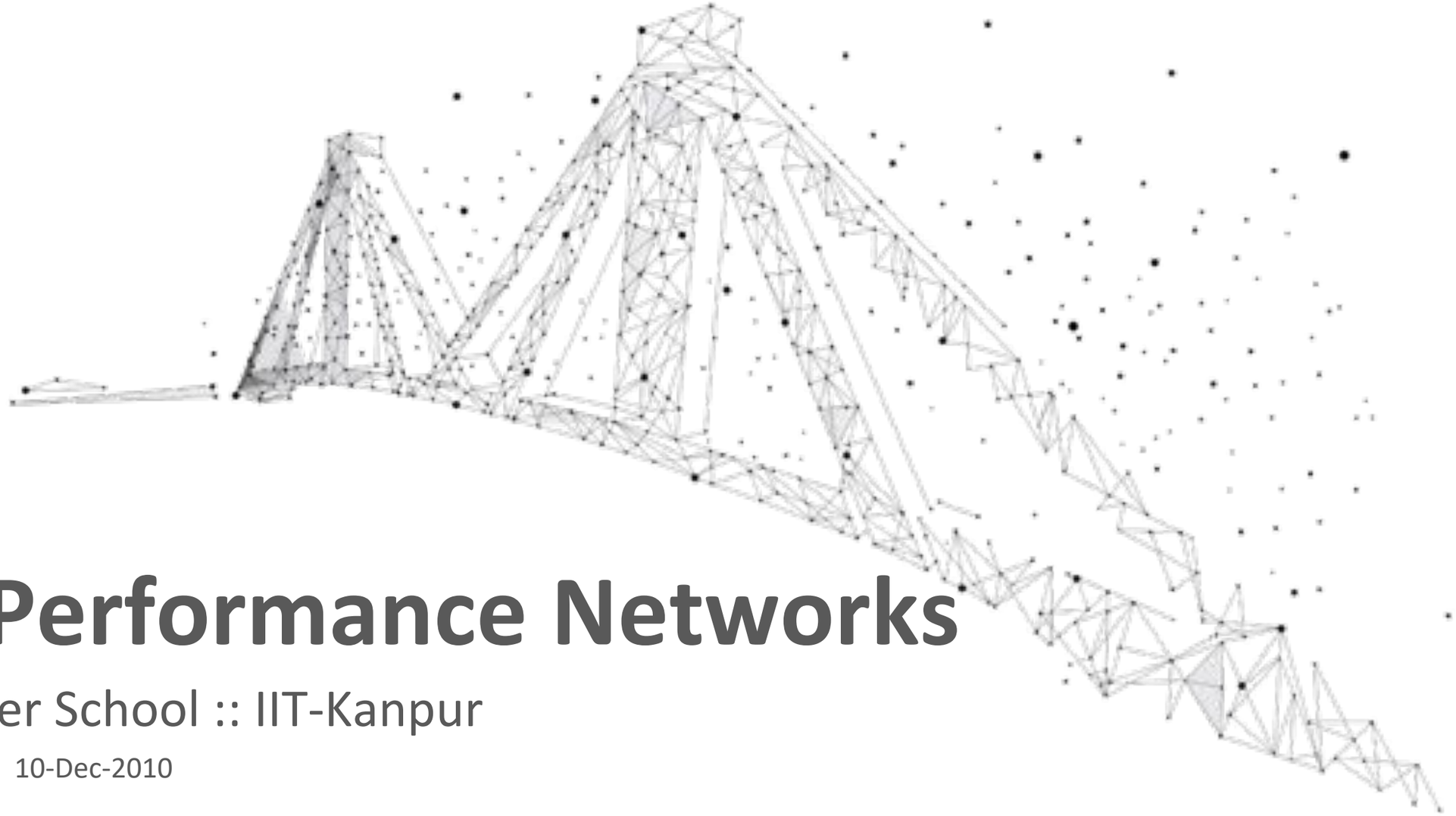


High Performance Networks

ACM Winter School :: IIT-Kanpur

Ashrut Ambastha 10-Dec-2010

Agenda

Connect the dots between what we study and how it maps to real supercomputing systems…

Networks

Interlinking of Multiple Compute Elements

SMP?? Clusters??

Cache Coherency?? Message Passing??

Bandwidth?? Latency??

HPC Interconnect History and Development

First Teraflop Supercomputer
Sandia ASCI Red
Intel

First Petaflop Supercomputer
LANL Roadrunner
IBM / Mellanox InfiniBand

Quadrics
QsNet
gnodal
QsNet with Gateway to Ethernet

Myricom
Myrinet

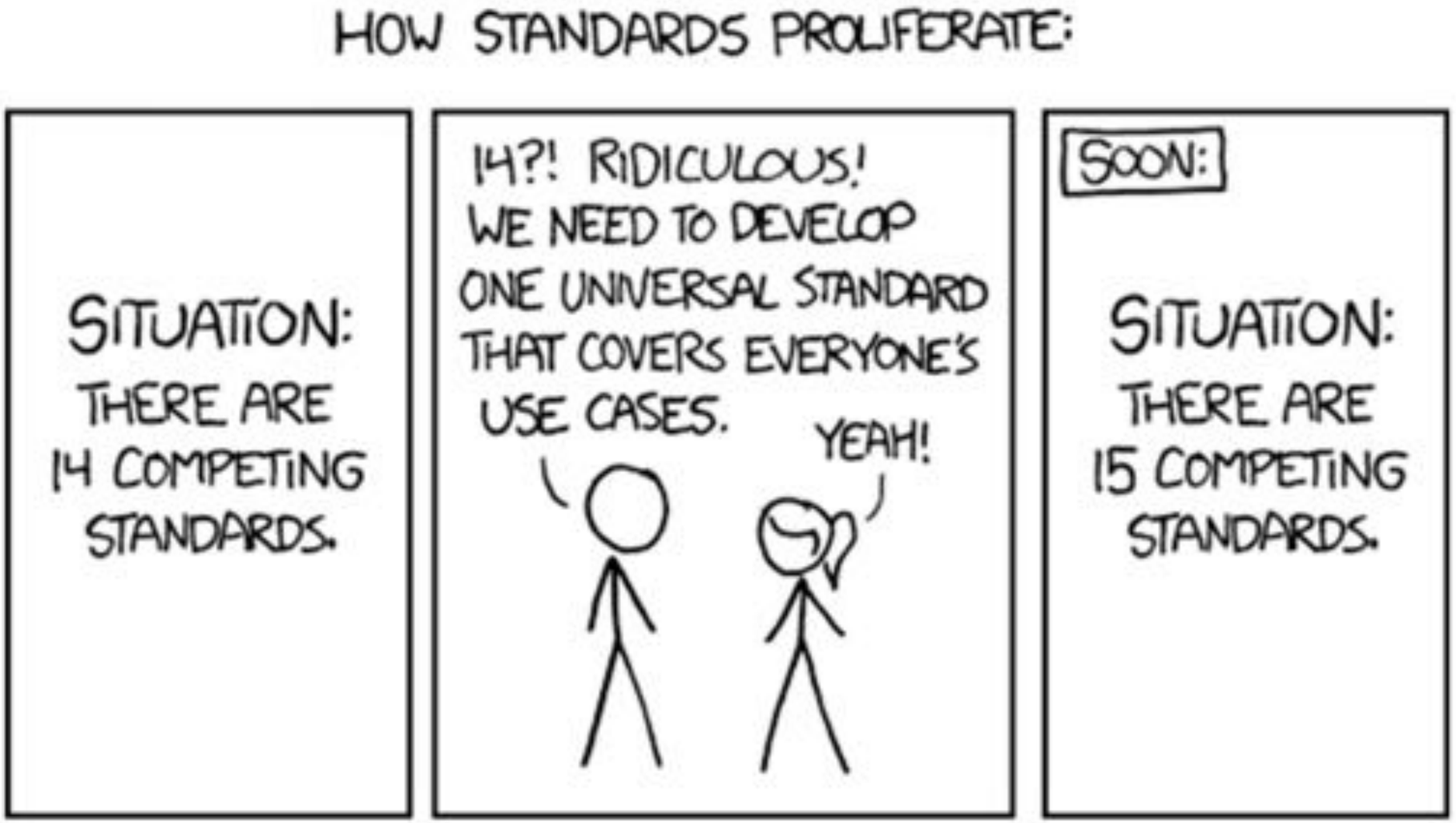
PathScale
InfiniPath (InfiniBand)
QLOGIC
TrueScale (InfiniBand)
intel
OmniPath

CRAY
Crossbar
Seastar
Gemini
Aries
Slingshot

Mellanox
InfiniBandSDR
DDR
QDR
FDR
EDR
HDR
NDR
XDR

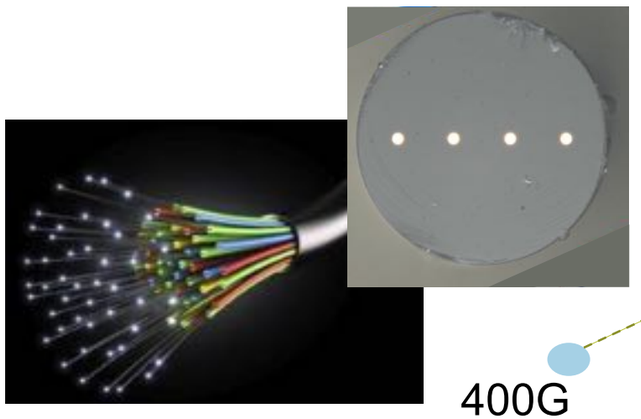
1995 2000 2005 2010 2015 2020 2025

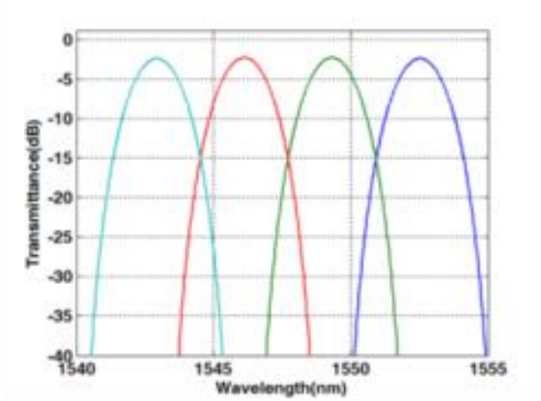
Fundamentals of Phy


HOW STANDARDS PROLIFERATE:
SITUATION:
THERE ARE
14 COMPETING
STANDARDS.
14?! RIDICULOUS!
WE NEED TO DEVELOP
ONE UNIVERSAL STANDARD
THAT COVERS EVERYONE'S
USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE
15 COMPETING
STANDARDS.

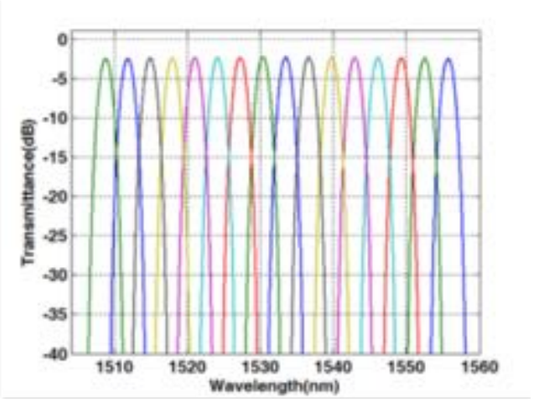
Future of Interconnect Speeds and Modulation

Speeds and Feeds

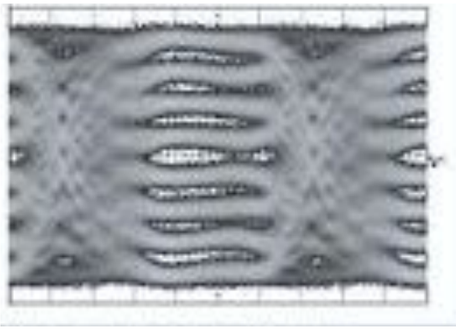




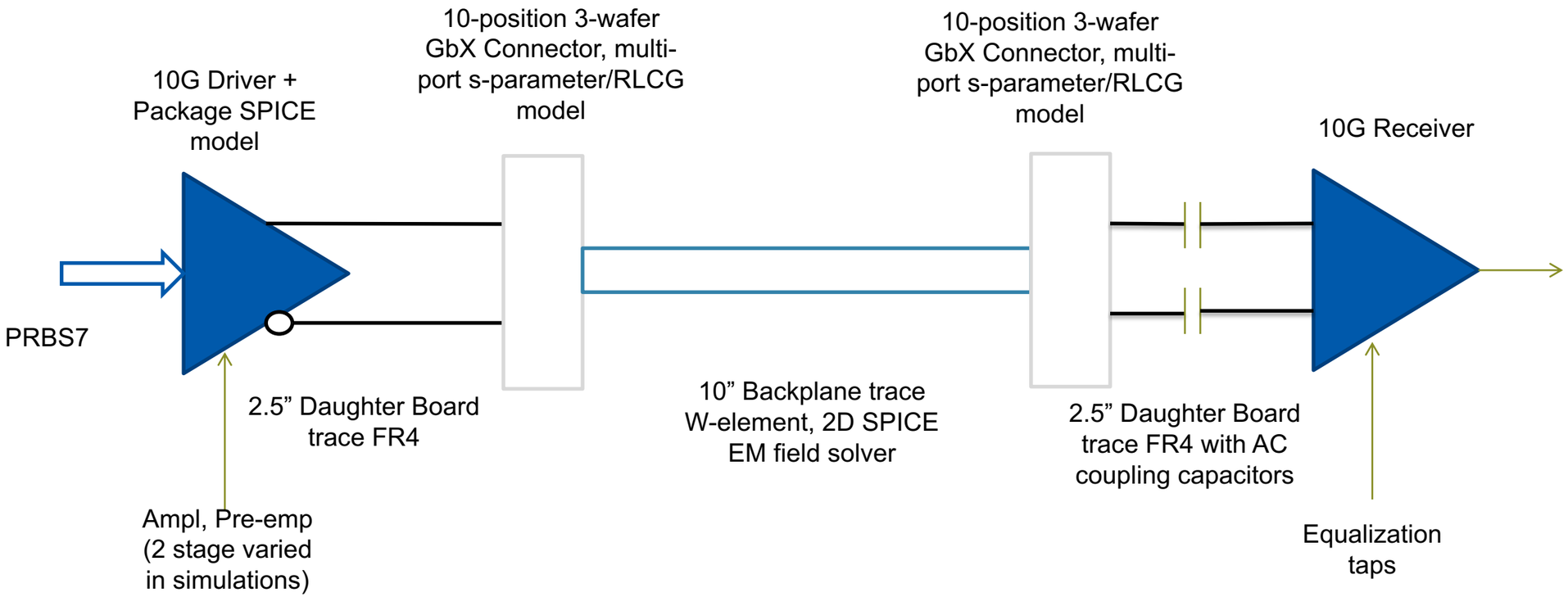
4x25Gbps, channel spacing: 400GHz



16x25Gbps, channel spacing: 400GHz



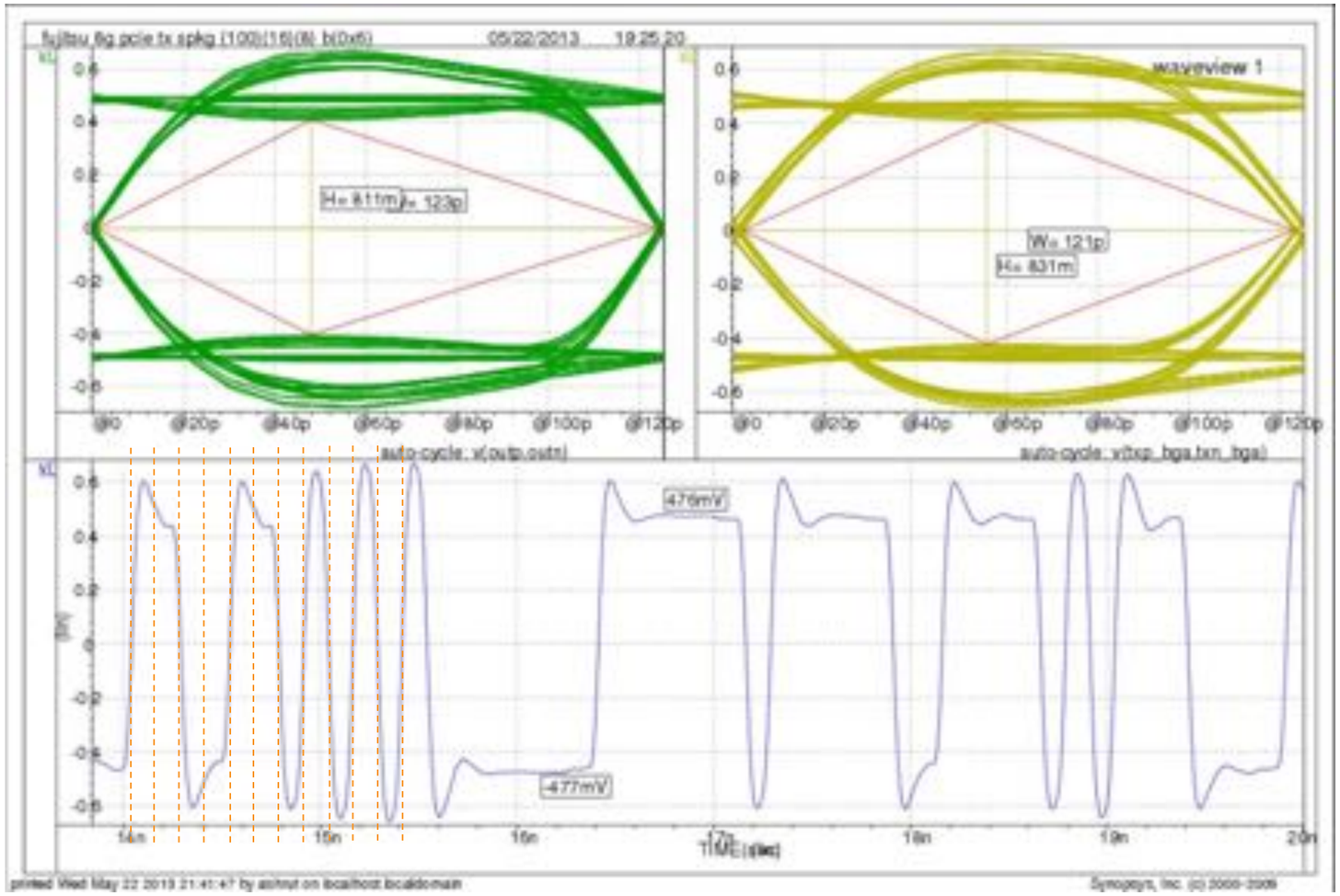
Block Diagram of an Example High-speed Channel



(1) The Eye Diagram

- For high-speed signaling, the most fundamental unit of measure or visual-aid is the eye-diagram
- Nothing but a representation of 0's and 1's in a "folded" time space.
- X-axis represents 1-bit period
- Y-axis represents waveform amplitude
- But this does not look very much like an eye….

PCIe Gen3 eye-diagram at o/p buffer (before and after package model)

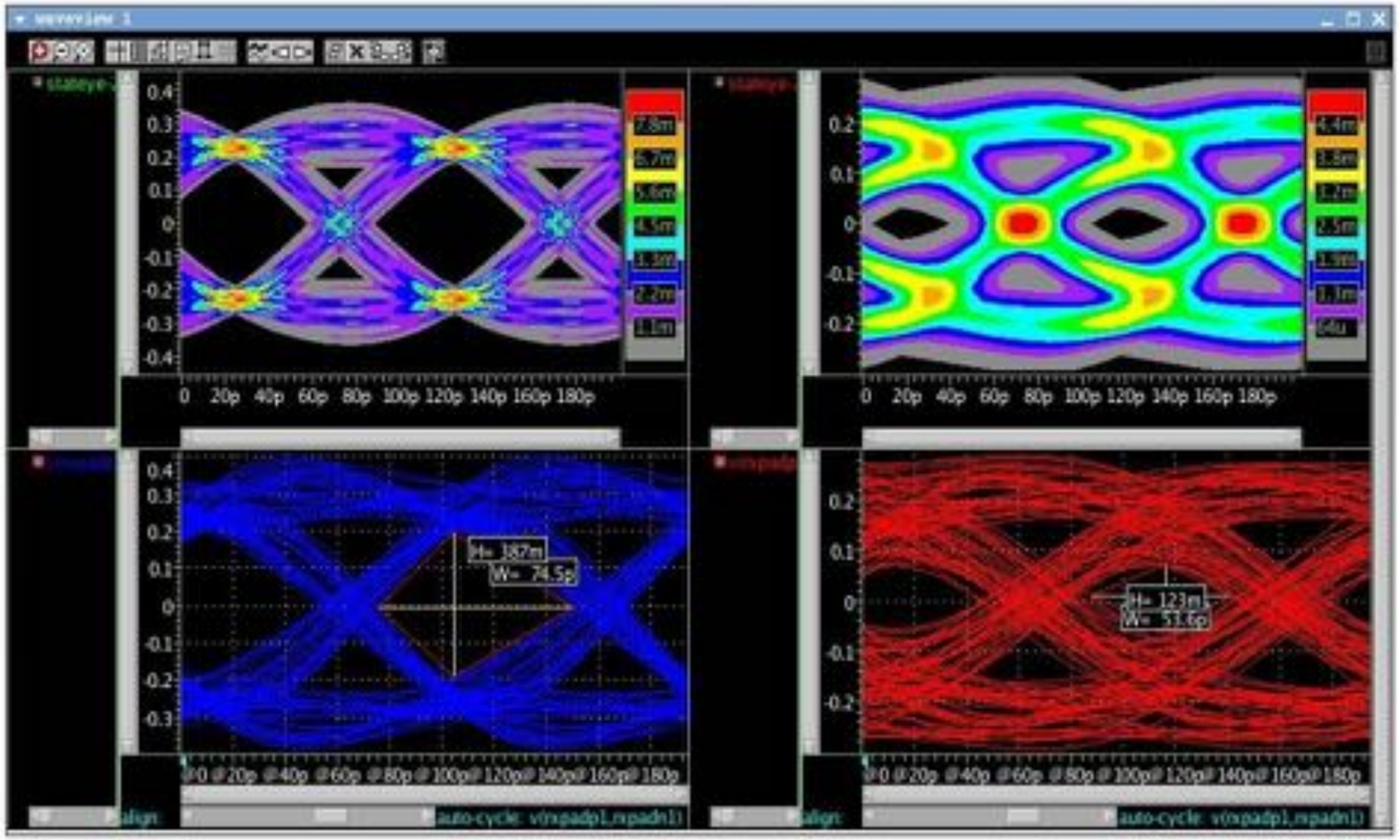


(1) The Eye Diagram cont..

