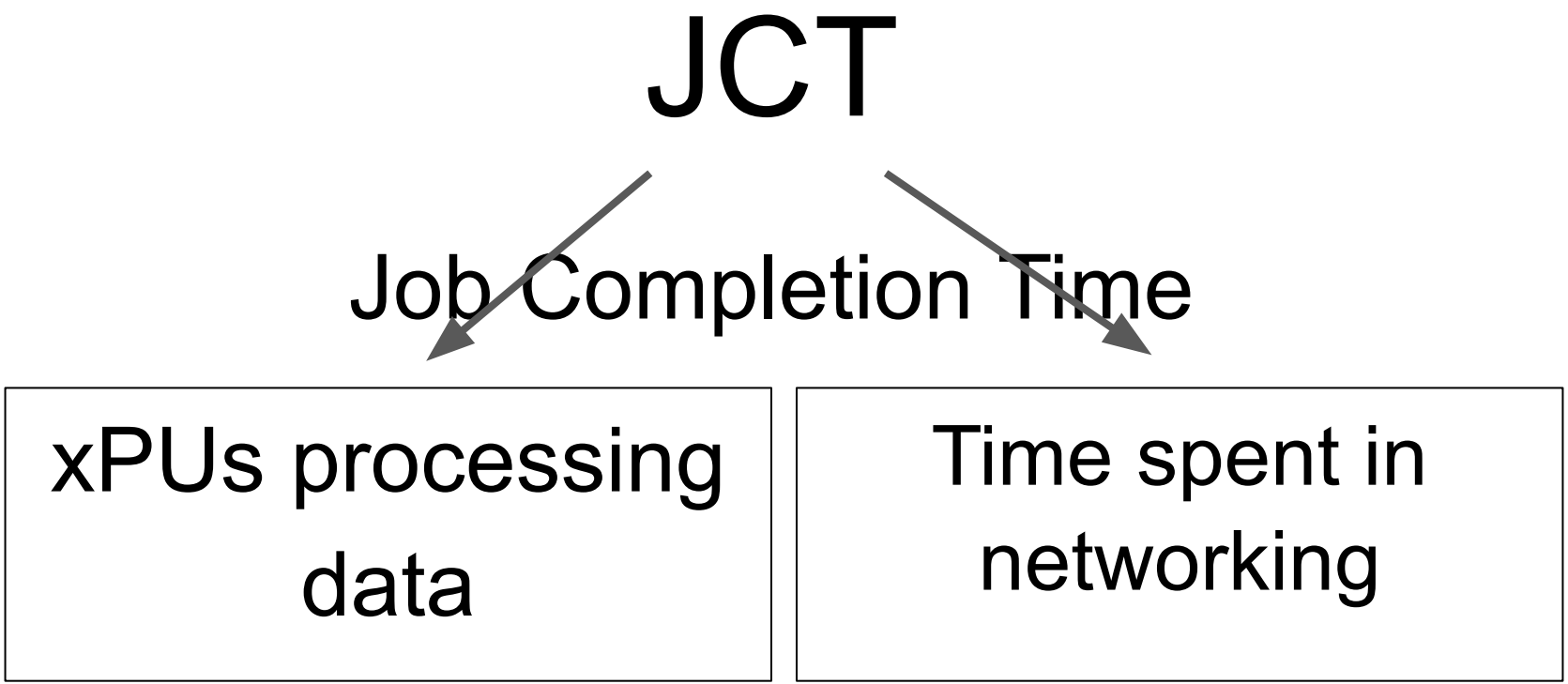
# Ultra Ethernet

Johan Ervenius
Systems Engineer, Arista Networks

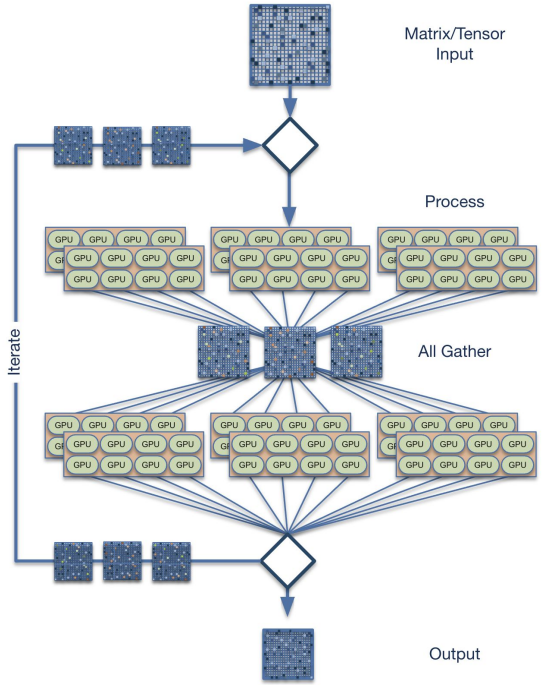
ARISTA


JCT
Job Completion Time
xPUs processing data
Time spent in networking

# What's the Problem?

## AI and HPC networks are different

- Endpoints are fast, Load is high
- Flows are few and high BW
- RTTs are short
- Flows are synchronized
- Completion time determined by slowest flow




**Vanilla networking doesn't meet the needs**

# Ultra Ethernet Consortium – Who and Why

AMD · ARISTA · BROADCOM · CISCO · EVIDEN an atos business · Hewlett Packard Enterprise · intel. · Meta · Microsoft · ORACLE

