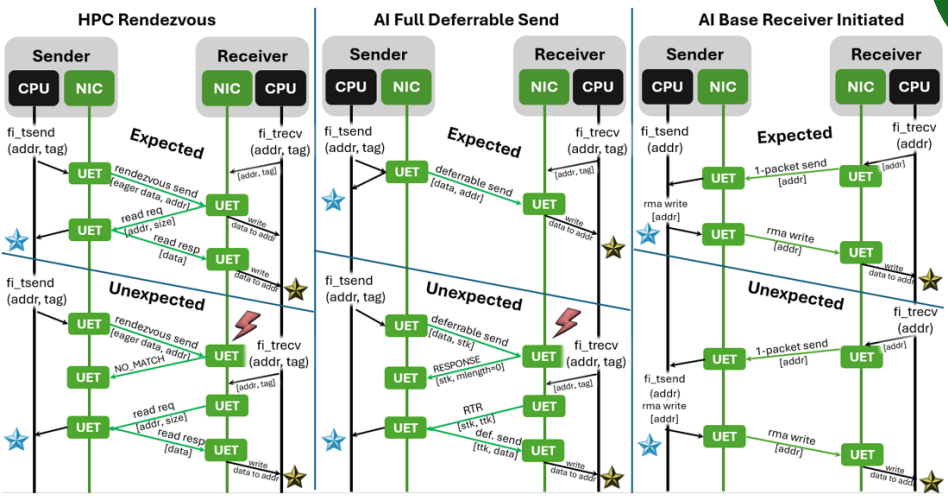
Open
SHMEM



Transport layer - sublayers

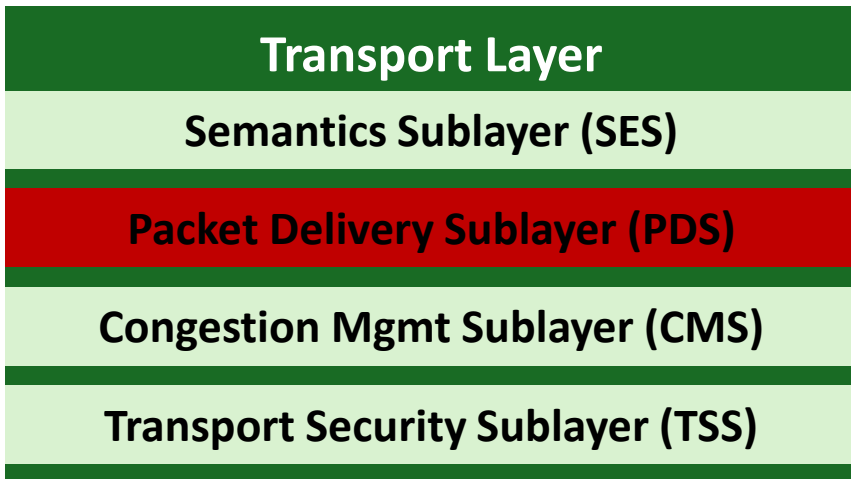


- Compatible with existing applications (libfabric) – **no change!**
- RDMA services: Send/Recv + RMA (Write, Read, Atomics)
 - Focus on MPI and *CCL semantics
- Scalable addressing to millions of endpoints
- Optimized extensions:
 - Deferrable Send for optimized HW (aimed at AI)
 - Rendezvous using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



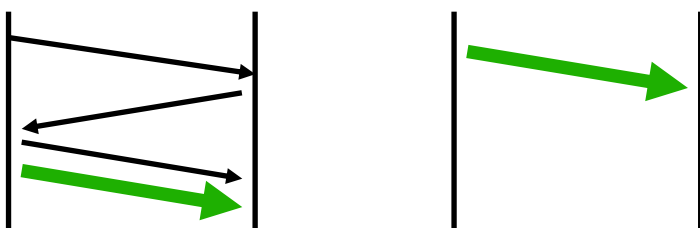
Use-case optimized communication profiles (AI Base, AI Full, HPC)