- Maybe now it resembles a bit more..
- Characteristics like amplitude, T_r/T_f, zero-crossing etc. of the transmitted bit pattern starts to change as the signal traverses the channel
- What receiver sees is a very degraded bit-pattern. Eye seems to be “closing”
- What causes this eye to close?

10G eye diagram (before and after “channel”)

What was transmitted | **What the Receiver sees**



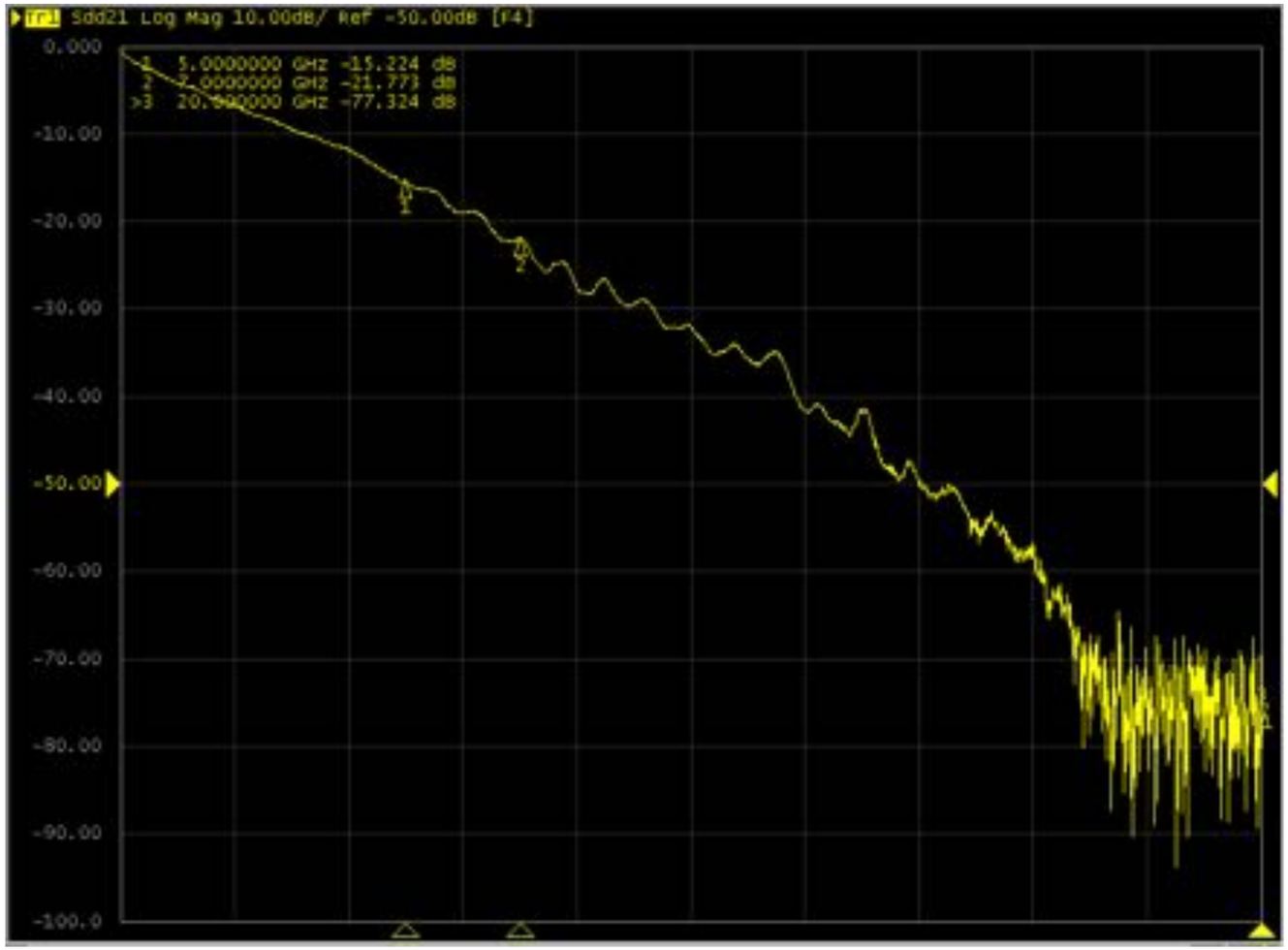
(2) Insertion Loss or Sdd21

Without going into the incident/reflected power and scattering matrix of 2-port networks.

Insertion loss can be simply defined as $20*\log_{10}$ of “forward voltage attenuation”

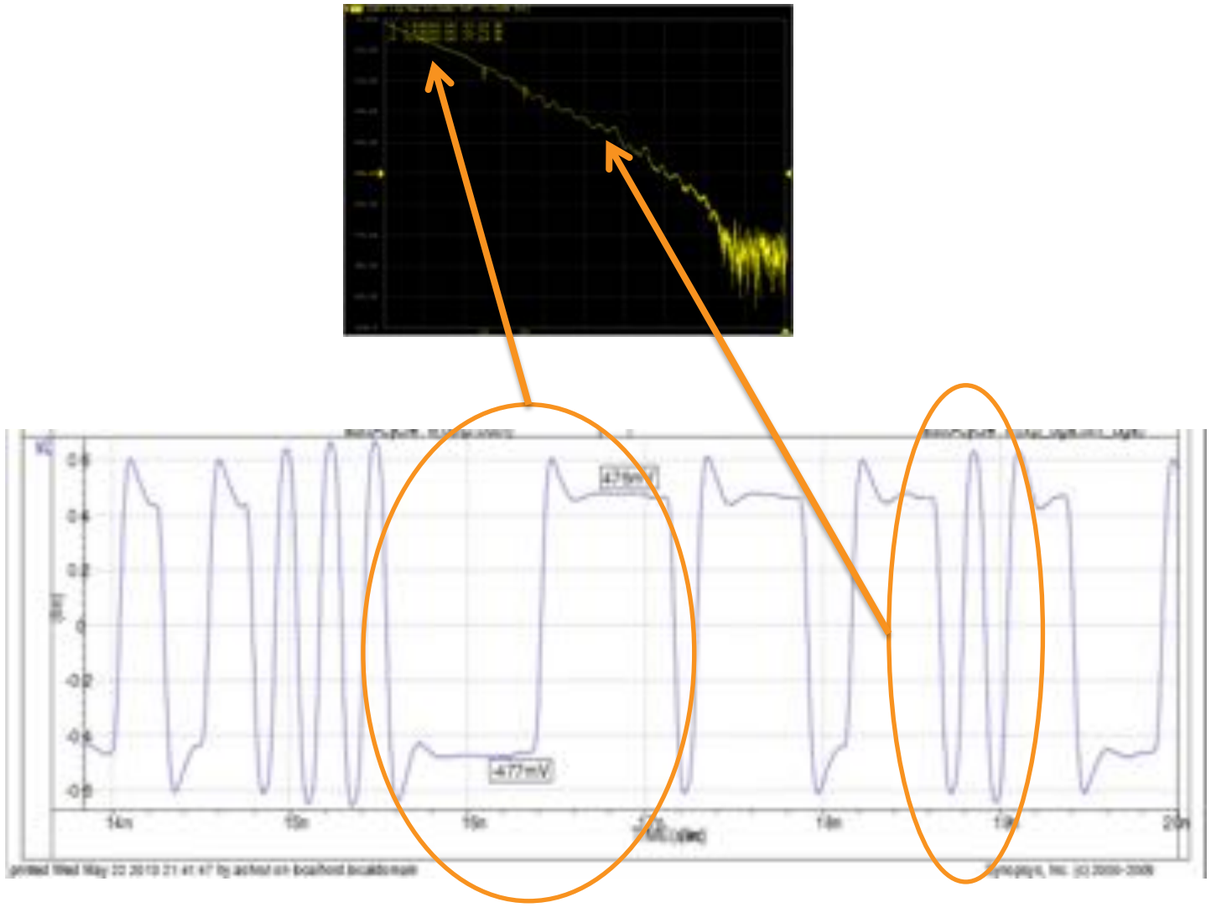
Freq sweep from DC-3rd harmonic

$$IL_{(dB)} = 20*\log(Vout/Vin)$$



(2) Insertion Loss or Sdd21 cont..

- IL starts increasing as frequency increases



Will face lesser attenuation

Will be attenuated more

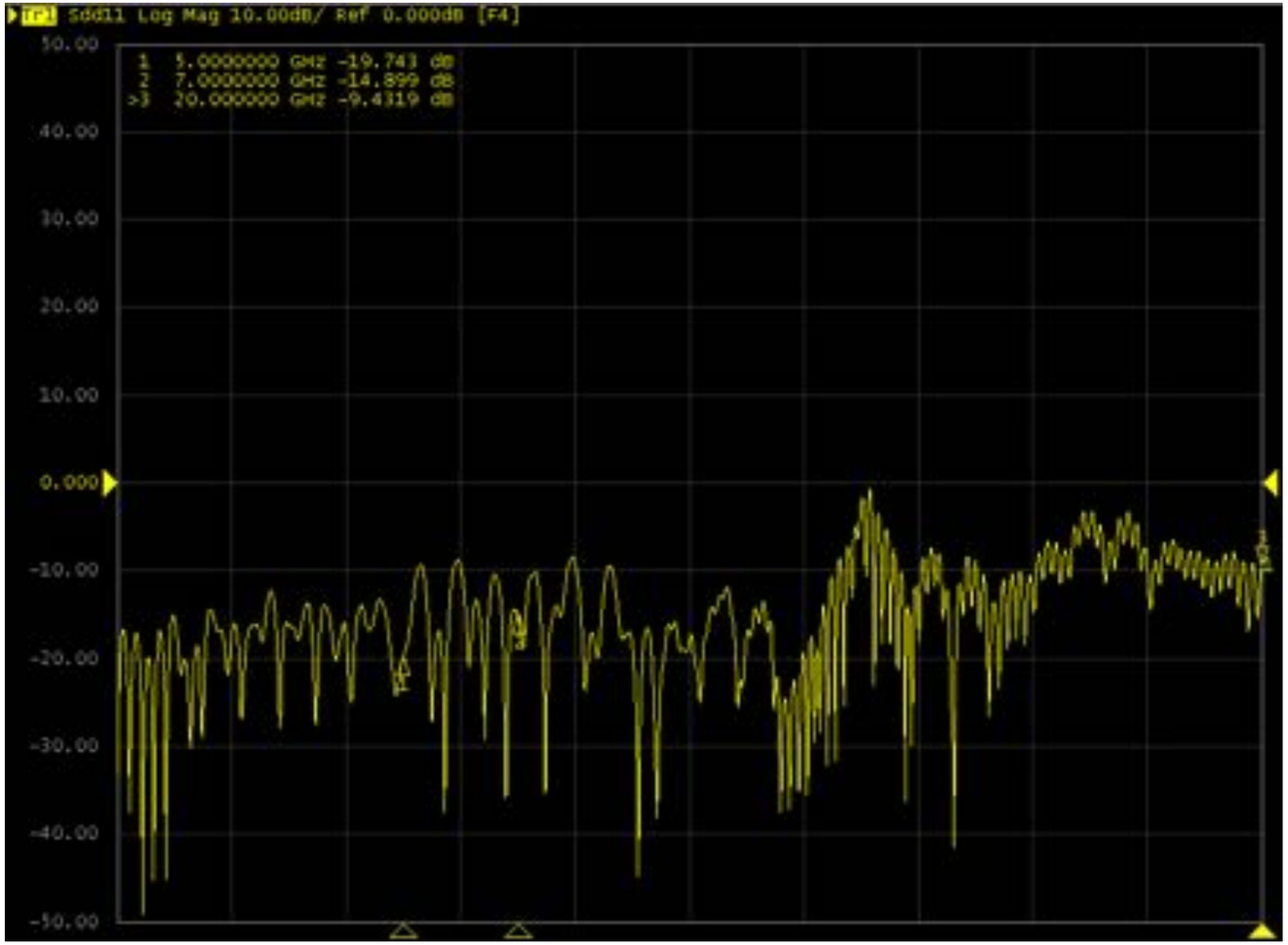
- IL is simply related to copper trace length and channel dielectric material properties
- Longer the channel (trace) length higher the loss

Cross-section view of a Transmission line

- Dielectric's ability to polarize/de-polarize causes freq dependence of Insertion Loss

(3) Return Loss or Sdd11

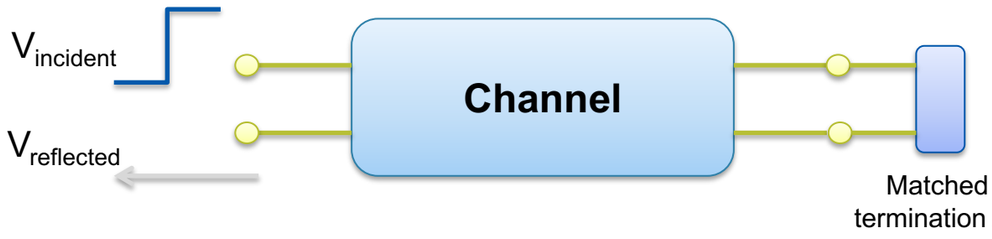
- Return loss in simple terms is the ratio of Reflected Power to the Incident Power on a dB scale
- Sdd11 like Sdd21 is again a frequency domain plot
- RL of -20dB means 10% of incident power was reflected back
- Why was it reflected?
- In simple terms, due to Impedance mismatch/discontinuities

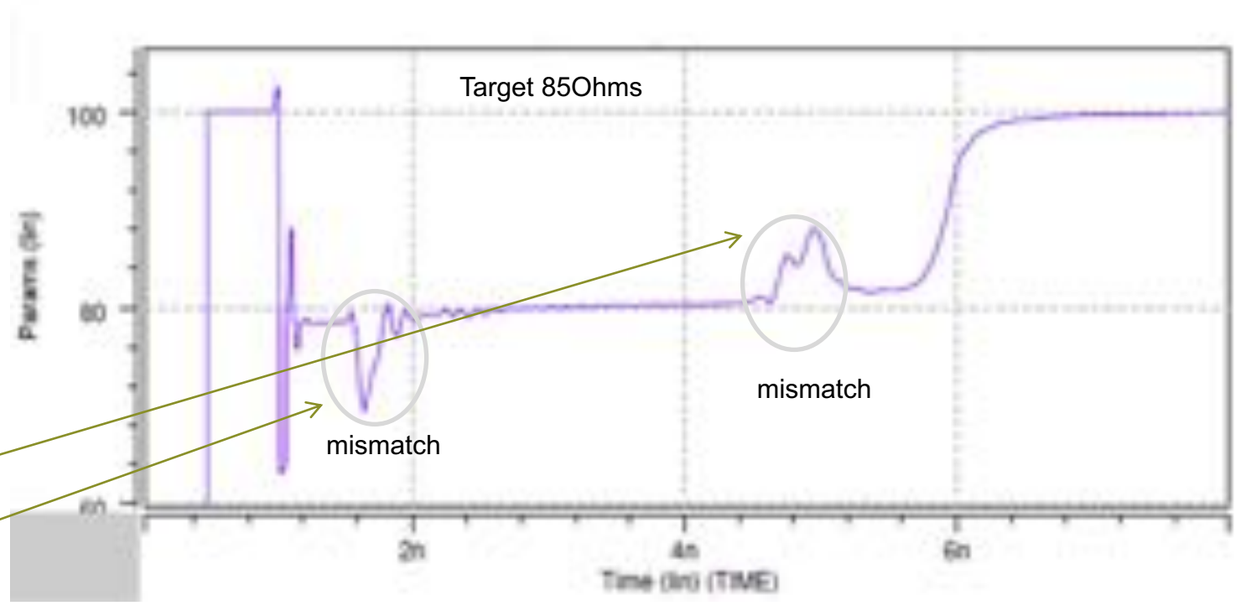


(4) Impedance Profile and TDR

- Time Domain Reflectometry (TDR) is the most fundamental method for measuring impedance variations across a channel
- Launch a fast rising pulse and measure the reflected wave
- $TDR_{(Z)} = 2*Z_0 \frac{(V_{incident}+V_{reflected})}{(V_{incident}-V_{reflected})}$
- From Transmission-line perspective

$$Z_0 = \sqrt{\frac{R+j\omega L}{G+j\omega C}}$$

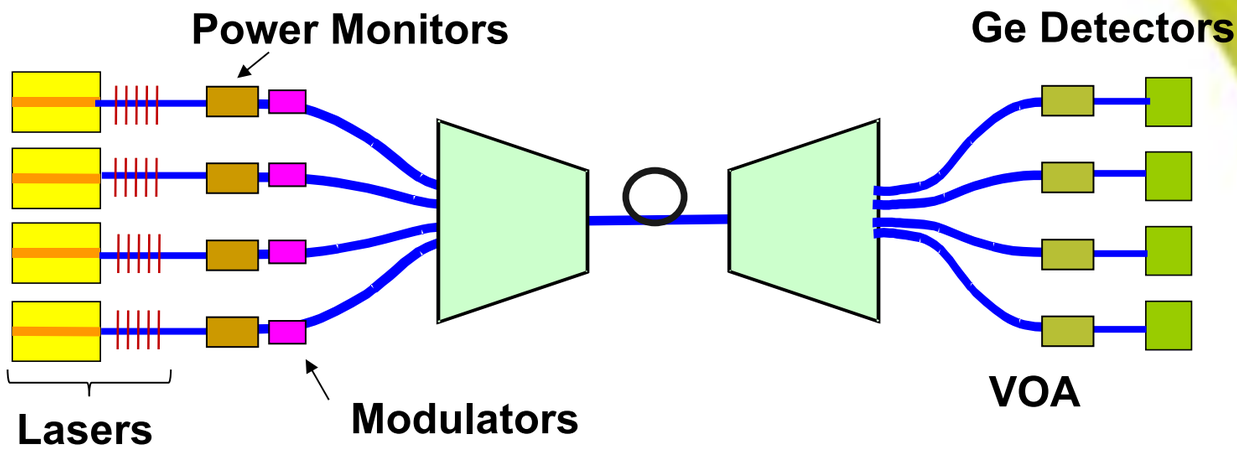


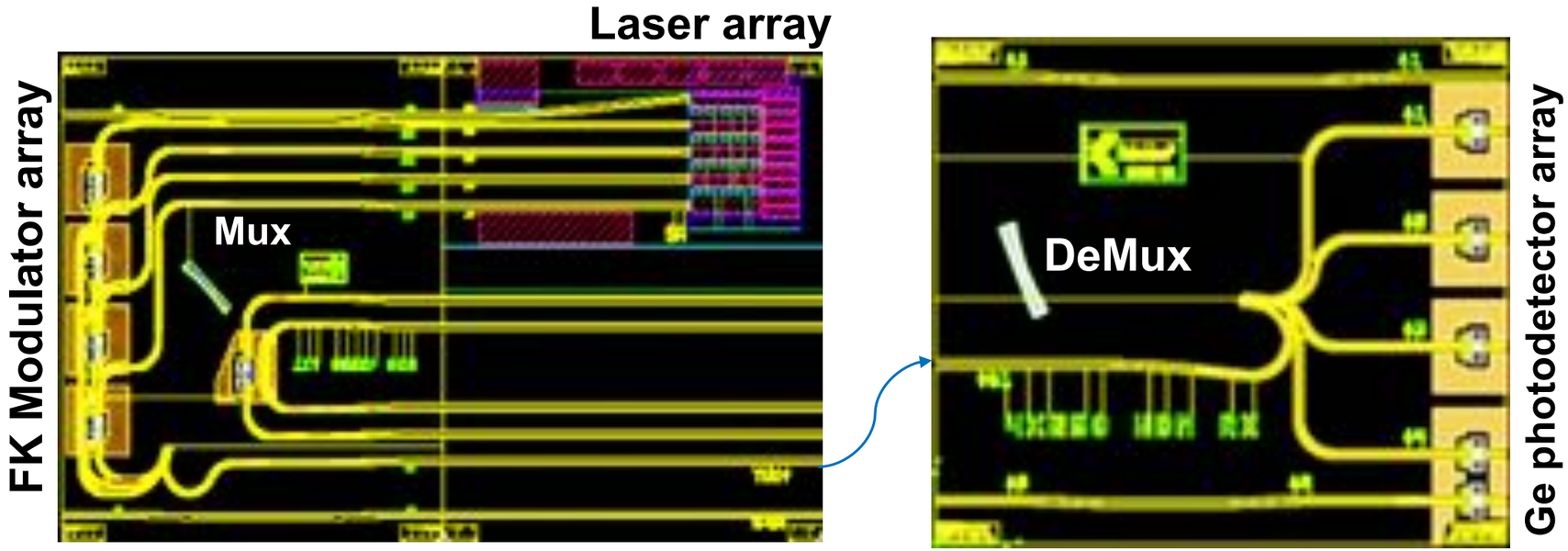


Measure Impedance profile of a channel comprising vias and connectors

Enter Optics.. VCSEL or Photonics

- 4x25G WDM on SMF28
- 4 dB link budget (Channel Insertion Loss)
- WDM laser solution based on low-cost external cavity lasers
- Fully integrated single chip Tx and Rx solutions