Alibaba Cloud · ARRCUS · Baidu 百度 · 世纪互联 VNET · ByteDance · cadence · CORNELIS NETWORKS · DELL Technologies · enfabrica · HUAWEI · IBM · JUNIPER NETWORKS · KEYSIGHT TECHNOLOGIES · Lawrence Livermore National Laboratory · Lenovo · MARVELL · H3C · NOKIA · NVIDIA · Preferred Networks · PURESTORAGE · Qualcomm · Spirent · SYNOPSYS · ZTE中兴

Accelink · ALPHAWAVE SEMI · AsteraLabs. · Astorfusion · auradine · NETSWIFT 阿迅科技 · centec · ciena · Clounix · CoMIRA SOLUTIONS · CREDO · DriveNets · FATHOM RADIANT · FUJITSU · Google Cloud · GRAPHCORE · GROVF · IFRAEO · IIJ Internet Initiative Japan · ip infusion · KALRAY · kiwi PACKETS

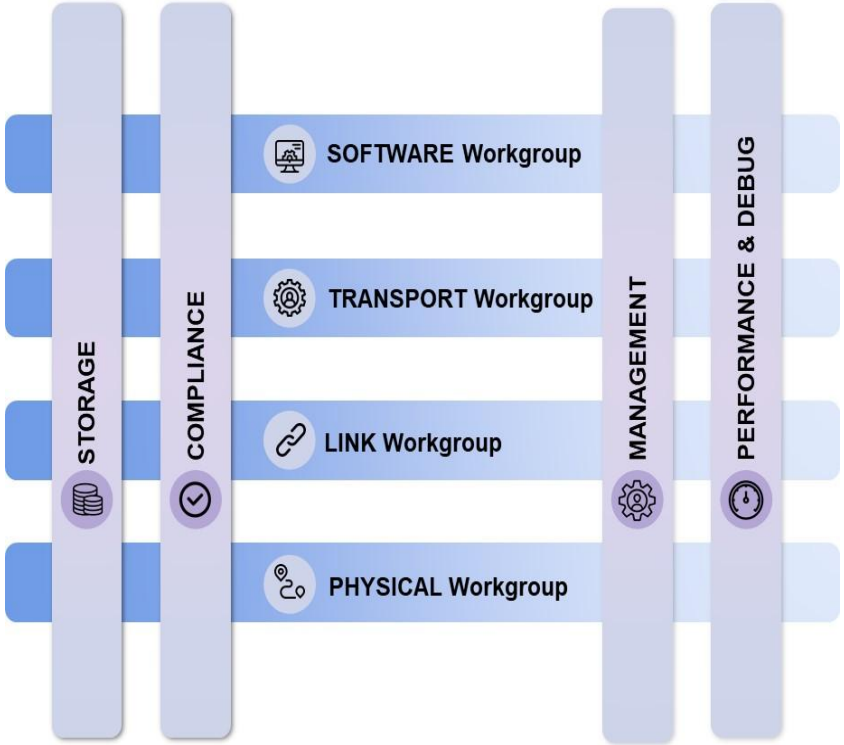
Los Alamos NATIONAL LABORATORY · MANGOBOOST · MemVerge · MICROCHIP · Micron · molex · NEUREALITY · NUMASCALE · Qumulo · Rivos · Ruijie · Salience Labs. · SAMSUNG SDS · Sandia National Laboratories · SAPEON · SCALA COMPUTING · SDTECH 数渡科技 · Tencent 腾讯 · ufiSpace · VDURA · Xsight · 云脉芯联 YUNSILICON

>100 member companies
>1300 active participants

Mission:
Advance an Ethernet-Based Open, Interoperable, High-Performance Full-Stack architecture to meet the Growing Demands of AI and HPC at Scale

# Ultra Ethernet Consortium Activities

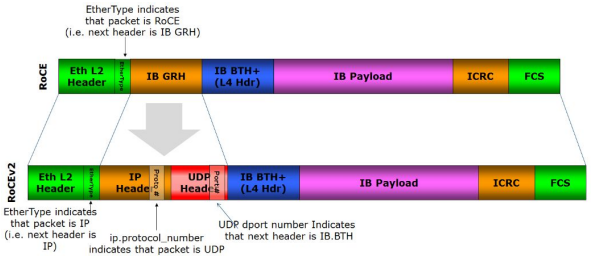
- Many working groups
- 1 specification, many layers
- The spec is 500 pages




**UEC is a JDF project and an International Standards Organization**

# RMA is critical to performance

- Accelerators today communicate with RMA
- RMA is hardware delivery straight to/from memory
  - Kernel bypass, zero-copy
  - Hardware loss detection, retrans, loss recovery
- RDMA over IP (RoCEv2) is a widely deployed RMA implementation

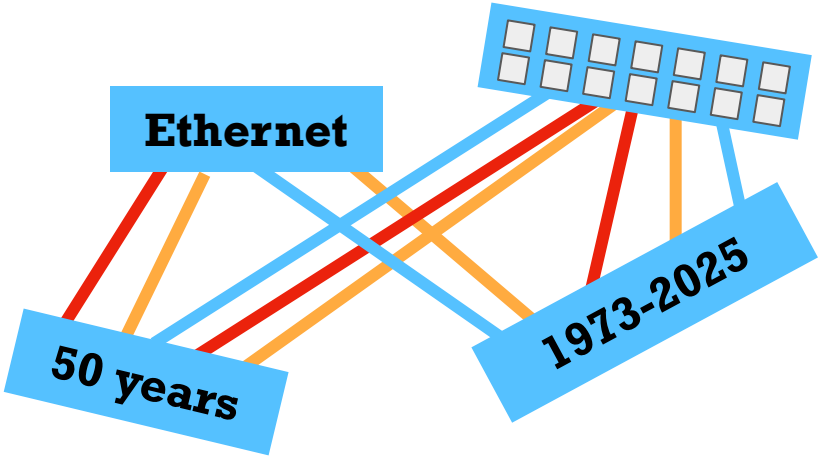



RoCE headers

**RMA is a *great* concept**

ARISTA

# Ethernet is the right foundation for RMA




- Broad ecosystem
  - NICs, switches, optics, cables
  - Multi-vendor at all layers
- Rapid innovation
- Many tools for operations, management, testing
- Scales to millions: addressing, routing, management, provisioning
- Universally understood - books, courses, websites, classes, …