Transport layer - sublayers



- 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

Zero-RTT Startup

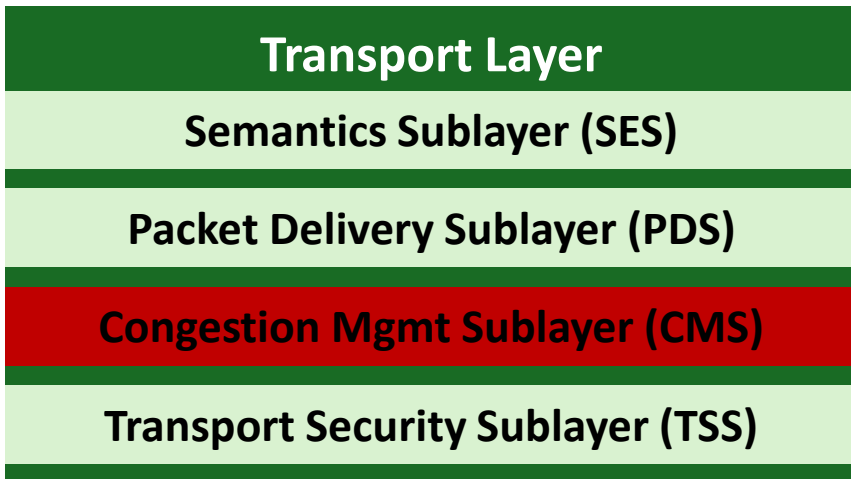


slow
(e.g., TCP)

fast
(UET)

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

Transport layer - sublayers



- 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

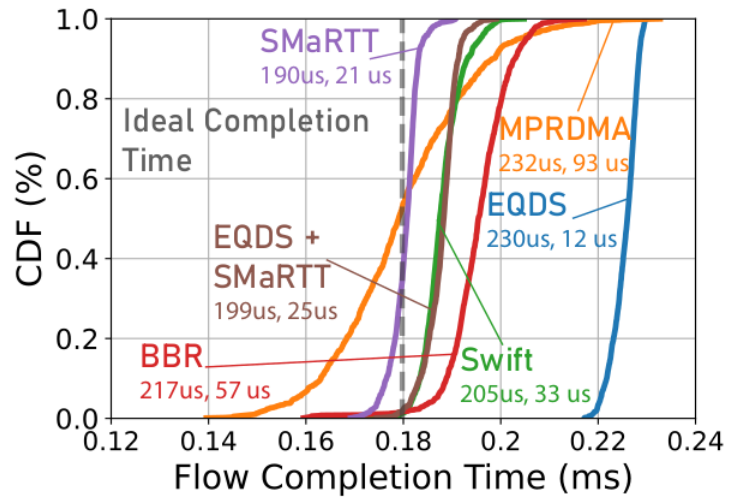
- Available in some UE products
- Receiver hands out credits
- Ideal for incast patterns

Work together for HPC+AI multi-pathing

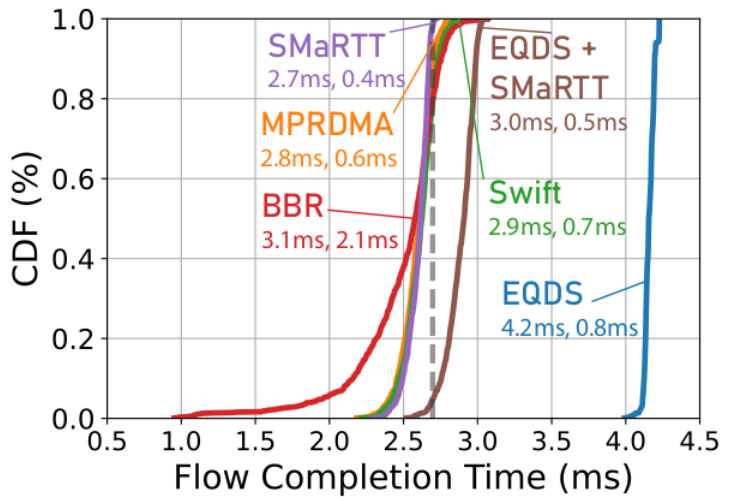
SMaRTT-REPS enables Modern Packet Spraying