3.2nm spacing channel plan

Next Generation Supercomputers with HDR 200G Infiniband

(Examples)

TACC TEXAS ADVANCED COMPUTING CENTER

23.5 Petaflops
8K HDR InfiniBand Nodes
Fat-Tree Topology

KMA Korea Meteorological Administration

50 Petaflops
7.2K HDR InfiniBand Nodes
Dragonfly+ Topology

Australian National University

3K HDR InfiniBand Nodes
Dragonfly+ Topology

MISSISSIPPI STATE UNIVERSITY

3.1 Petaflops
1.8K HDR InfiniBand Nodes
Fat-Tree Topology

CSC

FINNISH METEOROLOGICAL INSTITUTE

1.7 Petaflops
2K HDR InfiniBand Nodes
Dragonfly+ Topology

Microsoft Azure

Highest Performance Cloud

SDSC SAN DIEGO SUPERCOMPUTER CENTER

PITTSBURGH SUPERCOMPUTING CENTER

筑波大学 University of Tsukuba

1.6 Petaflops
Hybrid CPU-GPU-FPGA
Fat-Tree Topology

Large Record-Breaking AI Systems

ImageNet training record breakers

facebook

P100 x 256, EDR InfiniBand

Scaling efficiency ~90 %

Preferred Networks

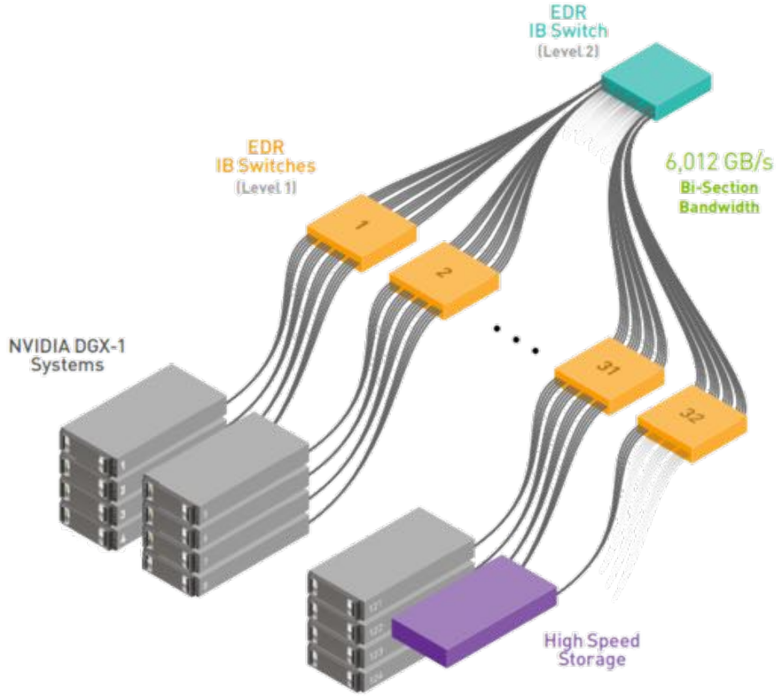
P100 x 1024, FDR InfiniBand

Scaling efficiency 80 %

SONY

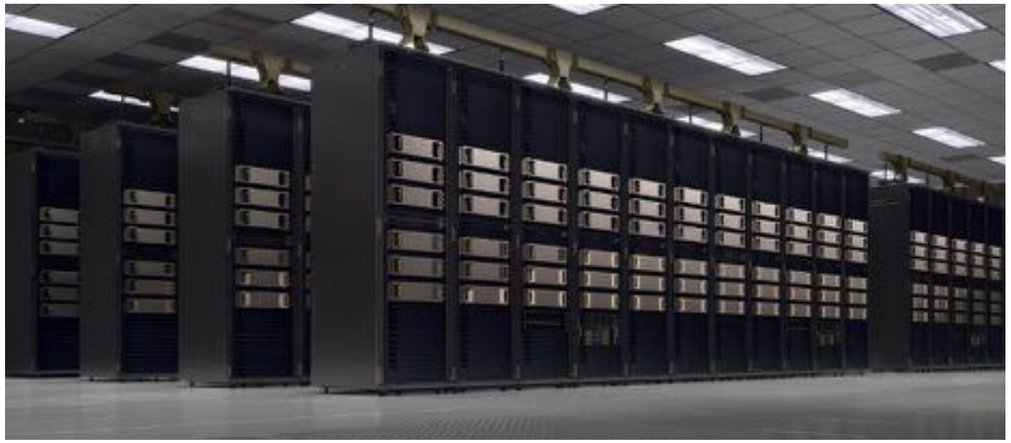
V100 x 1088, EDR InfiniBand

Scaling efficiency 91.62 %



NVIDIA DGX SATURNV

- 124 DGX-1 nodes interconnected by 32 L1 TOR Switches, in 2016
- Mellanox 36 port EDR L1 and L2 switches, 4 EDR per system
- Upgraded to 660 NVIDIA DGX-1 V100 Server Nodes, in 2017
- 5280 V100 GPUs, 660 PetaFLOPS (AI)



HDR Latency

Copper Cables

5ns per m

Optical Cables

5ns per m
+ 20ns*

5ns per m
+ 100ns

40p 1U Switch

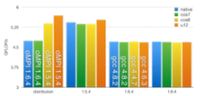
<90ns

800p Chassis Switch

<300ns

ConnectX-6 B2B

600ns



MPI OSU latency

800ns

* - Q4 2018

Need to Accelerate All Levels of HPC / AI Frameworks

Application

- Data Analysis
- Real Time
- Deep Learning

Communication

- SHARP In-Network Computing
- MPI Tag Matching
- MPI Rendezvous
- Software Defined Virtual Devices

MPI · HPC-X™

UCF Unified Communication Framework

SHARP Scalable Hierarchical Aggregation and Reduction Protocol

Network

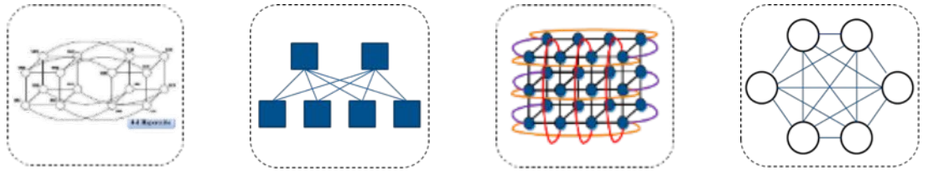
- Network Transport Offload
- RDMA and GPU-Direct RDMA
- SHIELD (Self-Healing Network)
- Enhanced Adaptive Routing and Congestion Control

RDMA GPUDirect

SHIELD

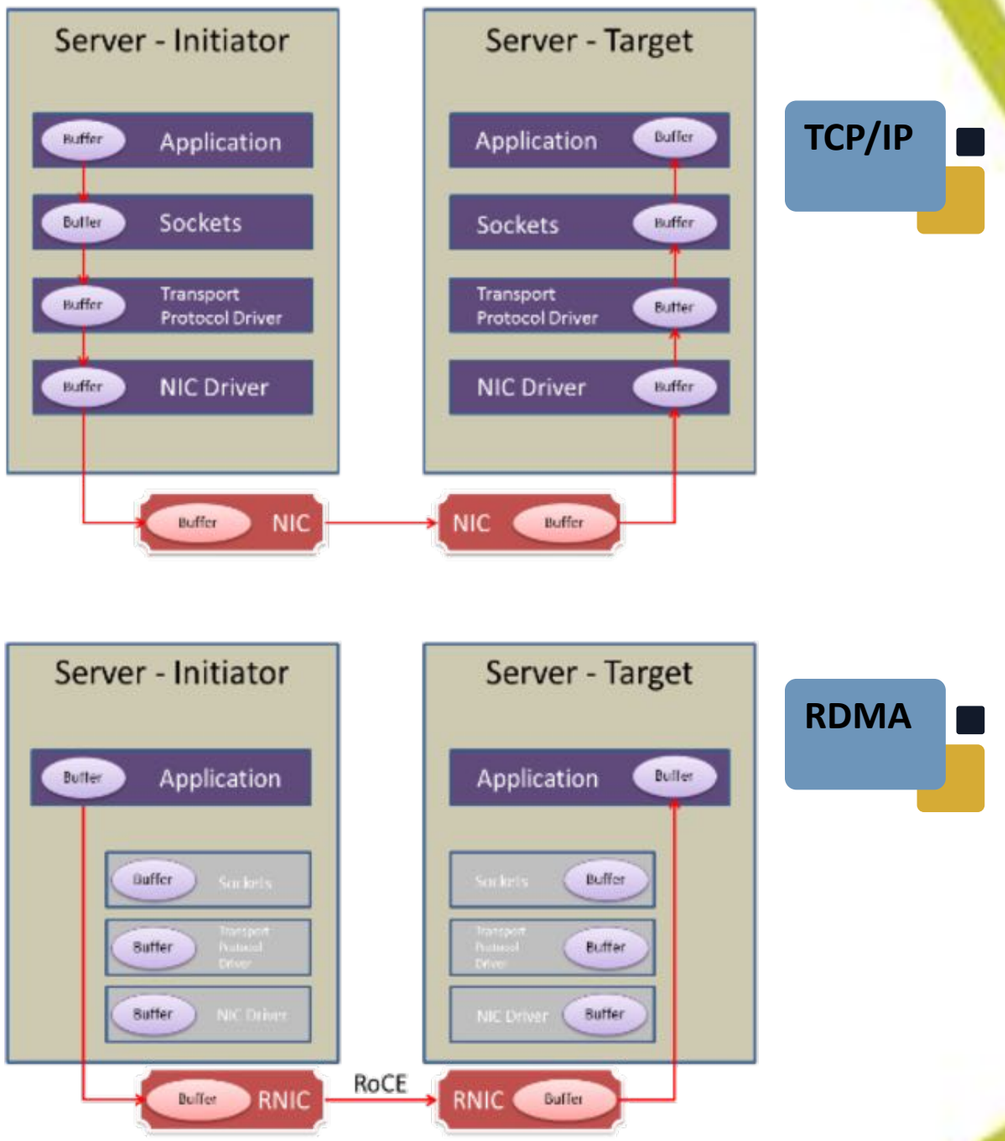
Connectivity

- Multi-Host Technology
- Socket-Direct Technology
- Enhanced Topologies



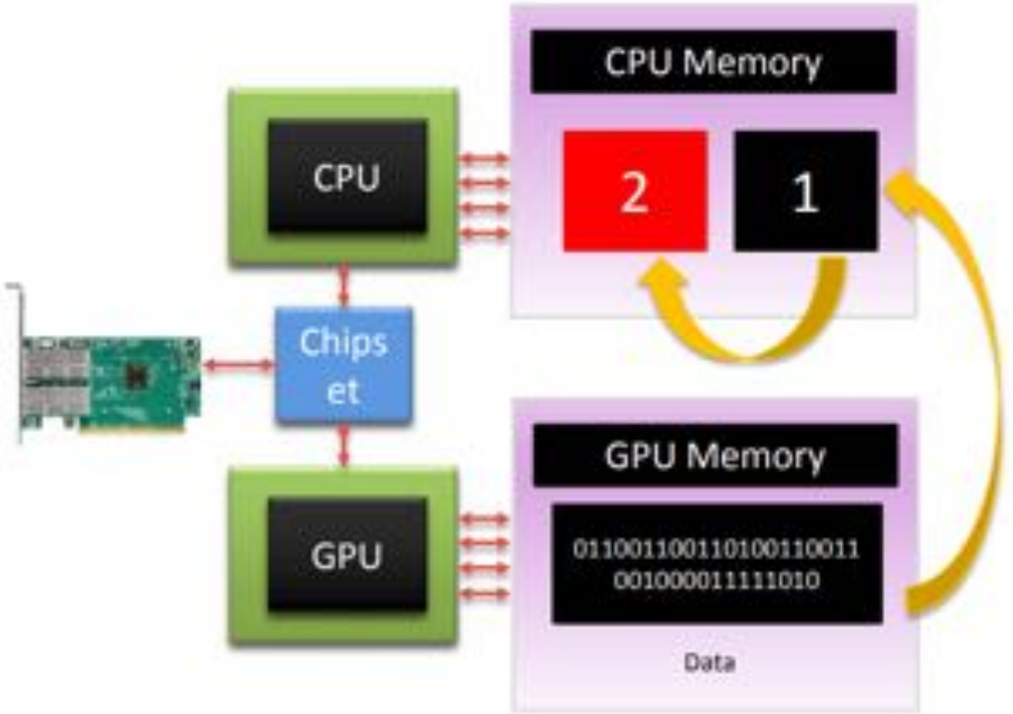
RDMA in SuperComputing Clusters

- Remote Direct Memory Access (RDMA)
- Advance transport protocol (same layer as TCP and UDP)
- Main features
  - Remote memory read/write semantics in addition to send/receive
  - Kernel bypass / direct user space access
  - Full hardware offload for network stack
  - Secure, channel based IO
- Application Advantage
  - Lowest latency
  - Highest bandwidth
  - Lowest CPU consumption
- RoCE: RDMA over Converged Ethernet
  - Available for all Ethernet speeds 10 – 100G
- Verbs: RDMA SW Interface (Equivalent to Sockets)



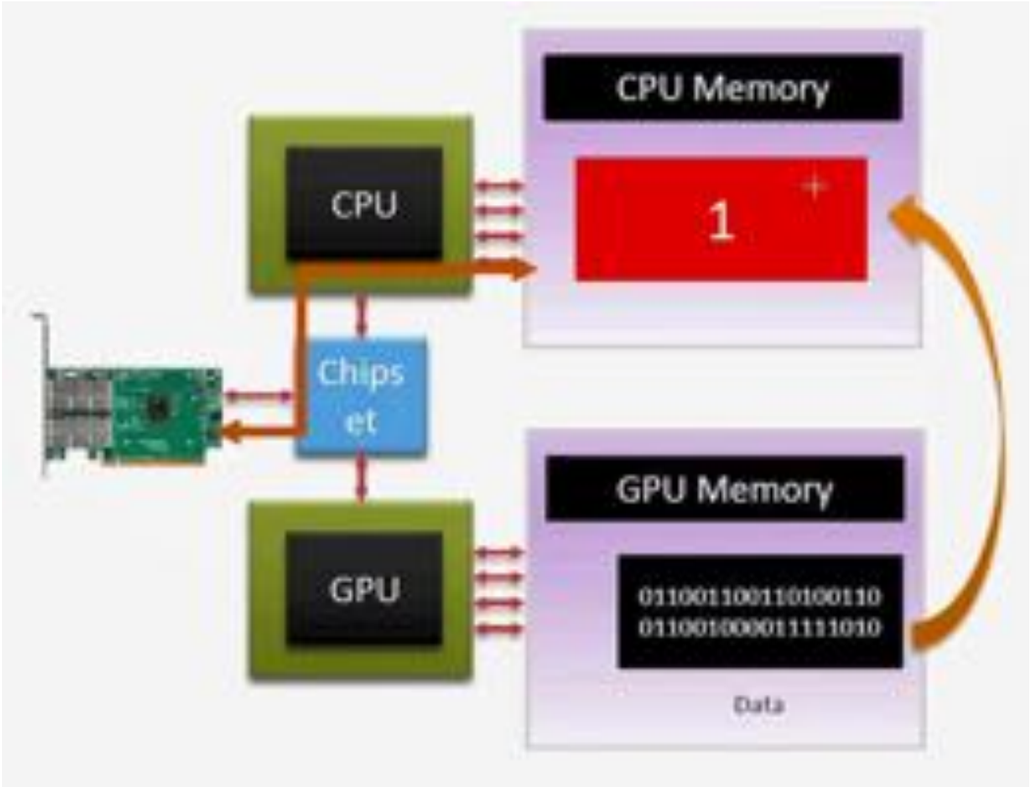
GPUDirect RDMA Evolution

- **Prior to GPUDirect**
  - GPU use driver-allocated pinned memory buffer
  - RDMA use pinned buffers for zero-copy kernel-bypass communication
- Impossible for RDMA drivers to pin memory allocated by GPU
- Two copies
  - GPU copies data from GPU internal memory to GPU driver system pinned memory (1)
  - User space needs to copy data between the GPU driver system pinned memory (1) and RDMA system pinned memory (2)
  - RDMA device sends data to network



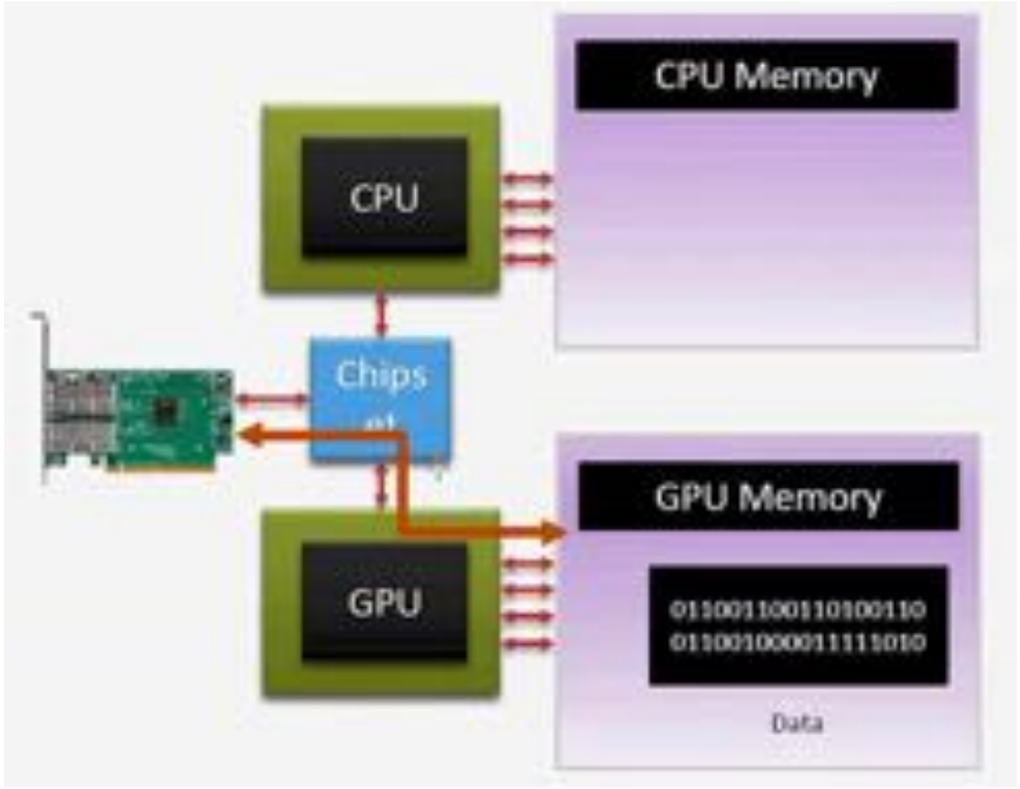
GPUDirect RDMA Evolution (cont.)

- **GPUDirect / GPUDirect P2P** (Peer-to-Peer)
  - GPU and RDMA devices share the same pinned memory buffer
- One copy
  - GPU copies data from GPU internal memory to system pinned memory (1)
  - RDMA device sends data to network



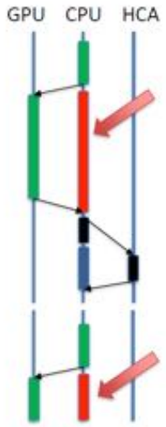
GPUDirect RDMA Evolution (cont.)