**UEC builds on Ethernet**

# Why revisit RMA … especially RoCE?

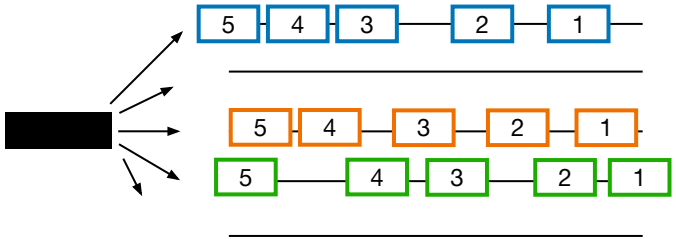

- Lack of multipathing
  - in-order packet delivery is limiting
- Go-back-N Recovery is inefficient, forcing lossless networks
- Congestion control (DCQCN) is hard to tune, not easy to (inter)operate
- Scale requirements are increasing
- Integrated security is important

**RMA is great, but it's time to revisit the protocol**

# Load Balancing

## The Key Problem to Solve

# Flows and packet ordering



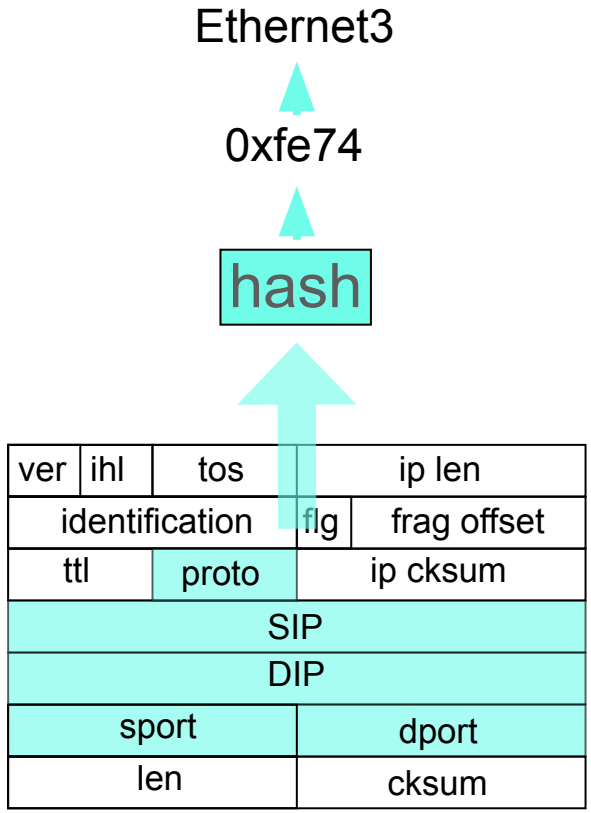

- Networks today keep packets within a single L4 flow in order
- Because transport protocols (TCP, RDMA) don't like out of order packets
  - out-of-order packets are interpreted as loss
  - repeated loss is interpreted as congestion
  - congestion results in slowing down

**so don't reorder packets within a flow**

# Choosing a path for each flow

**Spreading flows over all ECMP paths**

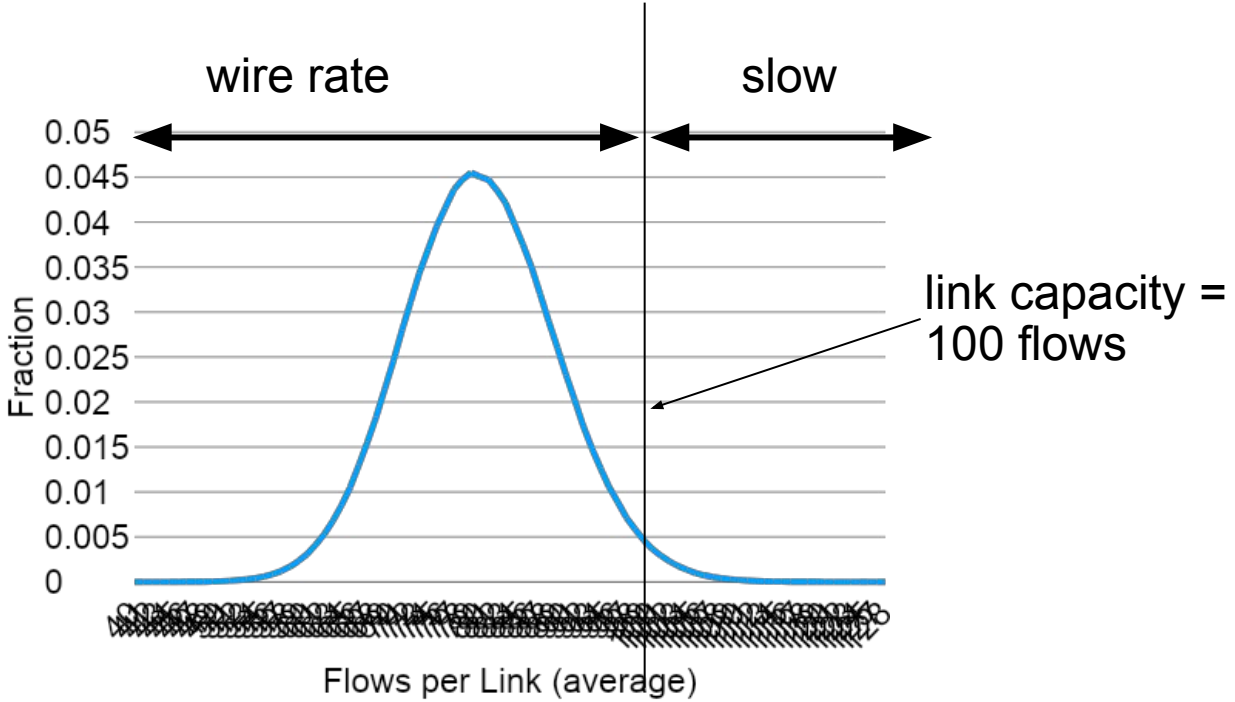
- Generally, with a hash of L4 ports and IP
- Works great if many small flows per link