- “State of the art” (2024), easily configured congestion control mechanisms

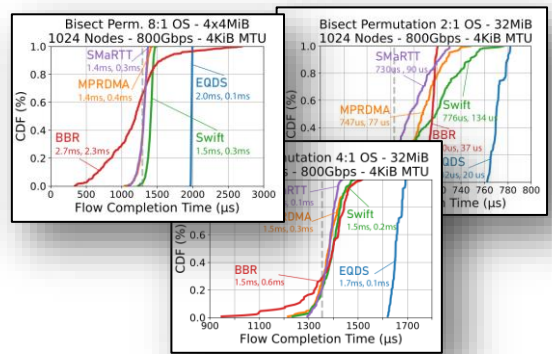
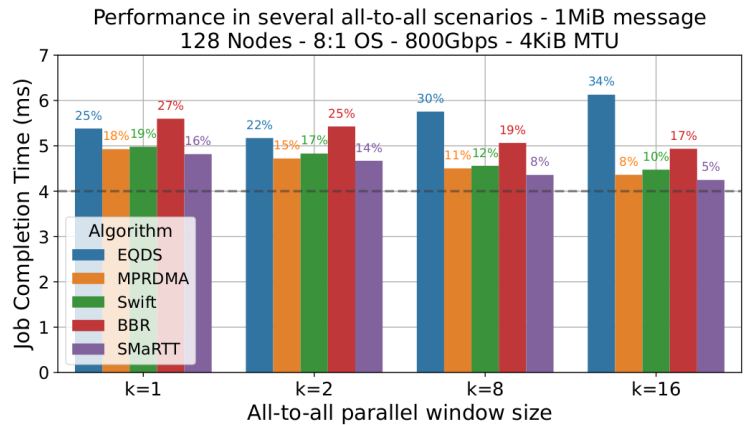


2 MiB Flows

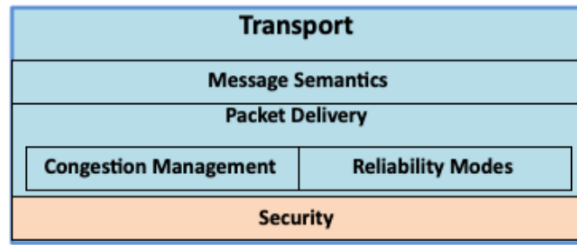


32 MiB Flows

Permutation traffic on 8:1 oversubscribed fat tree



```
Algorithm 1 SMaRTT Pseudocode
1: acked, bytes_ignored = 0
2:
3: procedure congestion_loop_loop(p)
4:   acked += p.size
5:   bytes_ignored += p.size
6:
7:   if p.is_ack then
8:     if bytes_ignored < bytes_to_ignore then
9:       return
10:    else
11:      return
12:   end if
13:
14:   can_decrease = wait_to_decrease(p)
15:   adp = quick_adapt(p)
16:   fast = fast_increase(p)
17:   if adp or fast then
18:     return
19:   end if
20:
21:   if p.con and p.rt <= rtt and can_decrease then
22:     fair_decrease(p)
23:   else if p.con and p.rt > rtt and can_decrease then
24:     multiplicative_decrease(p)
25:   else if p.con and p.rt > rtt then
26:     fair_increase(p)
27:   else if p.con and p.rt <= rtt then
28:     multiplicative_increase(p)
29:   end if
30:
31:   if p.is_trimmed or p.timeout_triggered then
32:     cond = p.size
33:     trigger_ga = true
34:     retransmit_packet(p)
35:     if bytes_ignored >= bytes_to_ignore then
36:       quick_adapt(p)
37:     end if
38:   end if
39:   cond = max(cond, bdp, mru)
40: end procedure
```



37 lines simple pseudo-code

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 | |

Transport layer features

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 in 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)