- **GPUDirect RDMA**
  - GPU memory is exposed to RDMA NIC
  - Direct data path from GPU to network - Data path is zero copy
  - **CPU** is involved in the control path - WQE preparation, ring doorbell, handles completions for incoming packets to GPU
- Zero copy
  - RDMA device sends data to network from GPU memory
  - RDMA device receive data from network to GPU memory
- The CPU still synchronizes between GPU tasks and data transfers




```
while(fin) {
    gpu_kernel <<<... , stream>>>(buf);
    cudaStreamSynchronize(stream);
    ibv_post_send(buf);
    ibv_poll_cq(cqe);
}
```

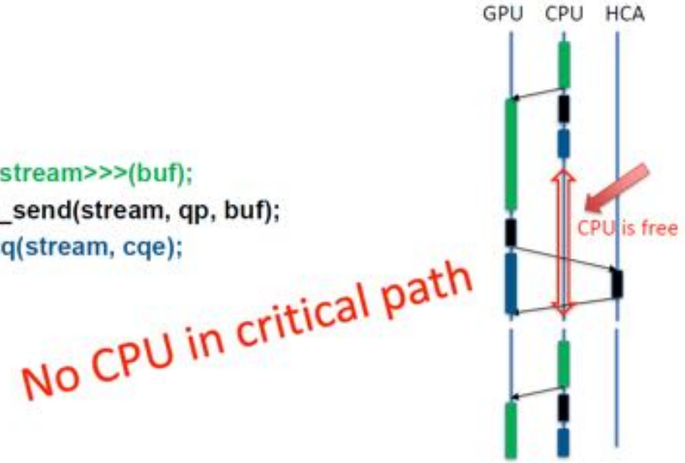
100% CPU Utilization

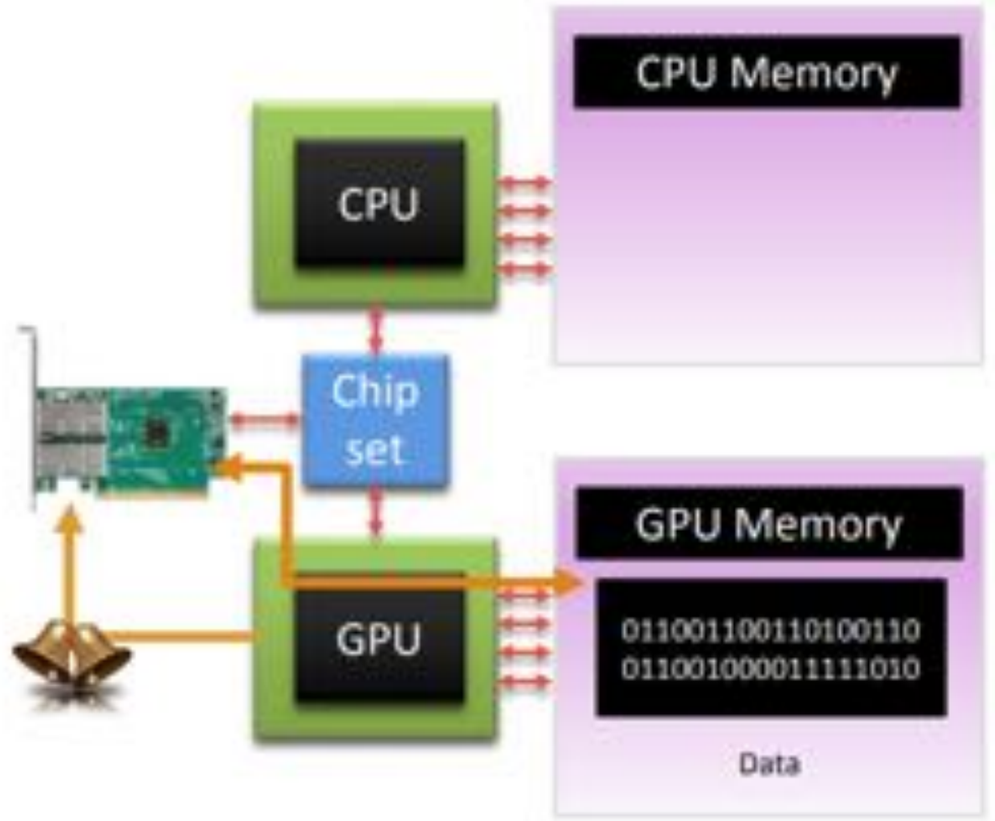


GPUDirect RDMA Evolution (cont.)

- **GPUDirect RDMA Async**
  - GPU memory is exposed to RDMA NIC
  - Direct data path from GPU to network - Data path is zero copy
  - **CPU** is involved in WQE preparation and release completed WQEs
  - **GPU** is involved in Ring Doorbell, Handles completions for incoming packets to GPU
- Zero copy
  - RDMA device sends data to network from GPU memory
  - RDMA device receive data from network to GPU memory
- Reduce CPU utilization

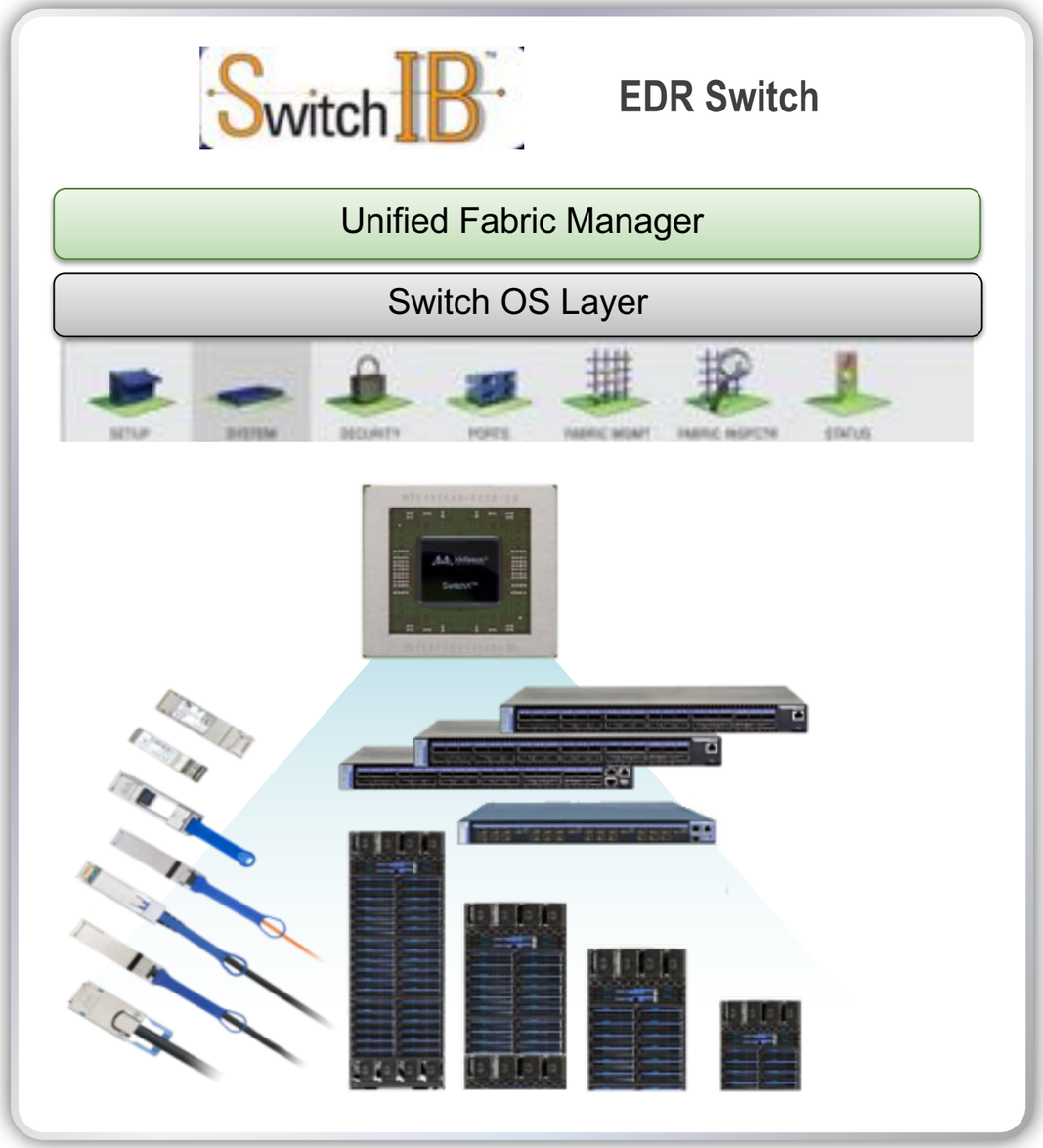
```
while(fin) {
    gpu_kernel <<<... , stream>>>(buf);
    gds_stream_queue_send(stream, qp, buf);
    gds_stream_wait_cq(stream, cqe);
}
```





Lets Build Supercomputers

The Building Blocks



36 ports IB up to 100 Gb/s

~90ns port-to-port latency

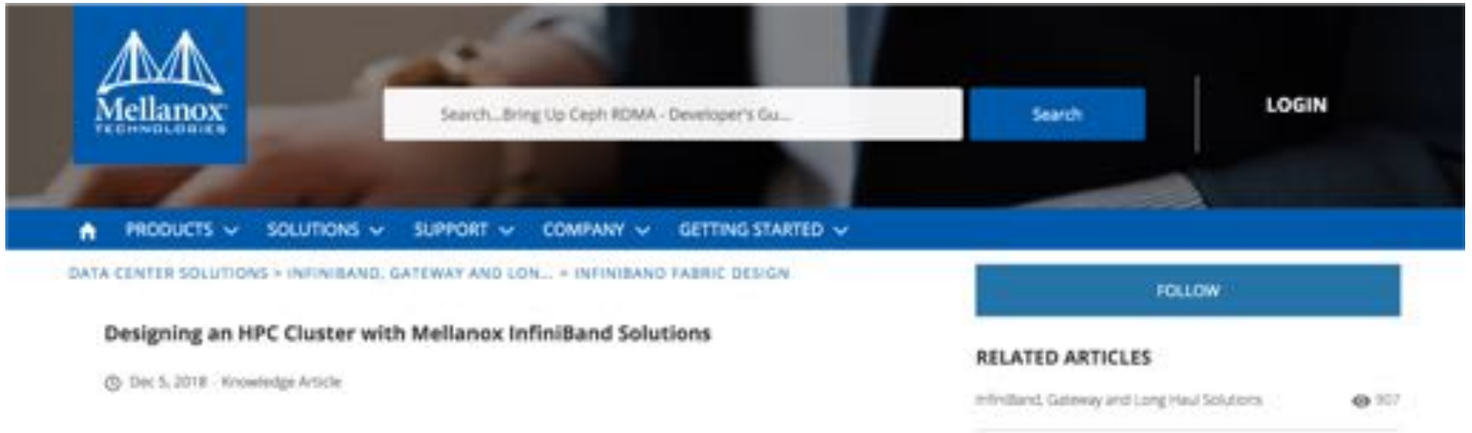
Full-crossbar

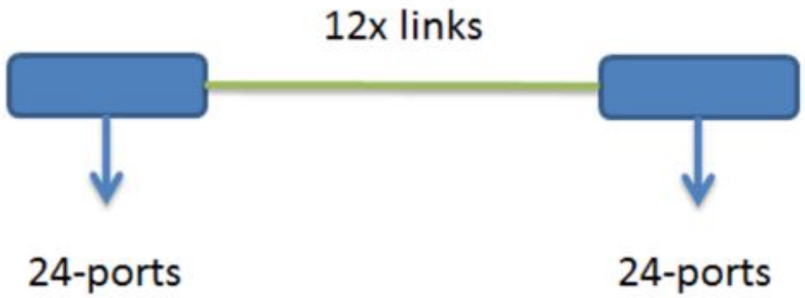
SwitchIB

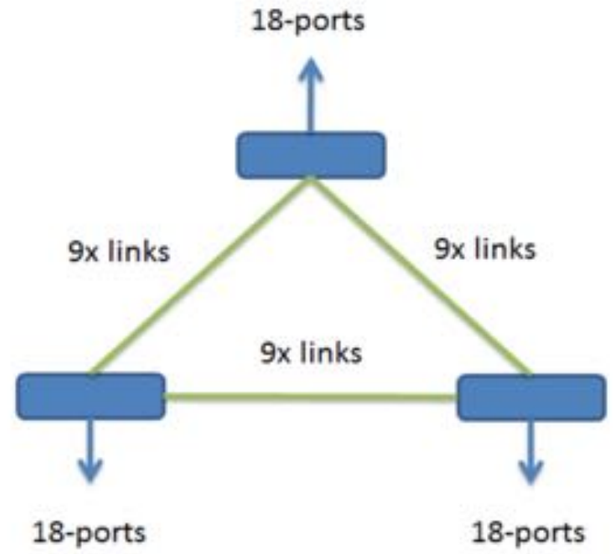
SB7700

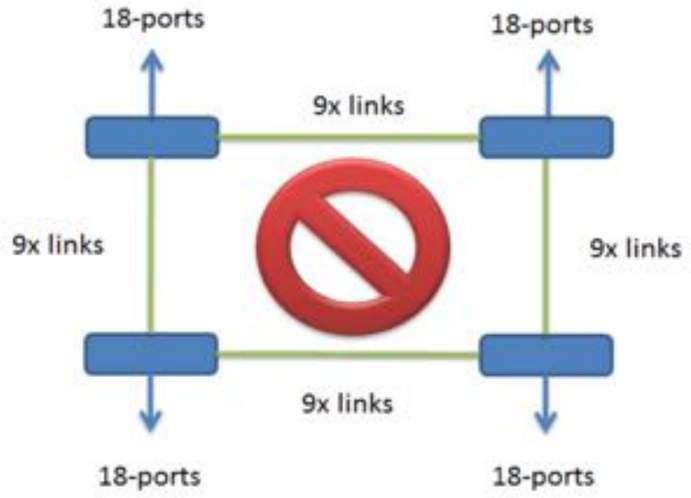
- 36 EDR (100Gb/s) ports
- Enhanced Management
  - MLNX-OS™
  - Embedded Subnet Manager (648 nodes)
  - Chassis management
- Fabric Inspector™
- UFM™ (Basic/Advanced)

Designing Small Infiniband Clusters





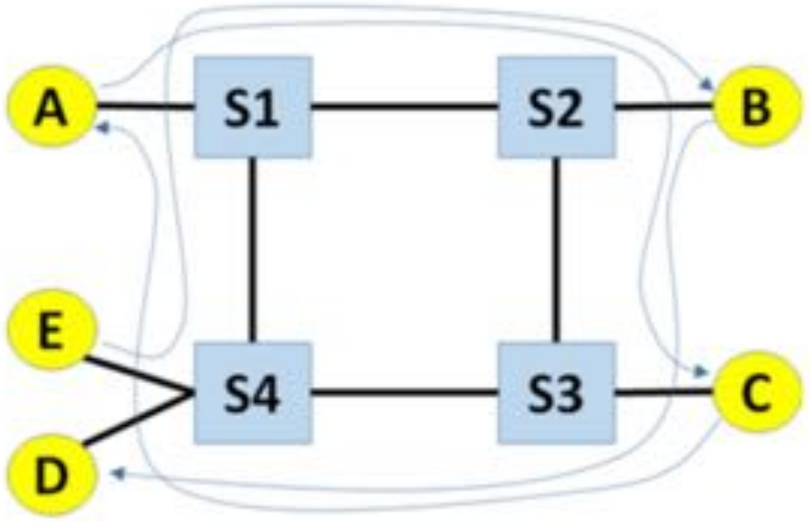




Credit Loops

Like many other networks, **InfiniBand dislikes loops**. Specifically, it dislikes **logical loops** where link back pressure can create a deadlock situation. These are called **credit loops**. Although the HOQ timers periodically clear such a deadlock, performance suffers. A credit loop only represents a potential deadlock, which depends on the traffic at each link in the loop. However, at InfiniBand speeds such a deadlock can occur very quickly given the right traffic pattern.

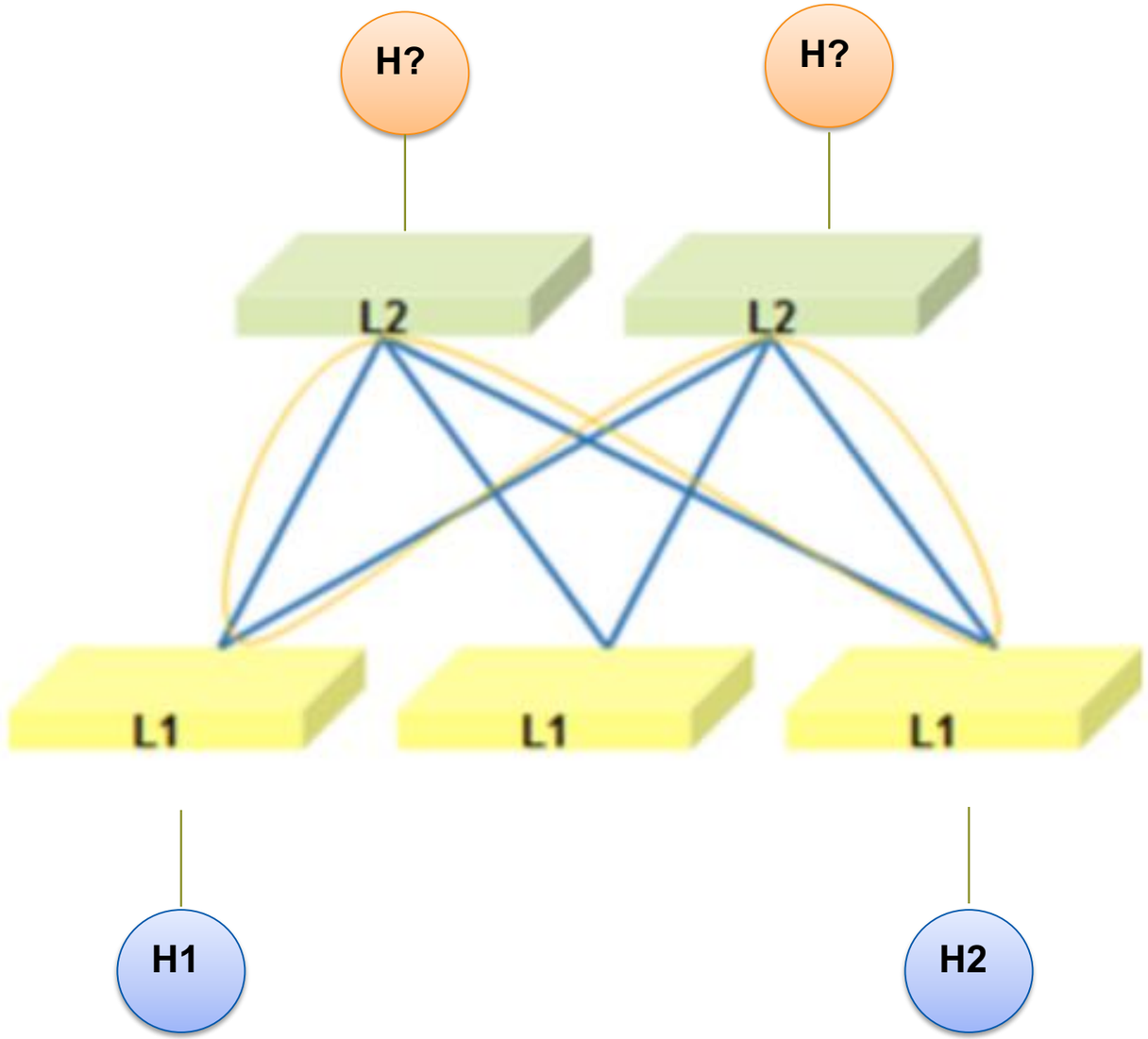
The following diagram illustrates a very simple credit loop.



In this example:

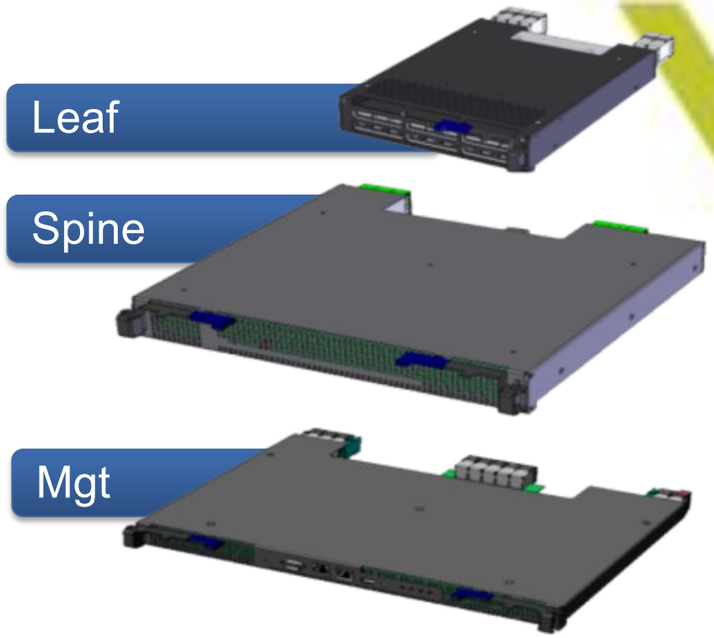
- Four Switch ASICs are connected in a ring topology.
- A host adapter (HCA) connects a server (Node) to each Switch, with an additional HCA on Switch 4.

Credit Loops Cont..