**...but it's hard to spread flows evenly when there are not many**

# So, how good is flow hashing?

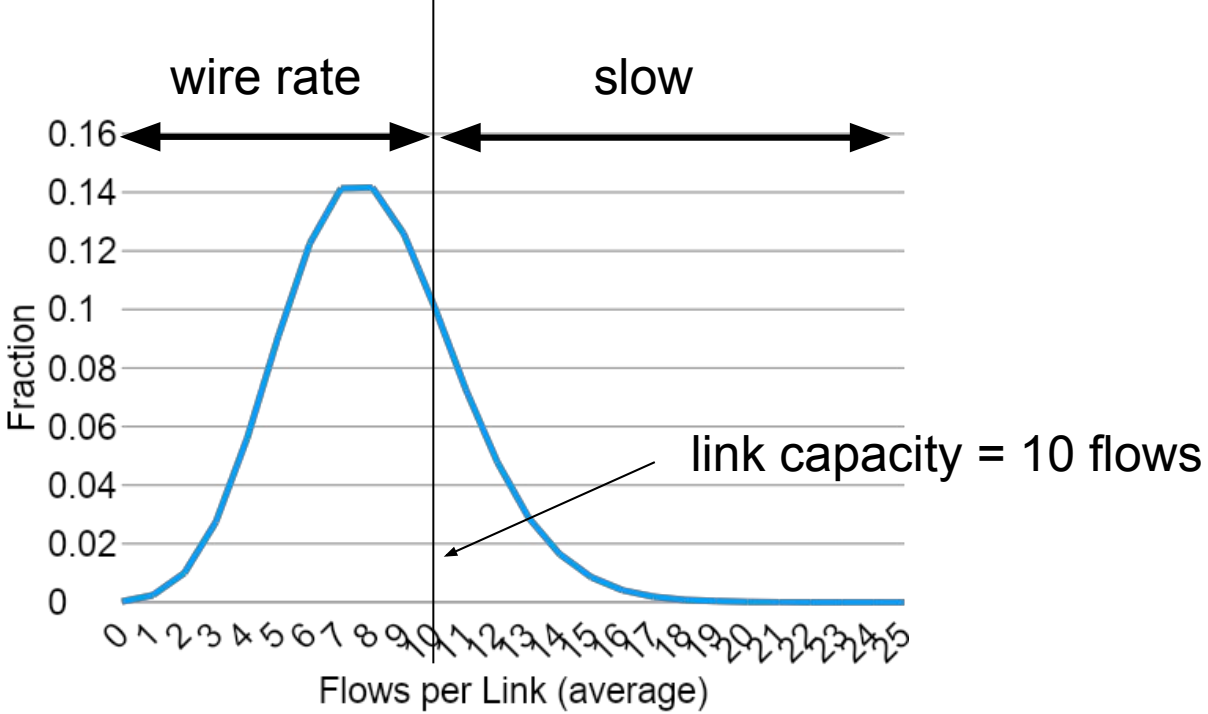
# Average Utilization




32 servers at 100G with 80 flows each at 1Gbps, 32 uplinks

99.95% throughput - great

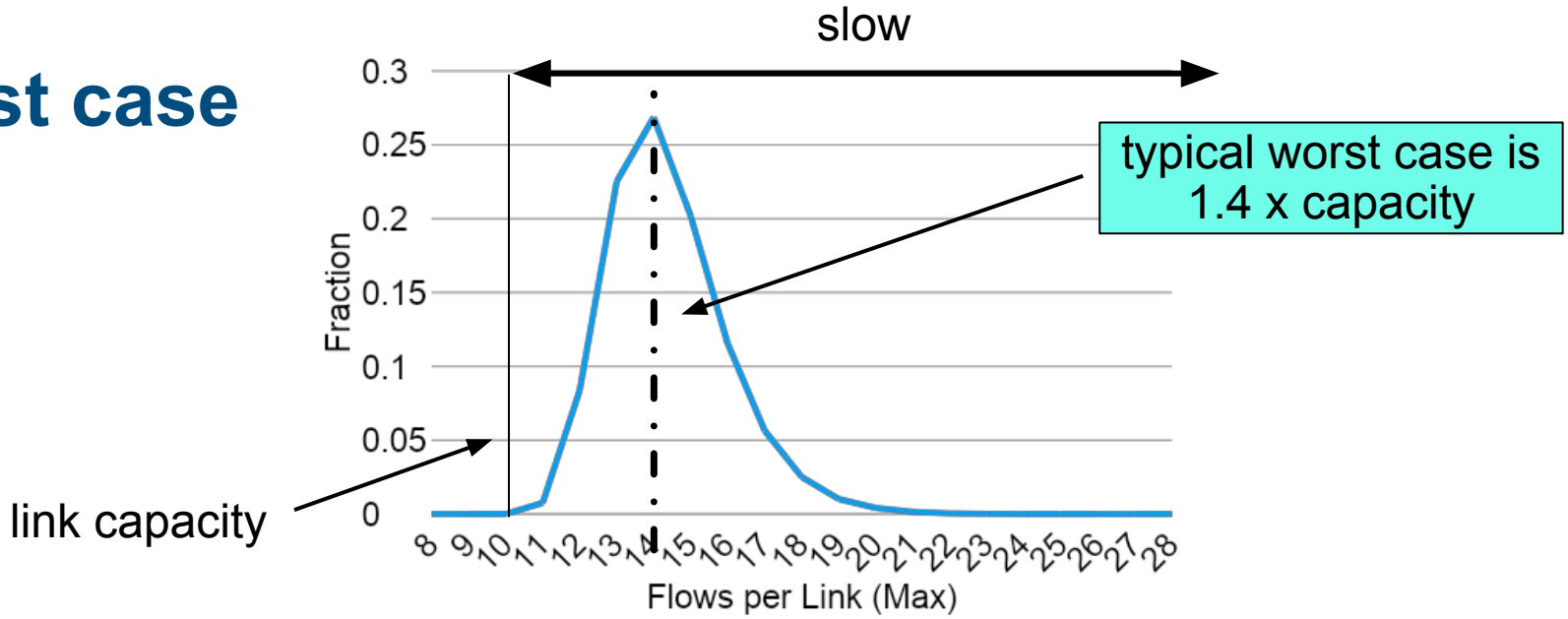
# Average Utilization




32 servers at 100G with 8 flows each at 10Gbps, 32 uplinks

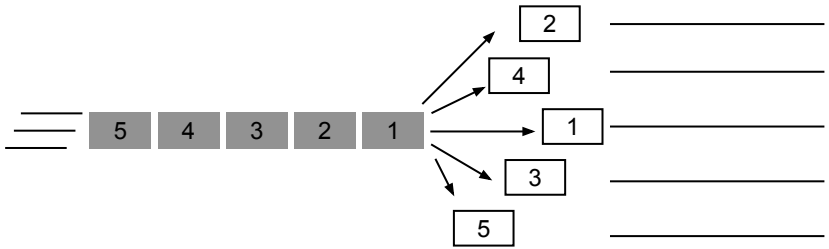
96.8% throughput - ok

# Worst case




32 servers at 100G with 8 flows each at 10Gbps, 32 uplinks

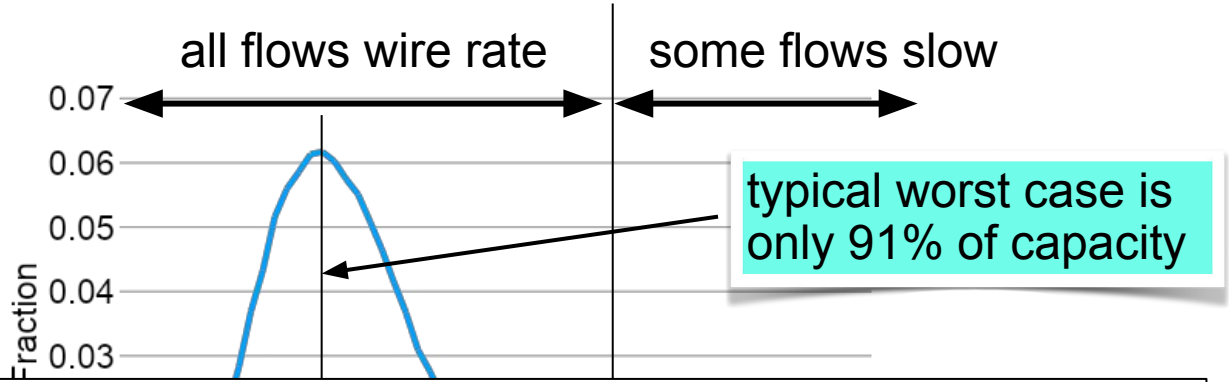
71% efficiency is the expected performance of the worst-case