- Connecting Storage or Hosts to the Spines??
- A good physical topology, including good node placement
- Better routing algorithms
- Multiple Virtual Lanes
- Best practice **is not to attach servers to L2 Switches**

Director Switches – Modular Design



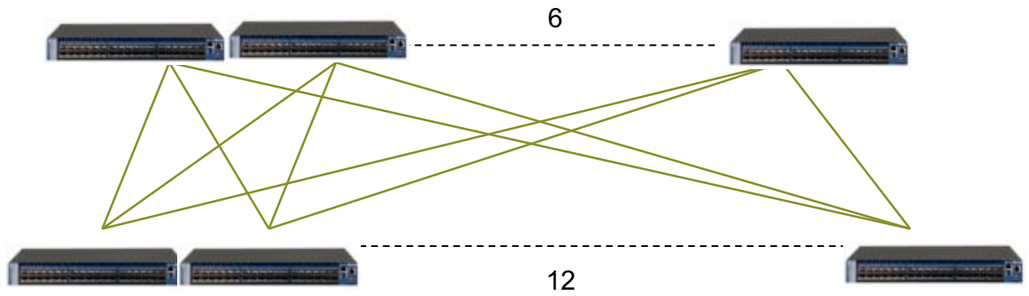
- Common leafs, spines, power supplies
- Common fans:
  - 108 & 216 ports switches
  - 324 & 648 ports switches
- Unique chassis:
  - Can be partially populated with leafs
  - Non-blocking or over subscribed

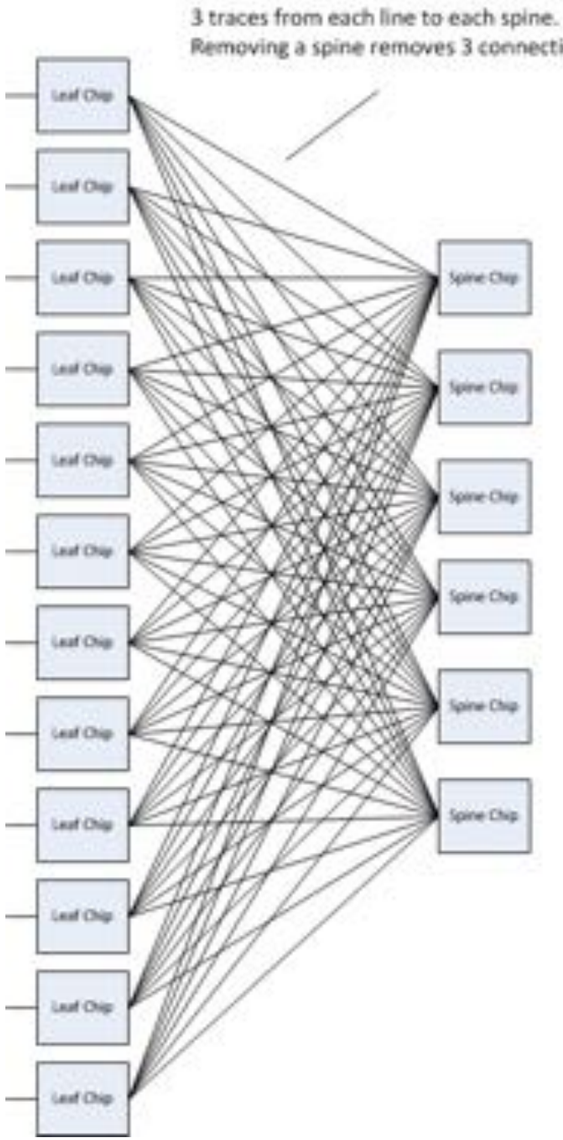
- High Availability
  - Redundant fans, power supplies, and management module
- All cables at the back
  - Cable management included
- N+N power supply redundancy

Internals of a Director Class Switch

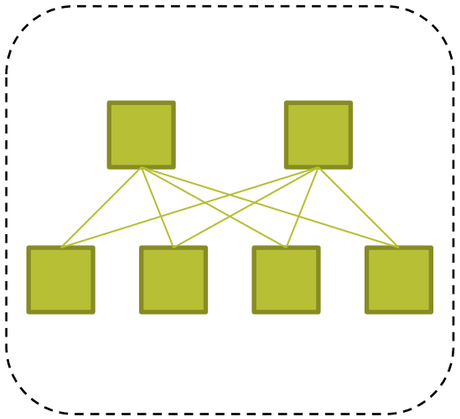
- 18 ports facing out per leaf
- 18 ports ‘internally’
- Each leaf spreads those 18
  - 3x3x3x3x3x3 to a 36 port internal spine chip

$$k(k/2)^{n-1}$$

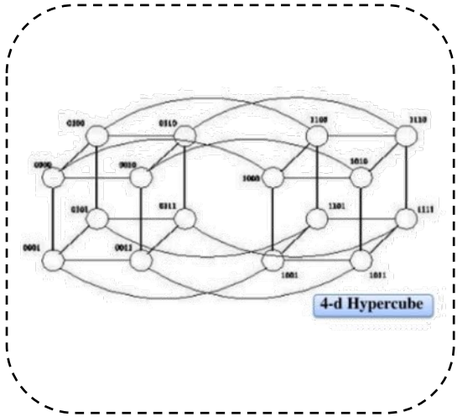




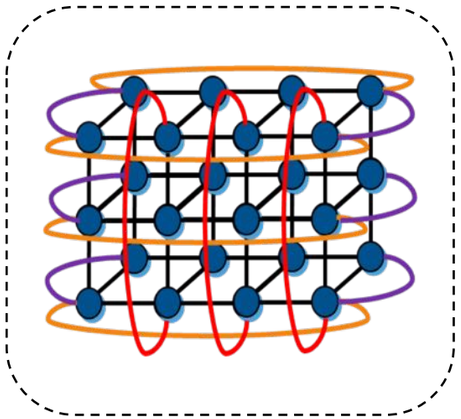
Lets Talk Some Network Topologies



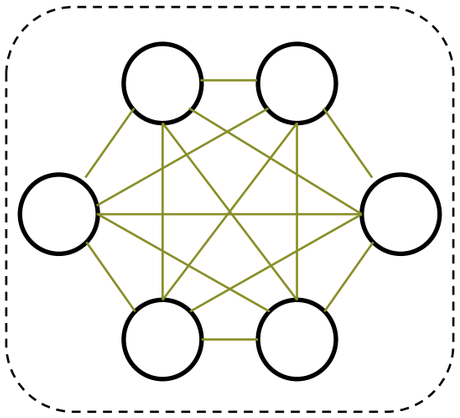
Fat Tree



Hypercube

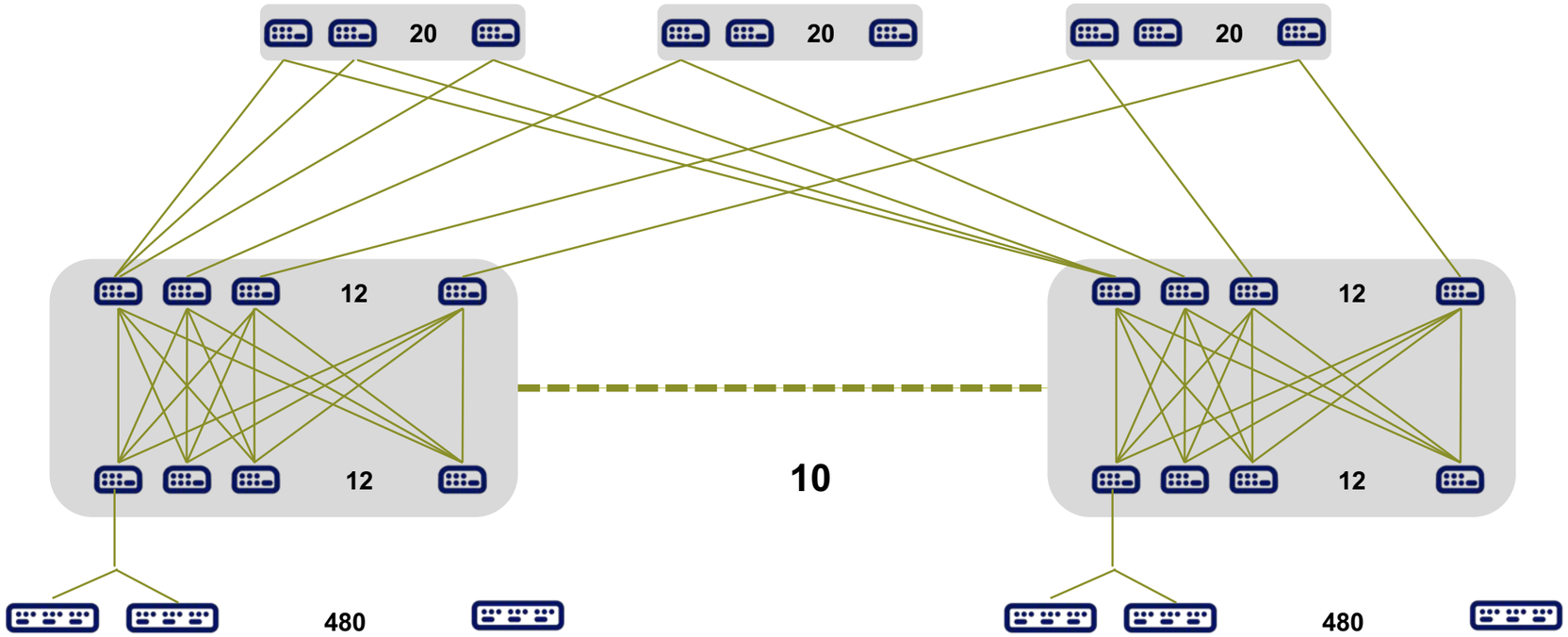


Torus



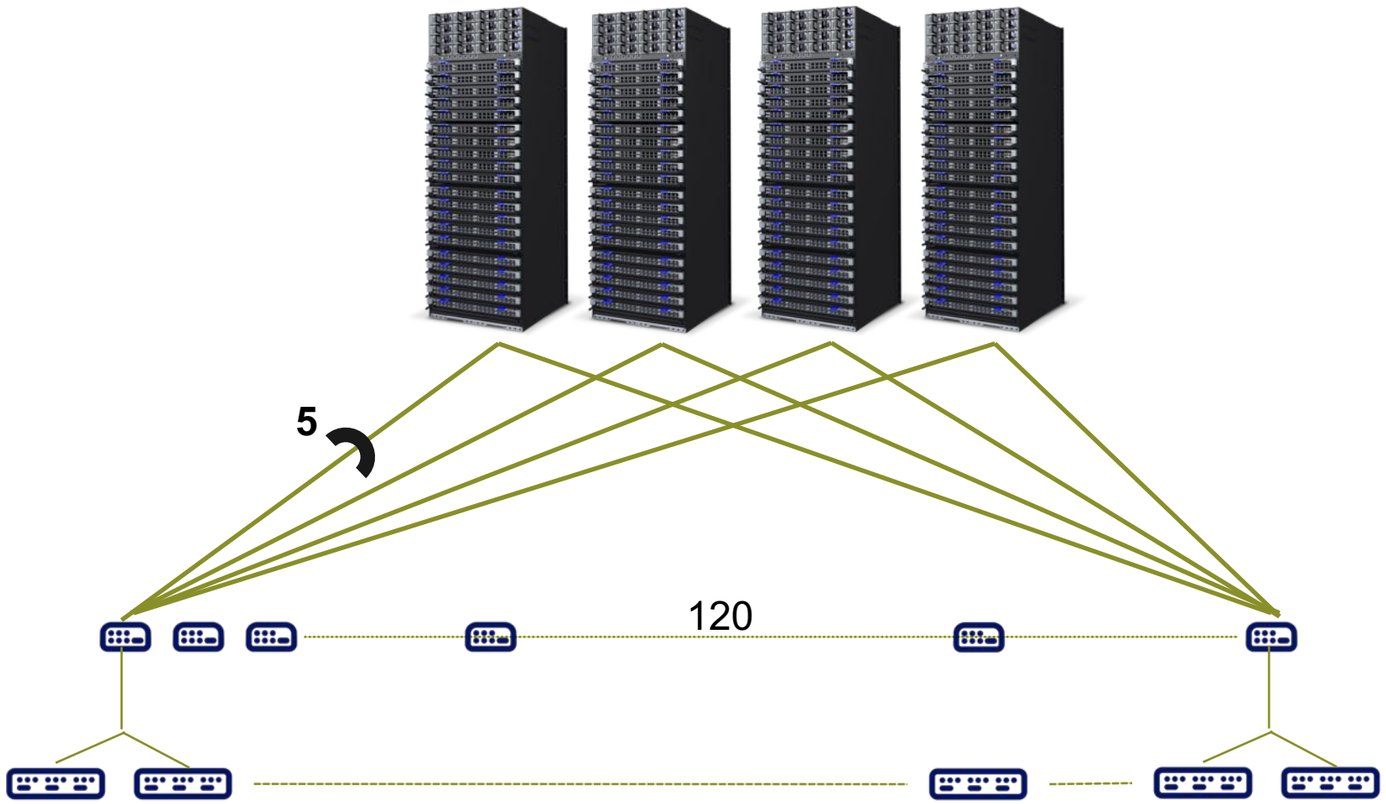
Dragonfly

3-level Clos with 1U HDR switches



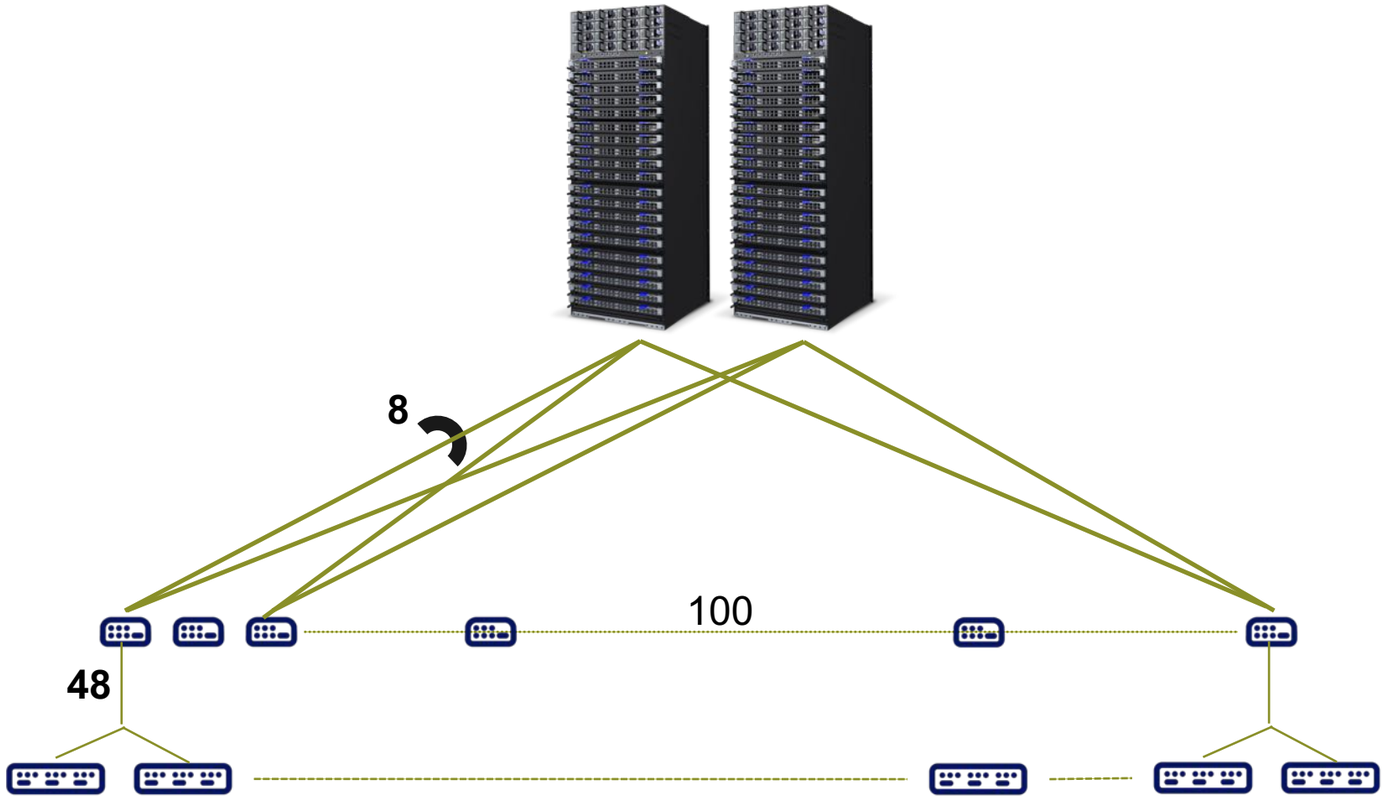
- **300 ASICs**
- 40 x 12 = 480 node sub-cluster (S1-S10)
- 10 Sub-clusters = 4800 port system
- Expansion requires re-design
- 5-hop network across any node-pair

3-level traditional Clos; chassis switches



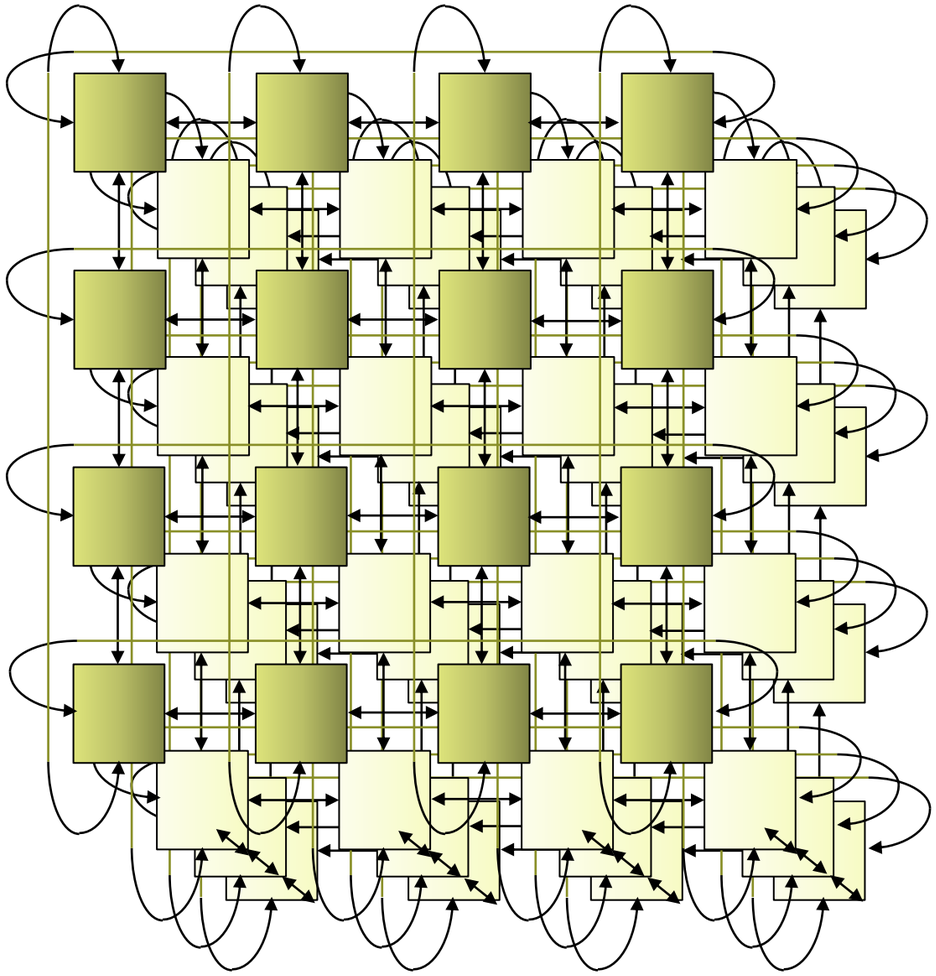
- **360 ASICs**
- 1:1 Fully non-blocking HDR100
- 4800 servers
- Max system size 6400x 100G ports
- 5-hop network across any node-pair
- 6x 3-hop sub-clusters with 800 nodes per sub-cluster

3-level traditional Clos; chassis switches {blocking}

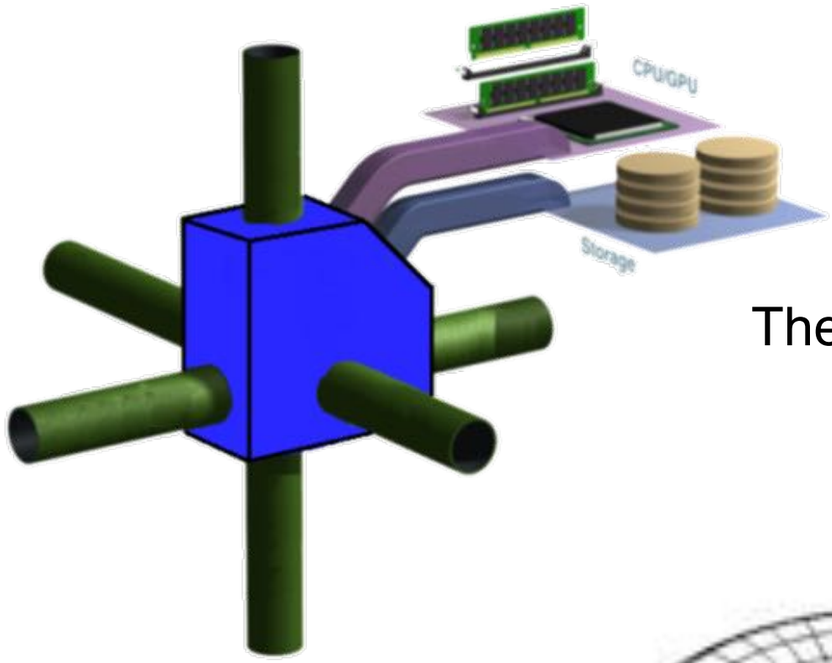


- 220 ASICs
- 1:1.5 Blocking HDR100
- 4800 servers
- Max system size 4800x 100G ports
- 5-hop network across any node-pair
- 4x 3-hop sub-clusters with 960 nodes per sub-cluster