forget about keeping packets of a flow in order…

# What if…
# One flow could use *ALL the* paths?

# Worst case




**but TCP and vanilla RDMA don't work**

32 servers, packet-sprayed 204 ways on 32 uplinks

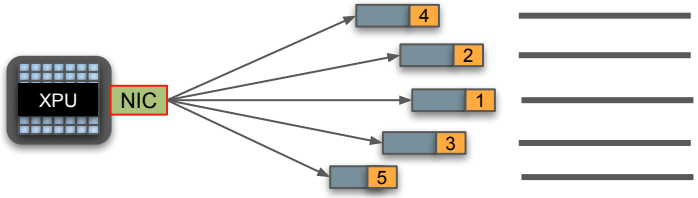
80% offered load

99.98% efficient for an application driven by worst-case

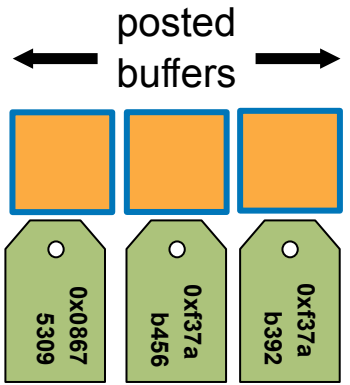
# Ultra Ethernet Transport

# Enable the transport protocol to spray

A key tenet of the UET





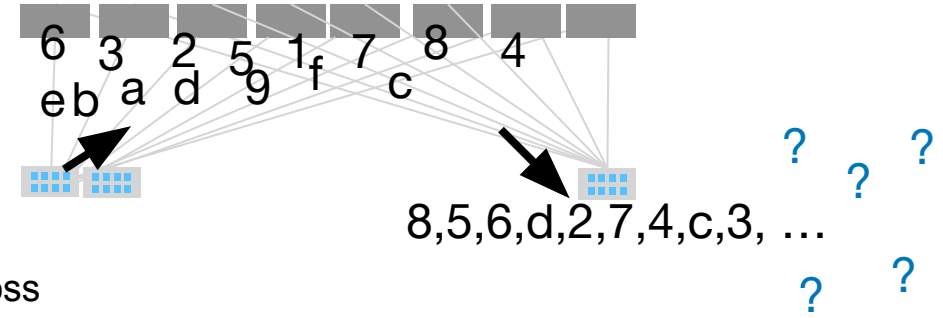



- Don’t insist on packet ordering within a flow
- Tag packets with their ultimate destination
  - eliminates the need to reorder on arrival
  - packets can be immediately placed in memory

**UET: RMA with out-of-order arrivals**

# Packet Spraying Challenge

## Loss Detection in an OOO protocol

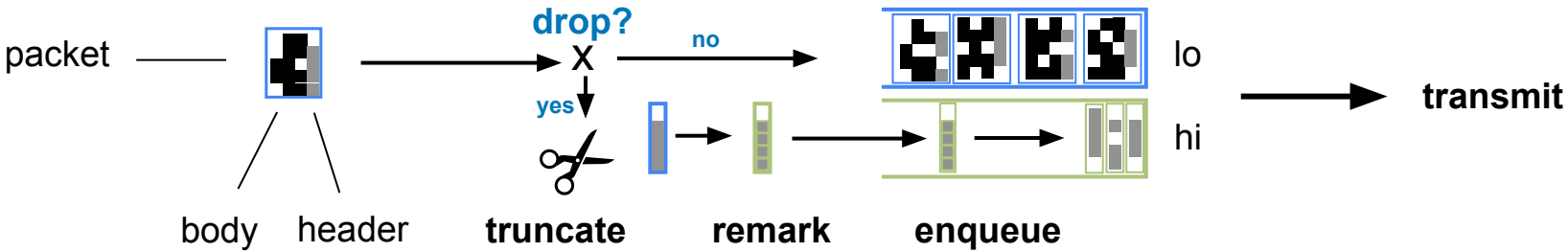



- Generally, timeout or out-of-order implies loss
- With spraying, out of order is not a simple concept
  - packets taking different paths can arrive in any order
- Fast timeouts are made harder because of variable delay across paths

**Need new methods to detect loss**

# Packet trimming

## Chop, don't drop!



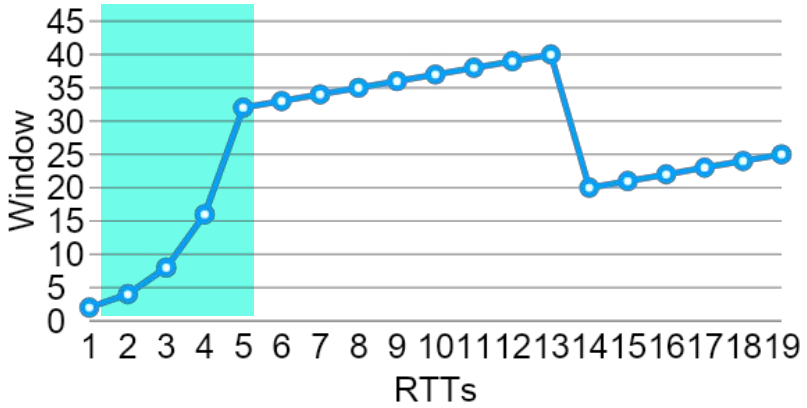

- Truncate ("trim") to 64 bytes instead of dropping
- Mark the DSCP as "trimmed"
- Enqueue truncated pkt in high priority queue for a faster congestion signal

**Switch support for fast loss detection**

# Packet Spraying Challenge

## Congestion Control with high bandwidth and short RTT

- How is UET CC different from TCP?
- Get to wire rate very quickly
  - 1MB takes 10 usec at 800gbps = 1 RTT
  - Must back off quickly when congestion is noticed
- No time to wait for TCP slow start




**Need new methods to detect loss**

ARISTA

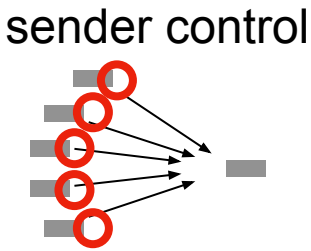
# Fast Speed-Up and Slow-Down

- We need to ramp quickly and slow down quickly
- Losses and/or delays tell the transport to slow-down
- UET needs new algorithms for a sprayed network

**Existing transports are too slow and/or depend on ordering**

# UET congestion control

## Network Signal Congestion Control (NSCC)

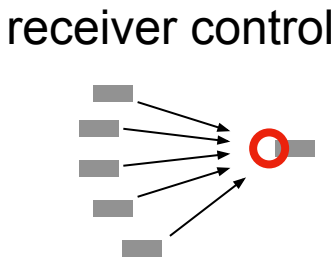



- **Sender-based** (default)
- Fast ramp, fast slowdown
- **Optimized to detect core congestion**
- Uses 3 congestion signals
  - Delay tracked as time from sending packet to receiving ACK
  - ECN as leading indicator of queuing
  - Trimming as indication of drops