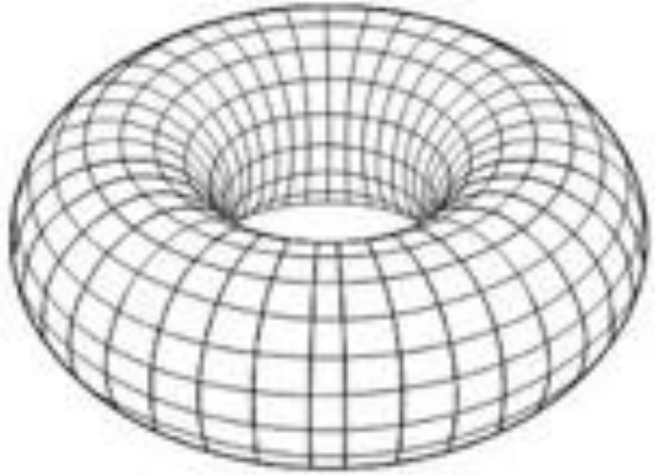
The 3-D Torus



A 4x4x4 Torus



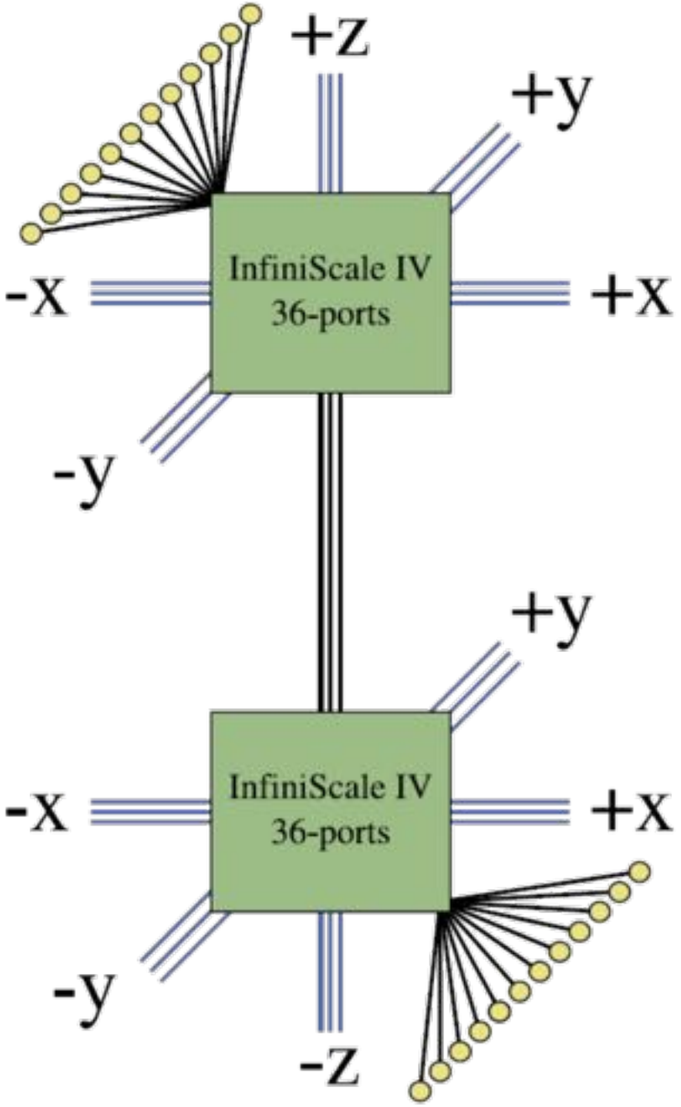
The Building Block



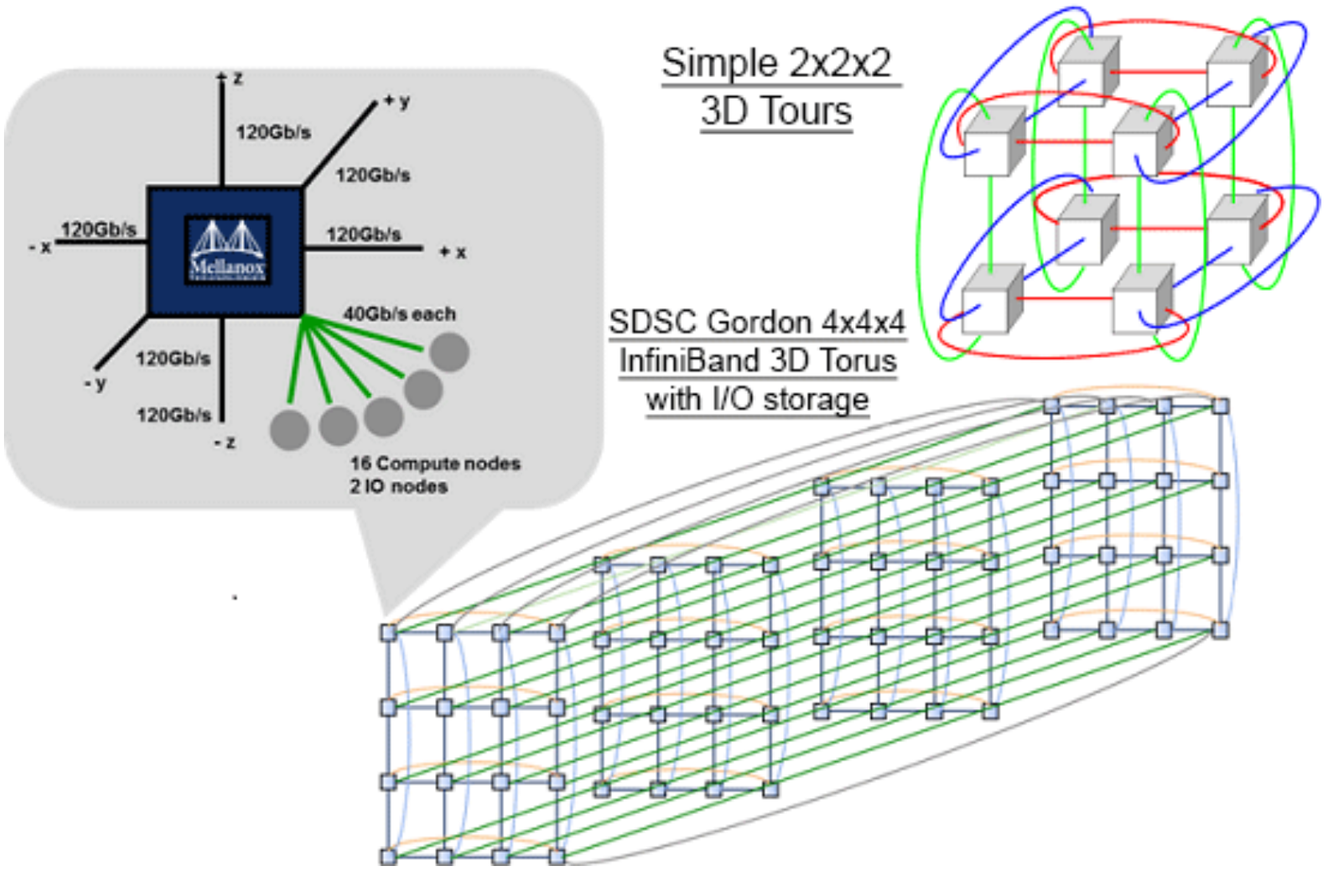
Natural “folding” to create a Torus

Creating the Torus with Infiniband Switches

- Allows for linear expandability without re-cabling
- Simple wiring using short cables
- Works well for localized communication
- Lower costs, power and cooling with fewer switches and cables
- Ability to connect storage into each of the cube junctions
- Fault tolerant with ability to handle multiple link and switch failures
- Built in support for adaptive routing, congestion control, QoS

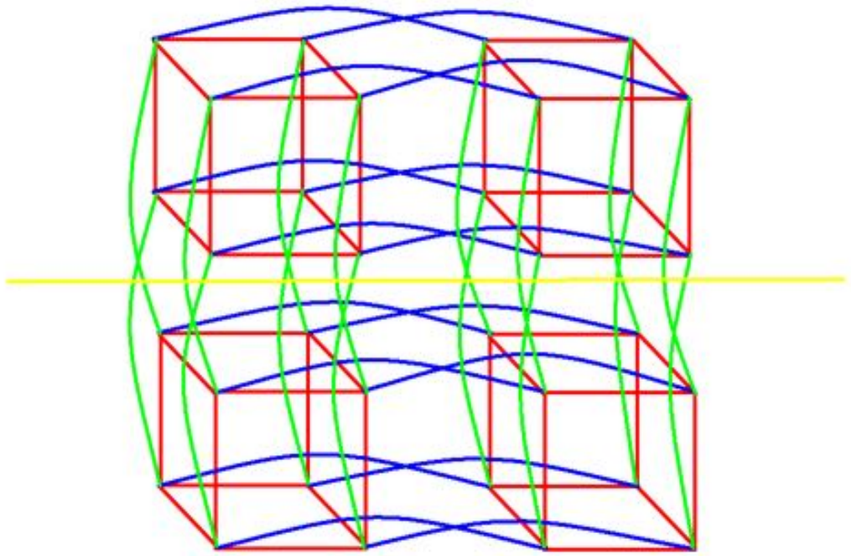


Example, The SDSC Gordon Cluster



- Diameter = 3k/2
  - For a 4x4x4 torus the max hop-count is 6
- Theoretically a torus can scale to nxnxn
- But will it actually scale?

Hypercubes

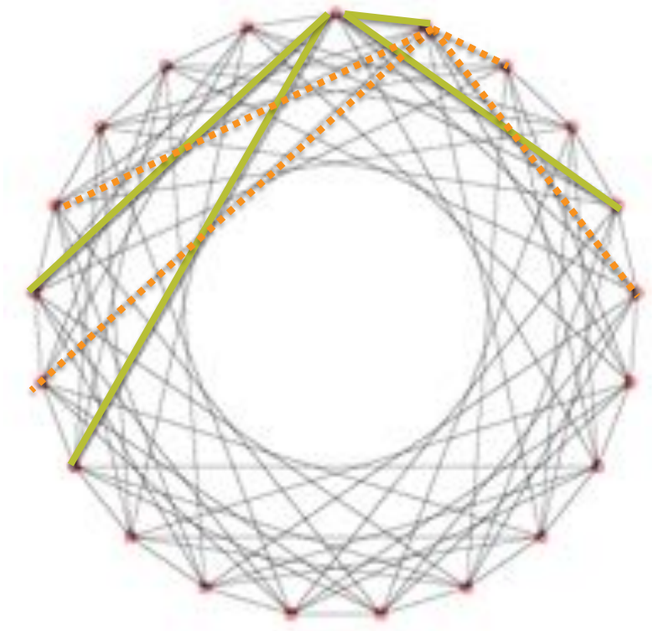


32-switch 5D Hypercube

- Number of Switches = 2^k
- Maximum diameter = k
- Scaling not as nice/easy as 3D Torus
- Again, like Torus, each vertex can be a 36p switch
- Servers to external link ratio dependent on size and CBB
- Ex; 2048 node cluster = 2^{11} (diameter=11)

The Incomplete Flat Mesh

- An “Almost all-connect” graph referenced in papers on Projective Geometry[1] / Perfect Difference Networks / Singer Sets

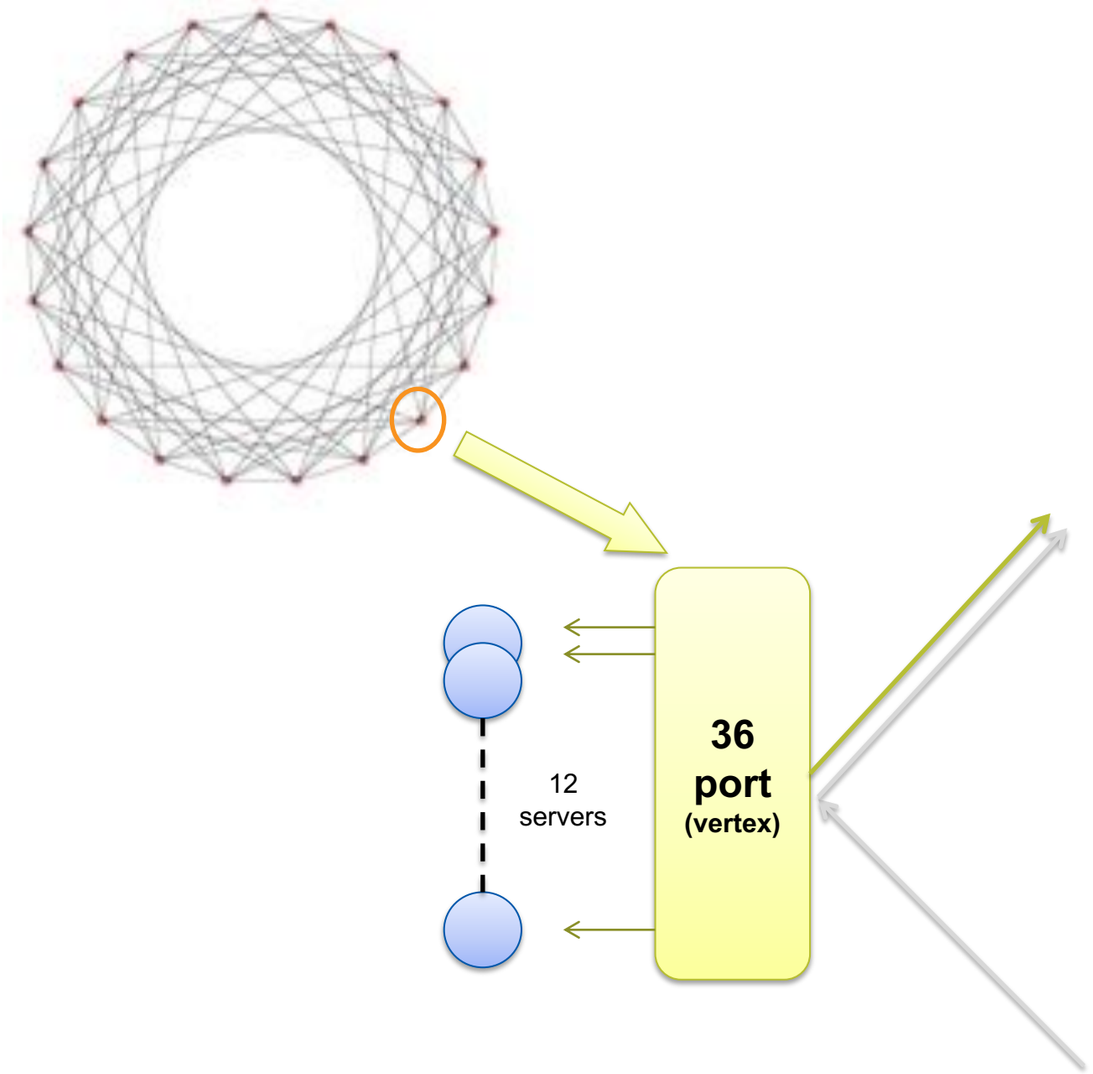


δ	*n*	Example PDS of order δ in normal form
2	7	0, 1, 3
3	13	0, 1, 3, 9
4	21	0, 1, 4, 14, 16
5	31	0, 1, 3, 8, 12, 18
7	57	0, 1, 3, 13, 32, 36, 43, 52
8	73	0, 1, 3, 7, 15, 31, 36, 54, 63
9	91	0, 1, 3, 9, 27, 49, 56, 61, 77, 81
11	133	0, 1, 3, 12, 20, 34, 38, 81, 88, 94, 104, 109
13	183	0, 1, 3, 16, 23, 28, 42, 76, 82, 86, 119, 137, 154, 175
16	273	0, 1, 3, 7, 15, 31, 63, 90, 116, 127, 136, 181, 194, 204, 233, 238, 255

[1] A new parallel architecture for sparse matrix computation based on finite projective geometries – Narendra Karmarkar
- 1991ACM 0-89791-459-7/91/0358

[2] Perfect Difference Networks and Related Interconnection Structures for Parallel and Distributed Systems
- Behrooz Parhami, Fellow, IEEE, and Mikhail Rakov

Example System Architecture



- Consider each vertex (switch) with 12 servers
- Every switch-switch link is 3x ports
- Total 8 x 3 = 24 ports used per switch for inter-switch link
- Every switch is involved in transferring data from 12 of its local servers

+

- Relaying data from 6 other servers (can be distributed anywhere)

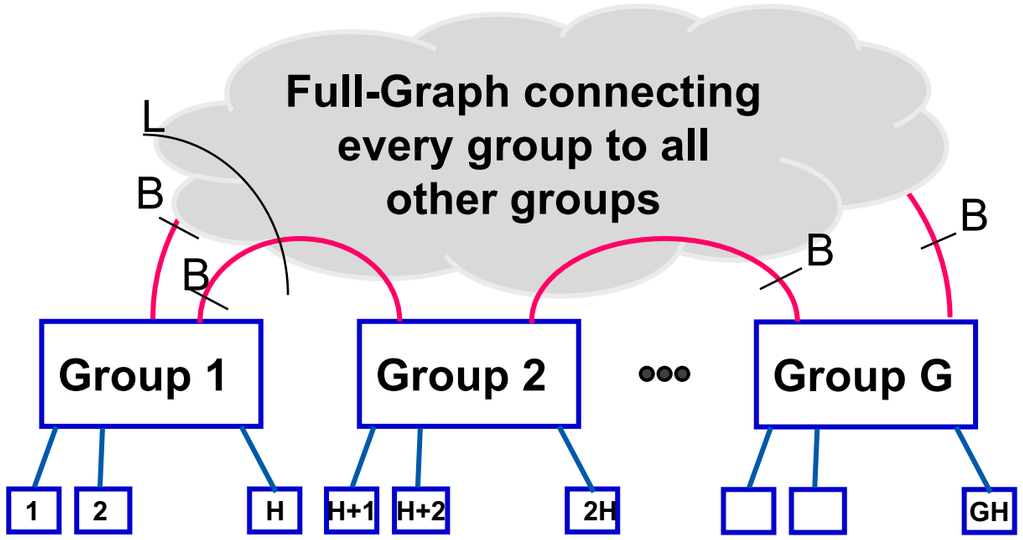
- Looks like each of the 12 “out-going” ports are carrying data from 18 servers

- In practice this is a 1.5:1 oversubscribed network

Dragonfly Topology

- Proposed by John Kim, 2008

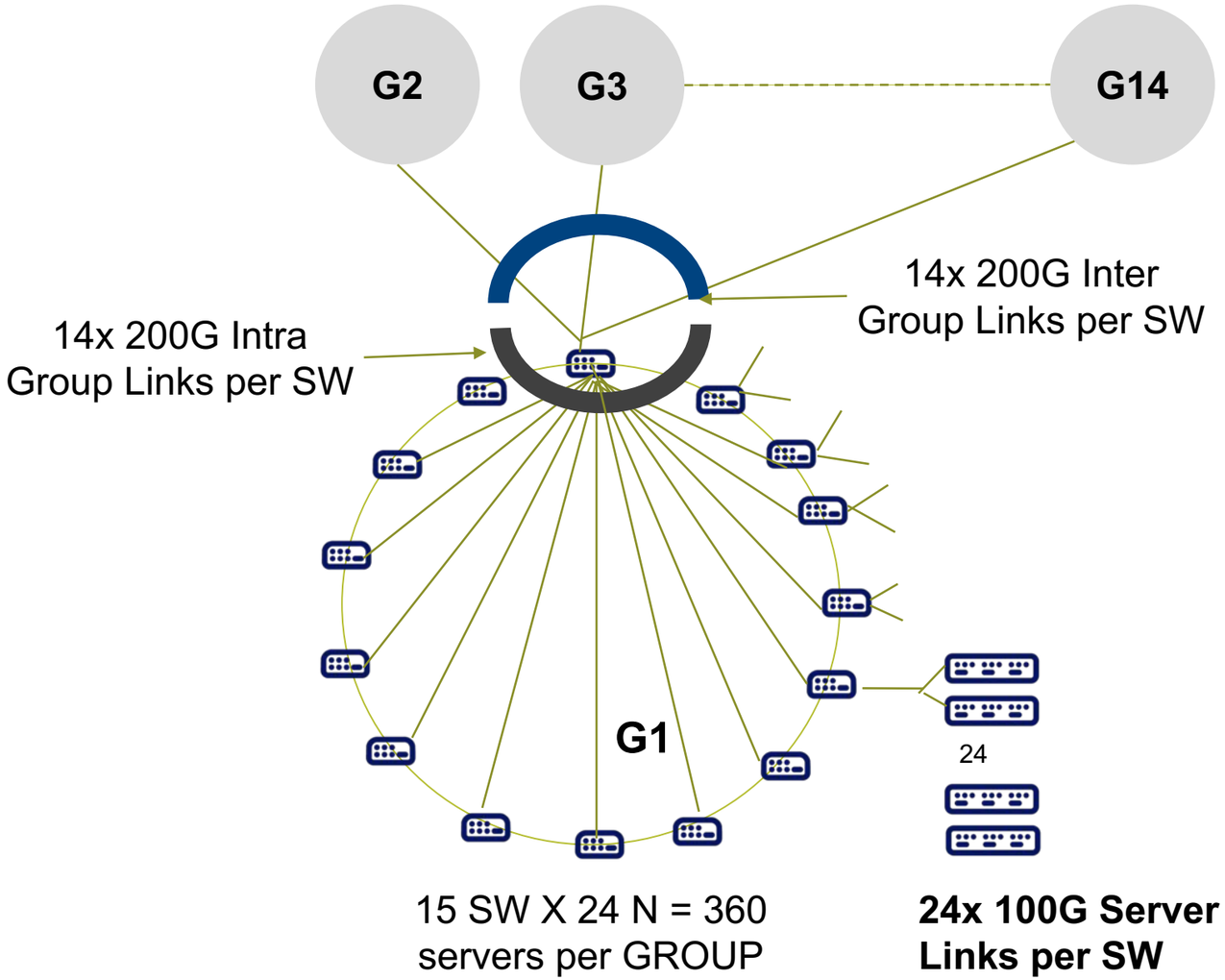
- Hierarchical topology with the following properties:
  - Several “**groups**”, connected together using **all to all links**
  - The topology inside each group can be any topology
  - Focus on reducing the number of long links and network diameter
  - Requires Adaptive Routing to enable efficient operation



Topology Parameters

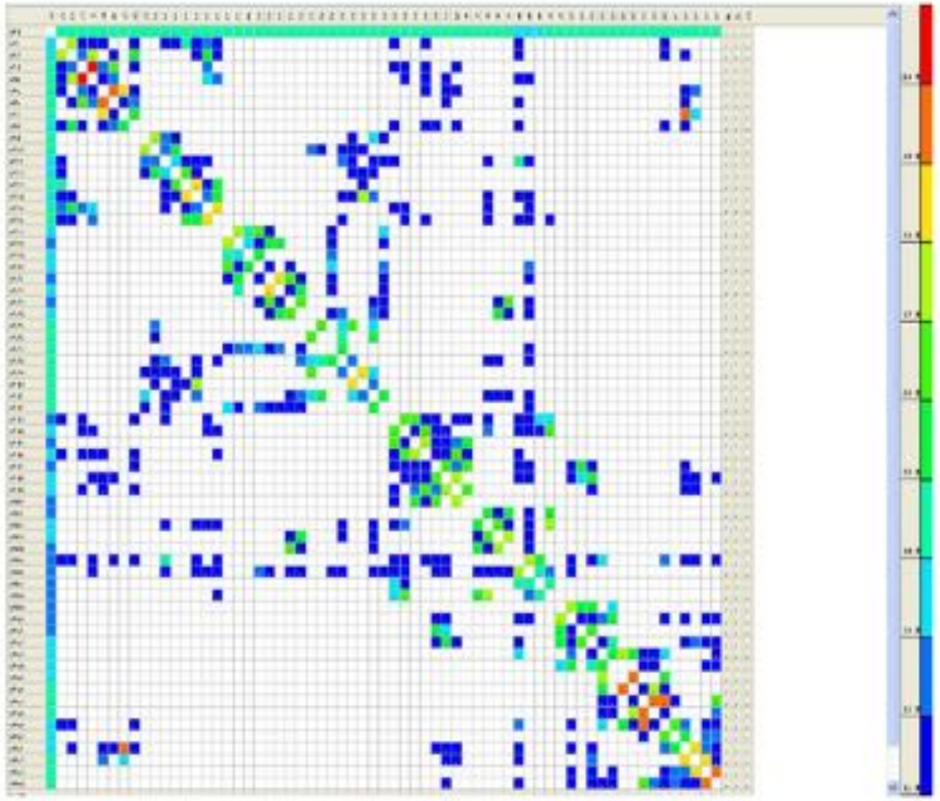
Attribute	Description
H	**H**osts per group
B	Number of **B**undled links between two groups
L	Total group **L**inks to other groups = B*(G-1)
G	Total number of **G**roups
N	Total compute **N**odes = G*H

Traditional DF {210 ASICs}



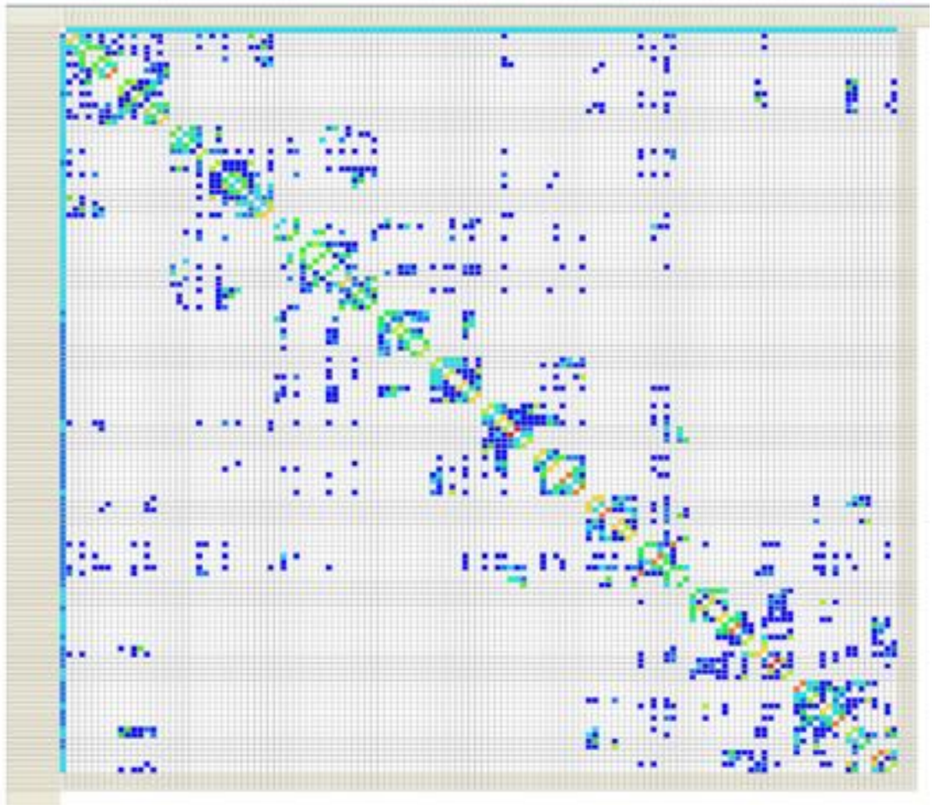
- Every Group G1 – G14 consists of;
  - 15x 40p 200G/port switch
  - 14x 200G Intra-Group links per switch
  - 14x 200G Inter-Group links per switch
    - Total 210x 200G ext bandwidth per group
  - 24x 100G servers per switch
    - Total 360x 100G servers per group
- System can accommodate 14 Groups
- Max system size 5040x 100G ports
- Min-hop routing will have 3-hop across any node pair
- Non Min-hop can be higher but need not be more than 4-hop

Example Application Communication Pattern



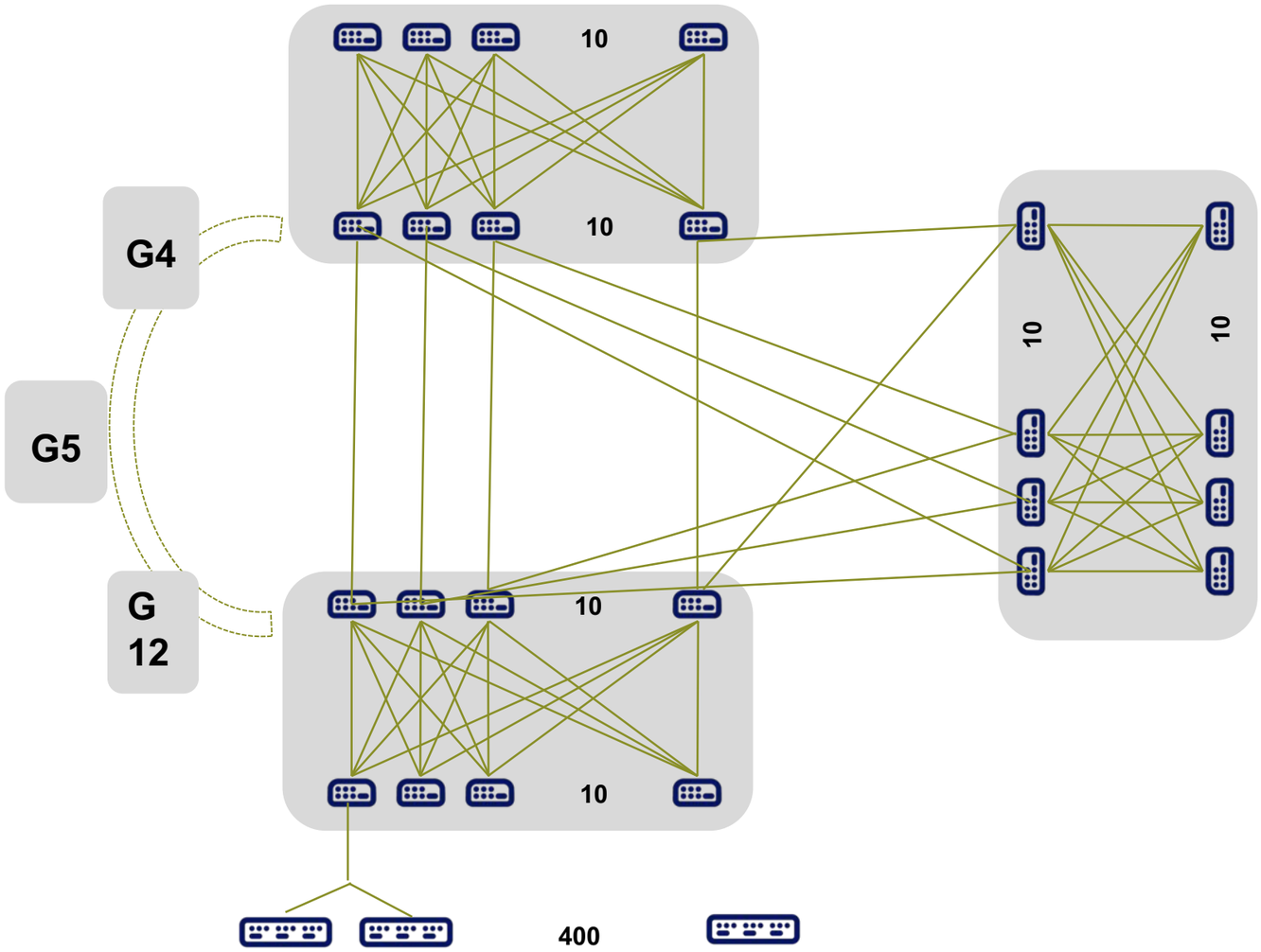
Communication pattern for 64-core run

CFD code



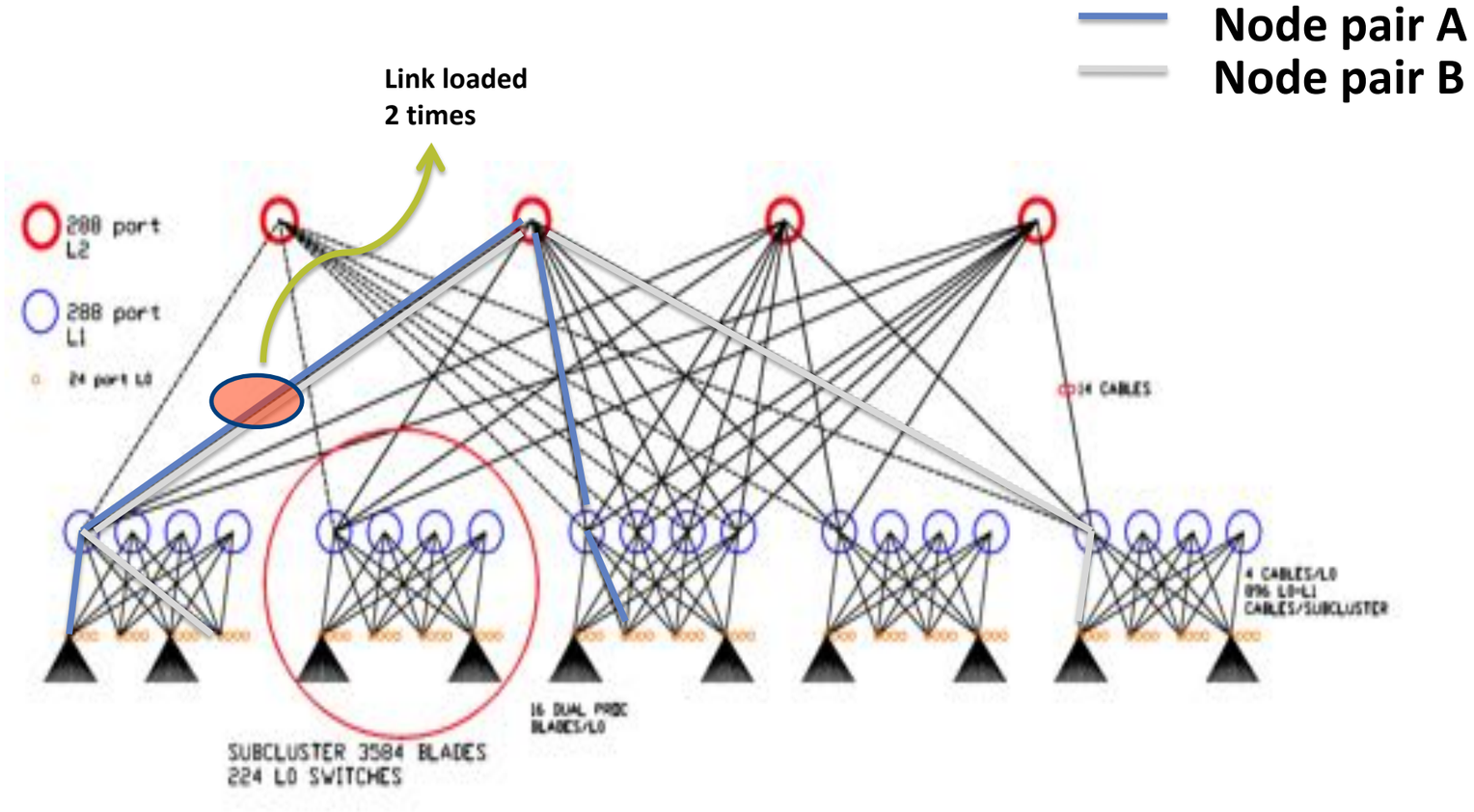
128-core run

Option-5 : Dragonfy+ {240 ASICs}



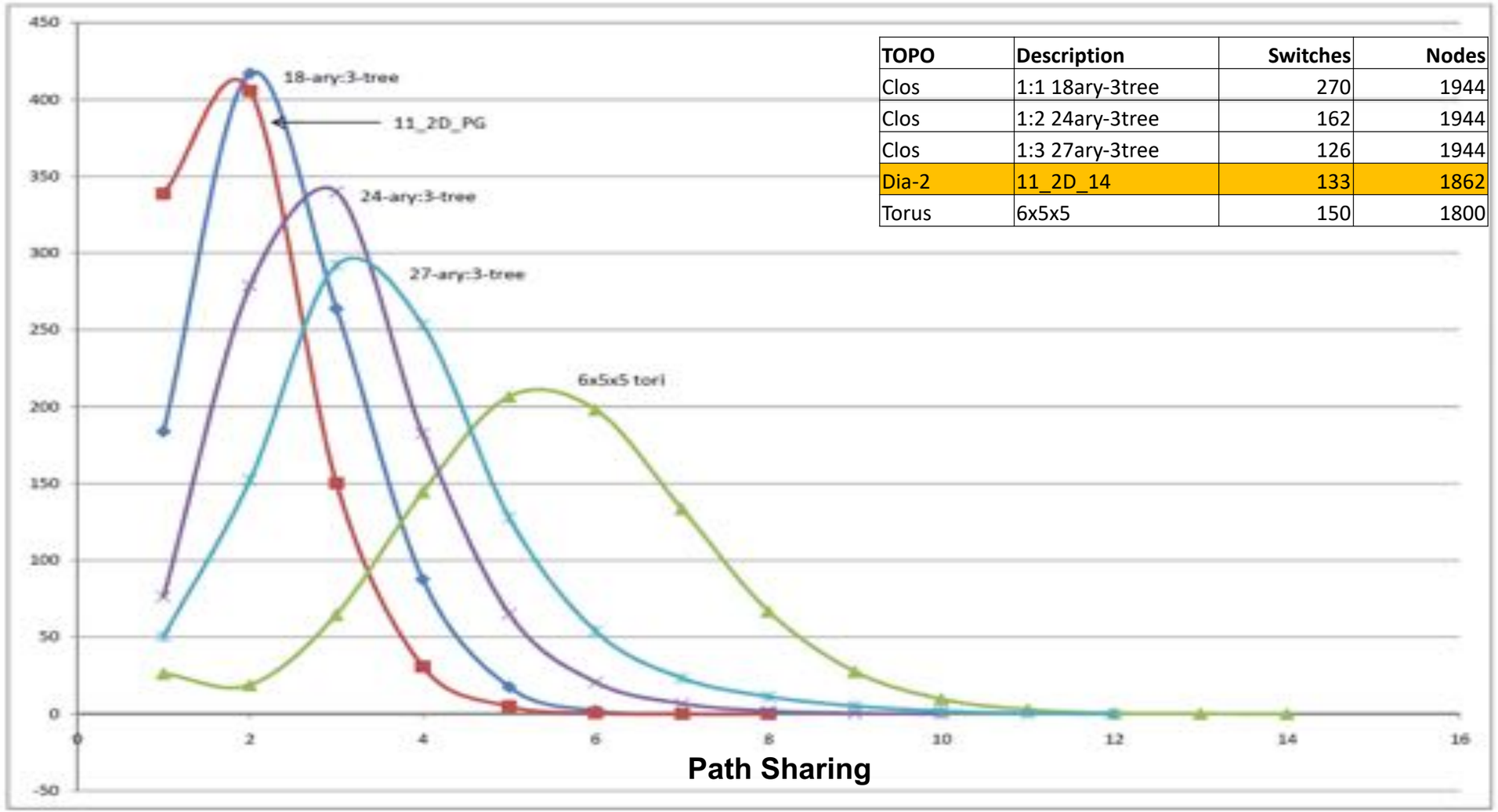
- Each Group is full bi-partite
  - 20x 40p 200G switch per group
  - 200x 200G intra-group links per switch
  - 200x 200G inter-group links per switch
- 400 servers per group
- System can accommodate up-to 20 groups {8000 servers}
- Min-hop routing will have 4-hops across any node pair

Over Subscription and Bisection in Static Routing



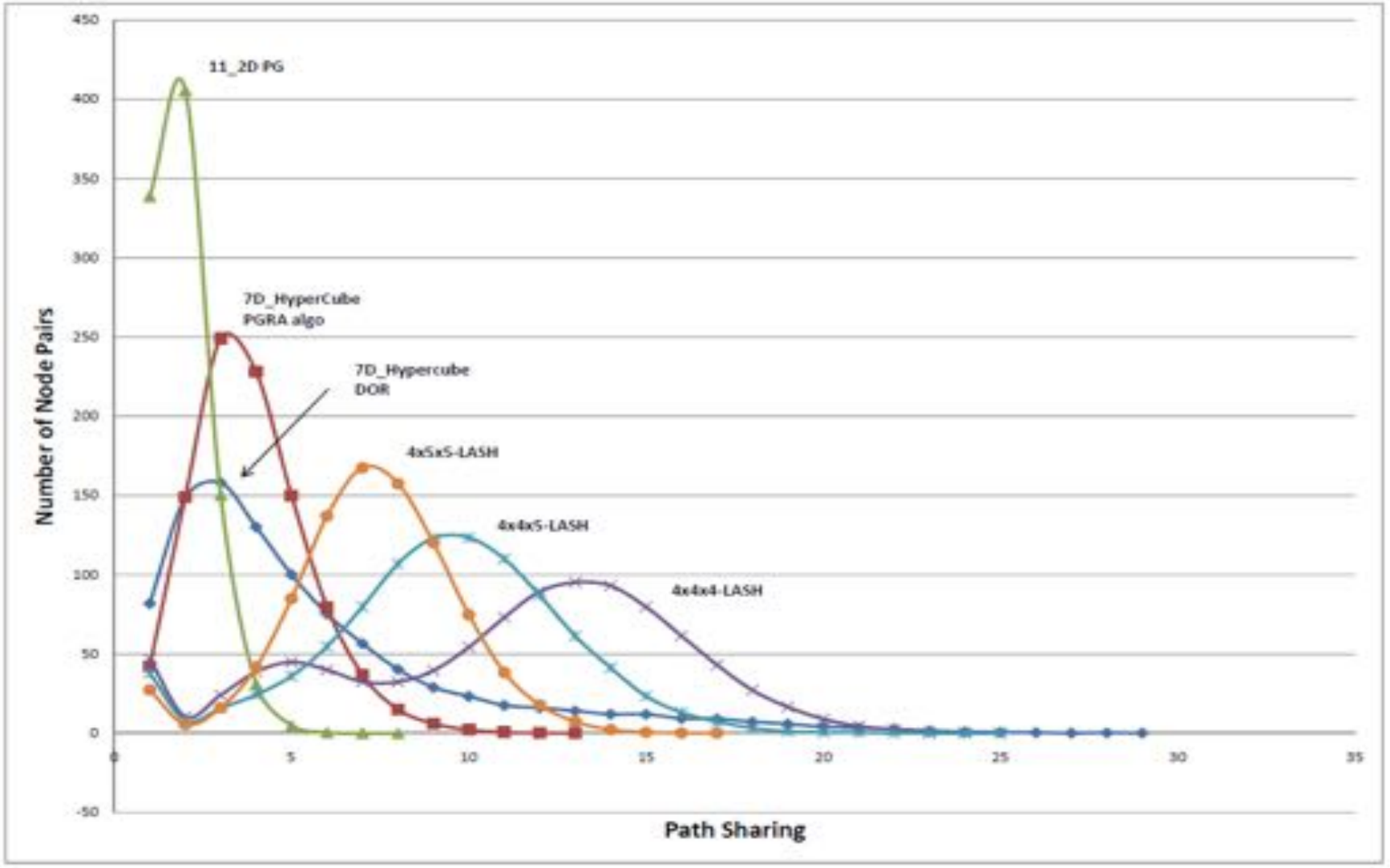
- Divide cluster into two subset of nodes (random)
- Subset-1 = Tx and subset-2 =Rx
- Trace data-flow path between each communicating pair
- Increment “subscription bucket” if any segment in flow path overlaps
- Repeat n times for statistical accuracy and plot