**Handles spraying and OOO**

# UET congestion control

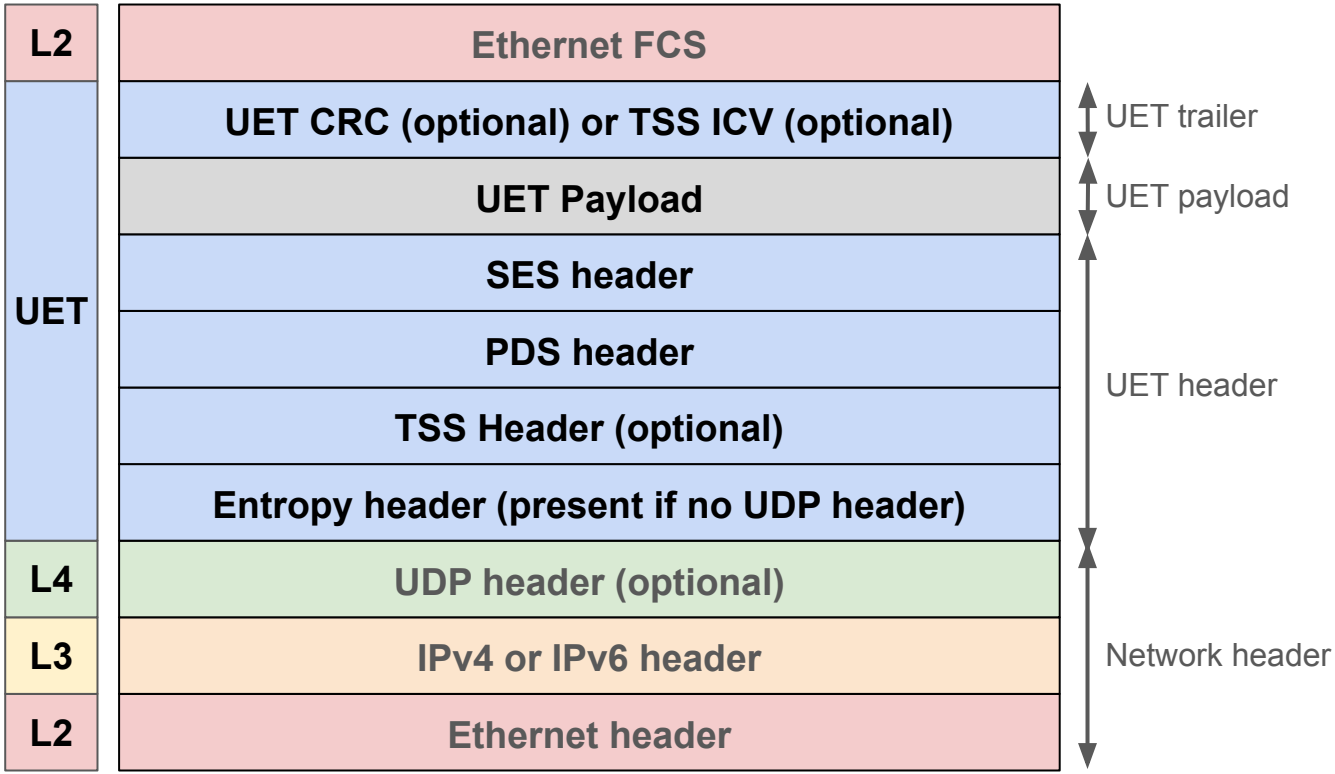
## Receiver-credit congestion control




- **Receiver-based** (optional)
- **Optimized for handling Incast**
- Receiver-generated credit
- Optimistic transmission before credits received
- Looks a lot like a VOQ architecture

**Works along or with sender based**

# UET Packet


L2
Ethernet FCS
UET
UET CRC (optional) or TSS ICV (optional)
UET Payload
SES header
PDS header
TSS Header (optional)
Entropy header (present if no UDP header)
L4
UDP header (optional)
L3
IPv4 or IPv6 header
L2
Ethernet header
UET trailer
UET payload
UET header
Network header

# Ultra Ethernet Across the Layers

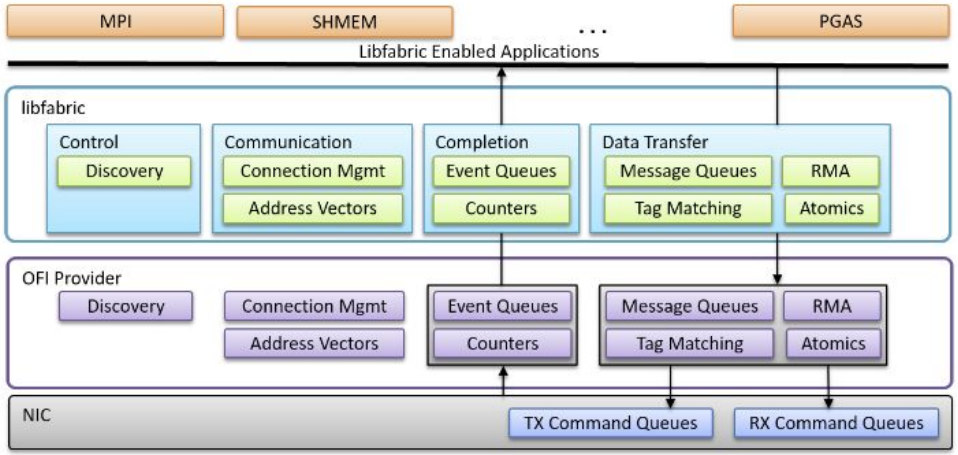
Application, Transport, Network, Link Layer

ARISTA

# libfabric

## by the OpenFabrics Alliance

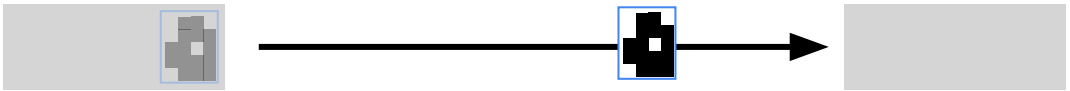
- UEC selected libfabric 2.0 as a modern API
- Generic APIs for High Performance Communication
  - RMA
  - Tagged messages, Atomics
  - Collective operations
  - event queues, completion queues




**Sockets API isn't rich enough for HPC/AI**

ARISTA

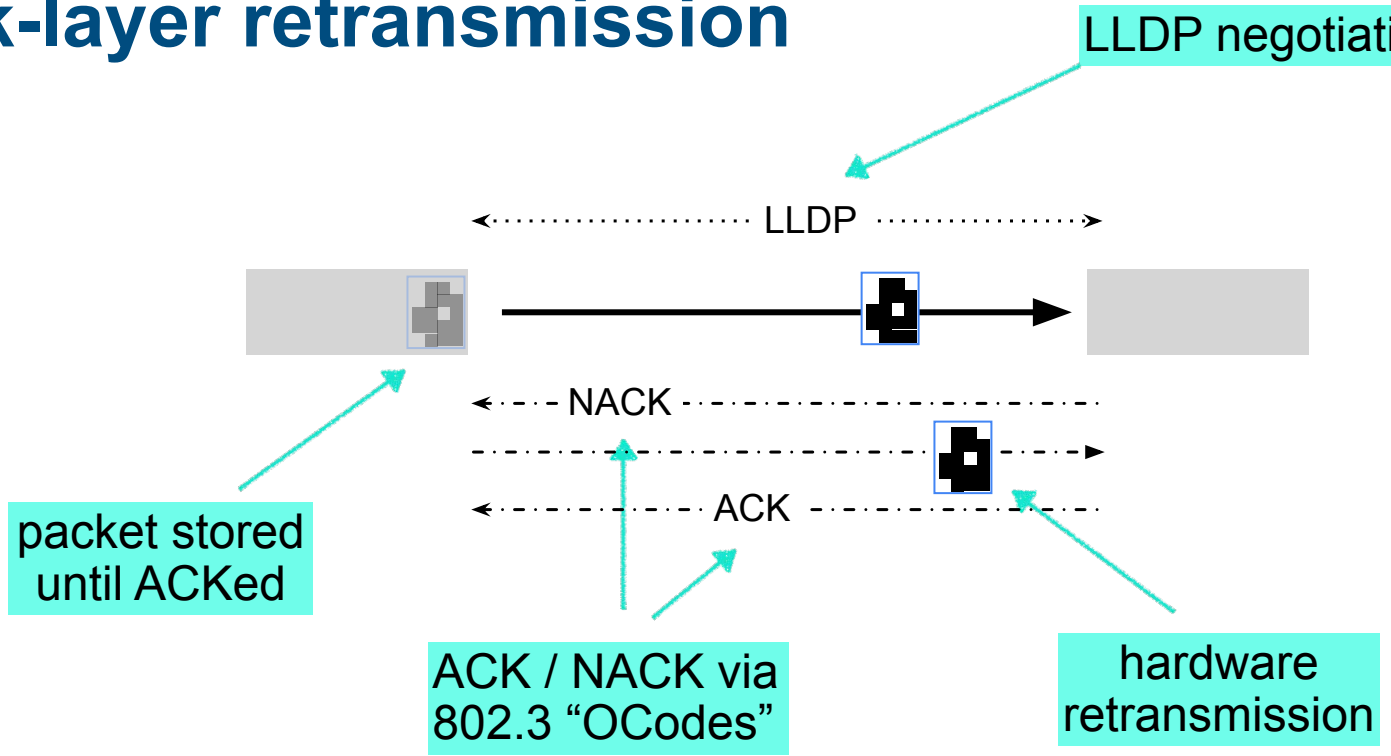
# Link-layer retransmission



- Link and transceiver failures are a fact — and impact workloads
- An AI/HPC datacenter could have 256,000-512,000 transceivers
- Local retransmission to avoid end-to-end retransmits

**Improves tail latency**

# Link-layer retransmission

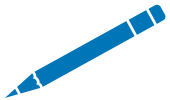



**Improves tail latency**

# Future

## UEC will continue after the 1.0 release

### Sooner



- Storage - Storage APIs on UET
- Management - OpenConfig / RedFish
- Compliance and Testing, for profiles and optional features
- Performance and Debugging
- Telemetry - CSIG and BTS
- In Network Compute

### Later, maybe…

- Programmable congestion control
- More topologies - DragonFly, DragonFly+, Slimfly, xFly
- UET for regional / metro?
- Scale-up?

# In Conclusion

Ultra Ethernet Consortium

## Networks for AI

- Ethernet: the standard solution for AI and HPC networks
  - Ethernet does and will support the features critical to AI and HPC
  - Ethernet will scale to 1,000,000s of GPUs
- Ultra Ethernet is ready for AI and HPC of the future

# Thank you!

ARISTA