And Some Simple Results



TOPO	Description	Switches	Nodes
Clos	1:1 18ary-3tree	270	1944
Clos	1:2 24ary-3tree	162	1944
Clos	1:3 27ary-3tree	126	1944
Dia-2	11_2D_14	133	1862
Torus	6x5x5	150	1800

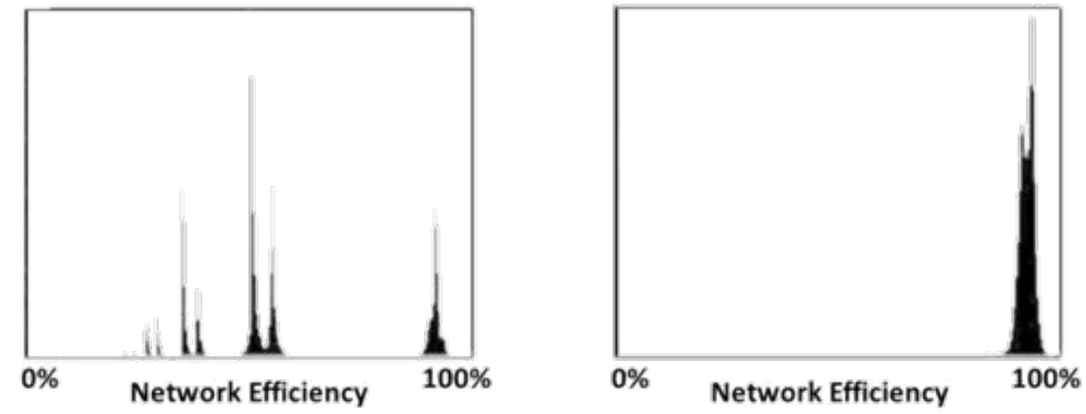
And Some Simple Results Cont..



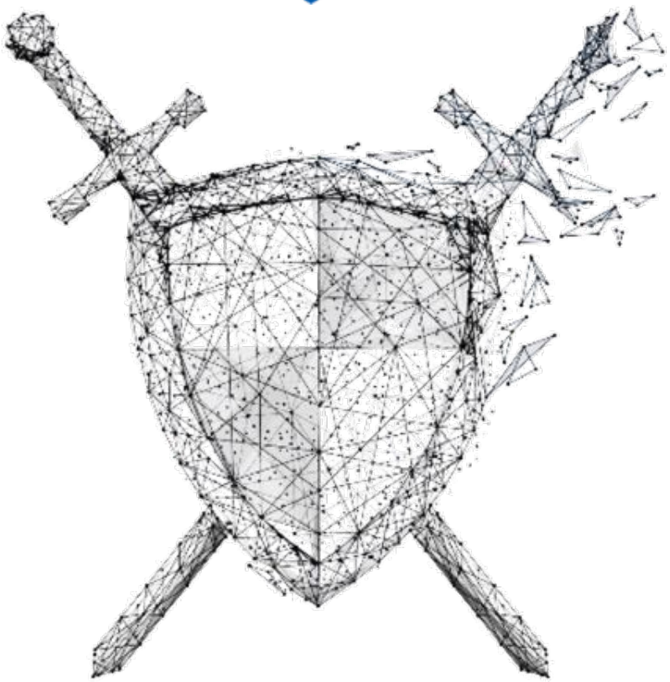
Adaptive Routing (AR) Performance – ORNL Summit

- Oak Ridge National Laboratory – Coral Summit supercomputer
- Bisection bandwidth benchmark, based on mpiGraph
  - Explores the bandwidth between possible MPI process pairs
- AR results demonstrate an average performance of 96% of the maximum bandwidth measured

mpiGraph explores the bandwidth between possible MPI process pairs. In the histograms, the single cluster with AR indicates that all pairs achieve nearly maximum bandwidth while single-path static routing has nine clusters as congestion limits bandwidth, negatively impacting overall application performance.



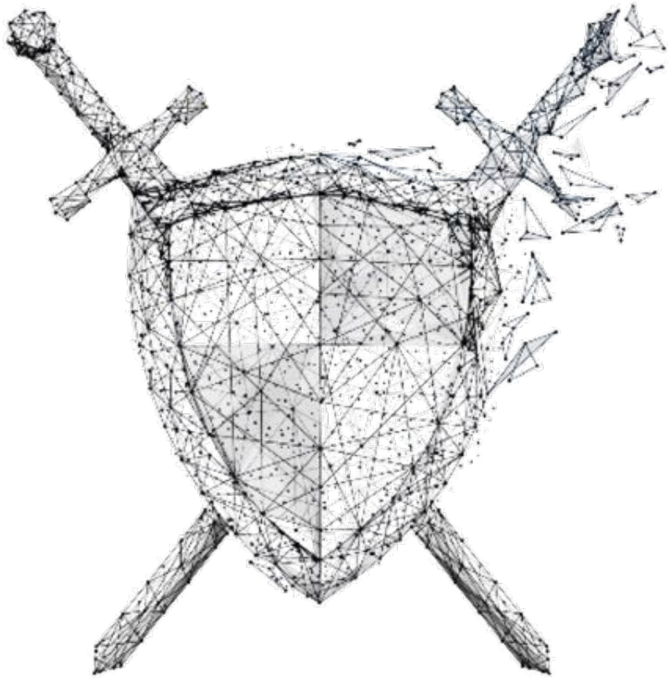
SHIELD
Self Healing Technology



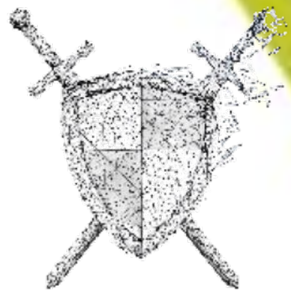
Self Healing Technologies

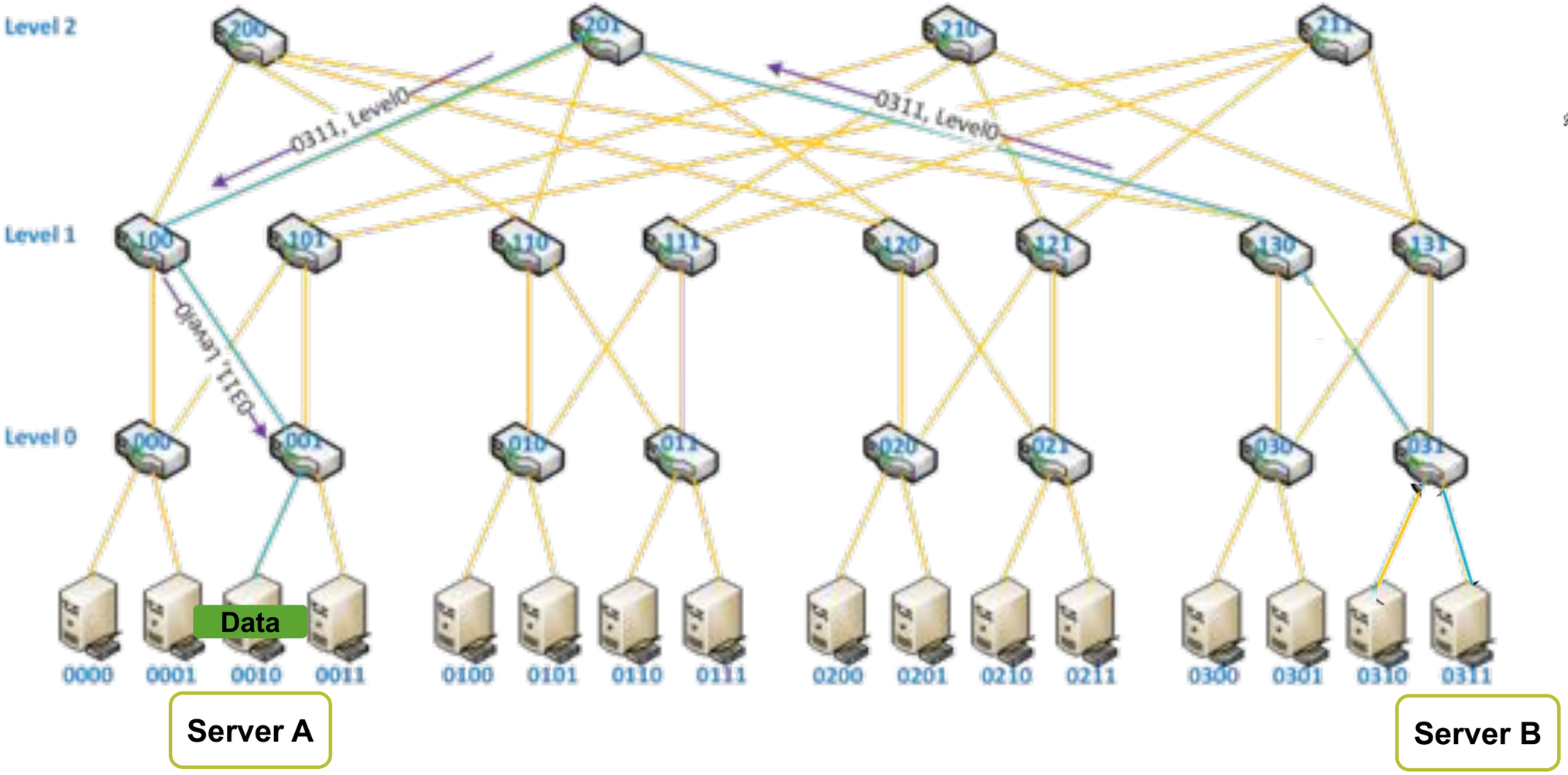
Enables Unbreakable Data Centers

- The ability to overcome network failures, locally, by the switches
- Software-based solutions suffer from long delays detecting network failures
  - 5-30 seconds for 1K to 10K nodes clusters
  - Accelerates network recovery time by 5000X
  - The higher the speed or scale the greater the recovery value

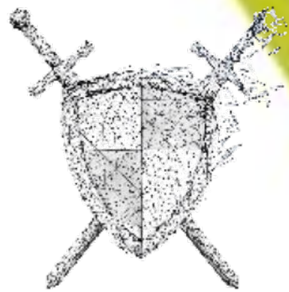


Consider a Flow From A to B





The Simple Case: Local Fix



Level 2

Level 1

0311, Level0

0311, Level0

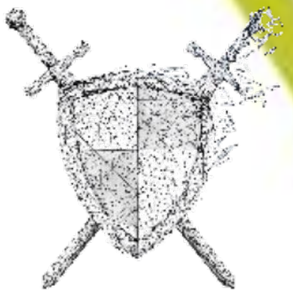
0311, Level0

Data

Server A

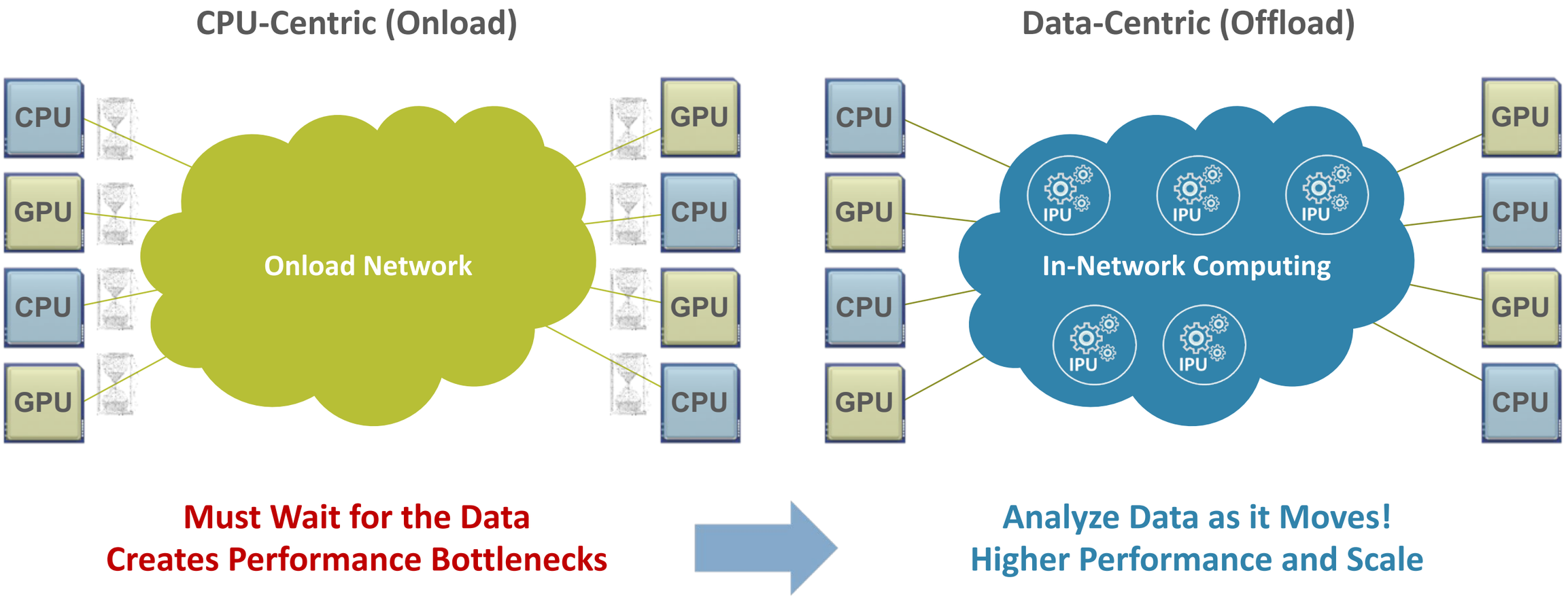
Server B

The Remote Case - Using Fault Recovery Notifications



The Need for Intelligent Networks

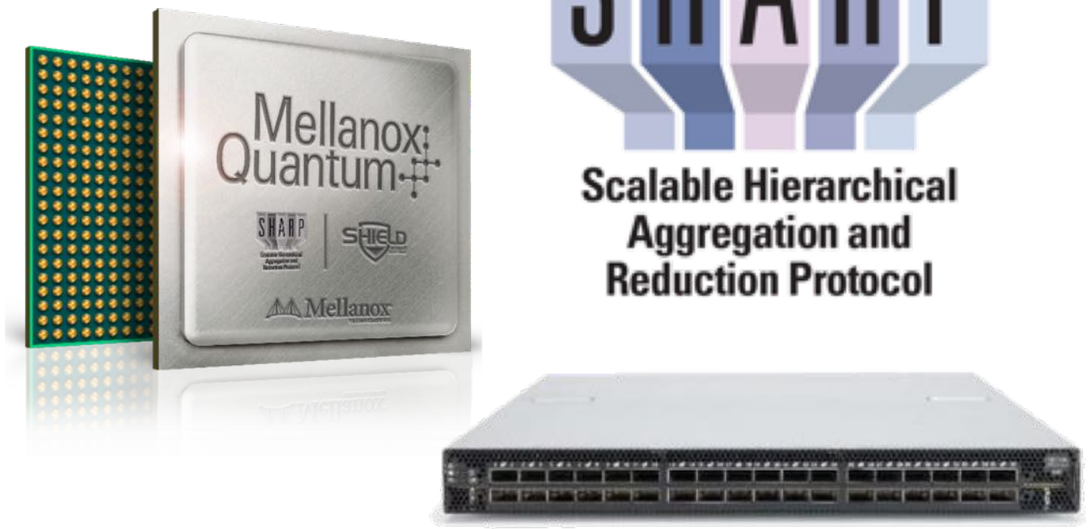
Faster Data Speeds and In-Network Computing
Enable Higher Performance and Scale



Must Wait for the Data
Creates Performance Bottlenecks

Analyze Data as it Moves!
Higher Performance and Scale

Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)

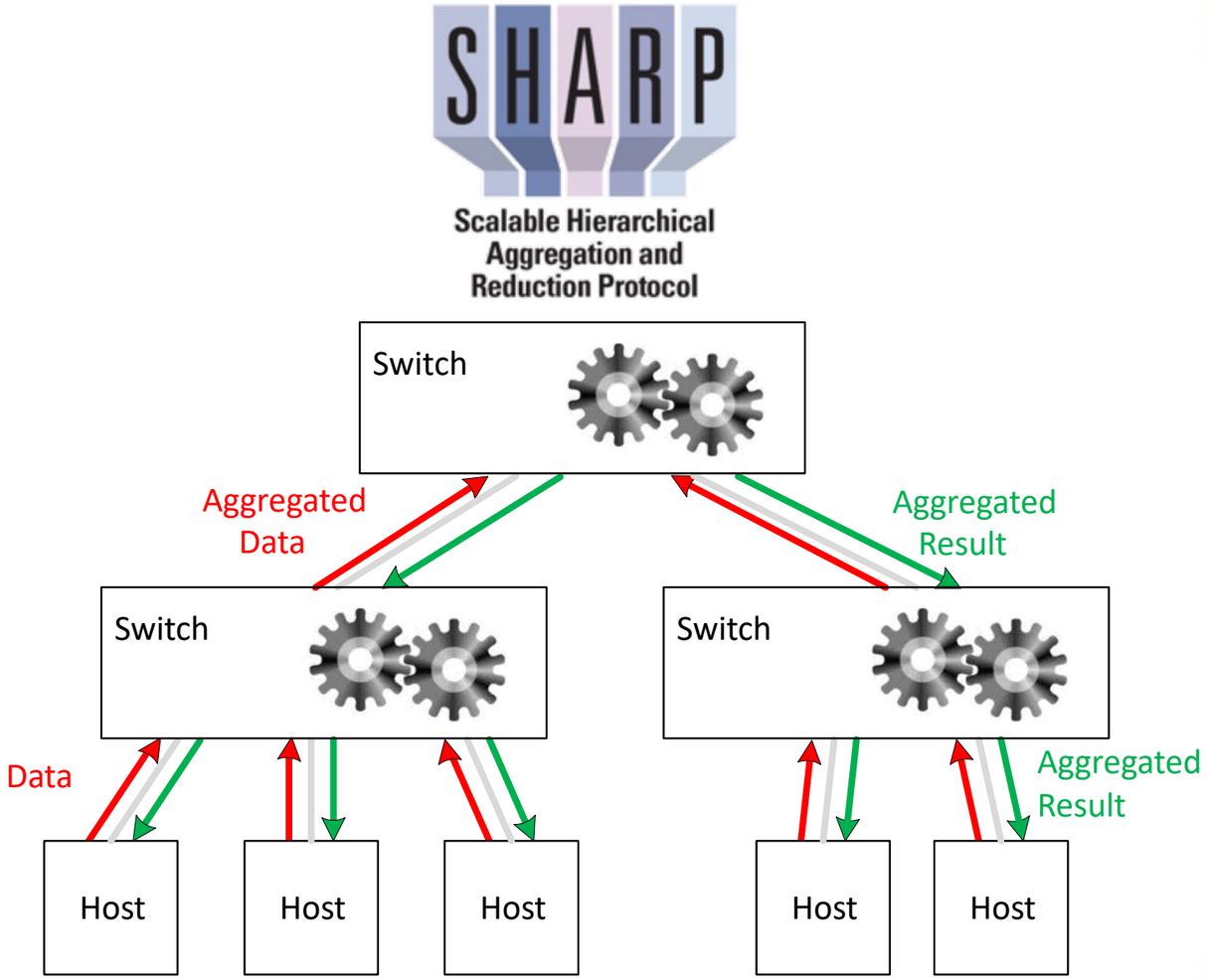


SHARP

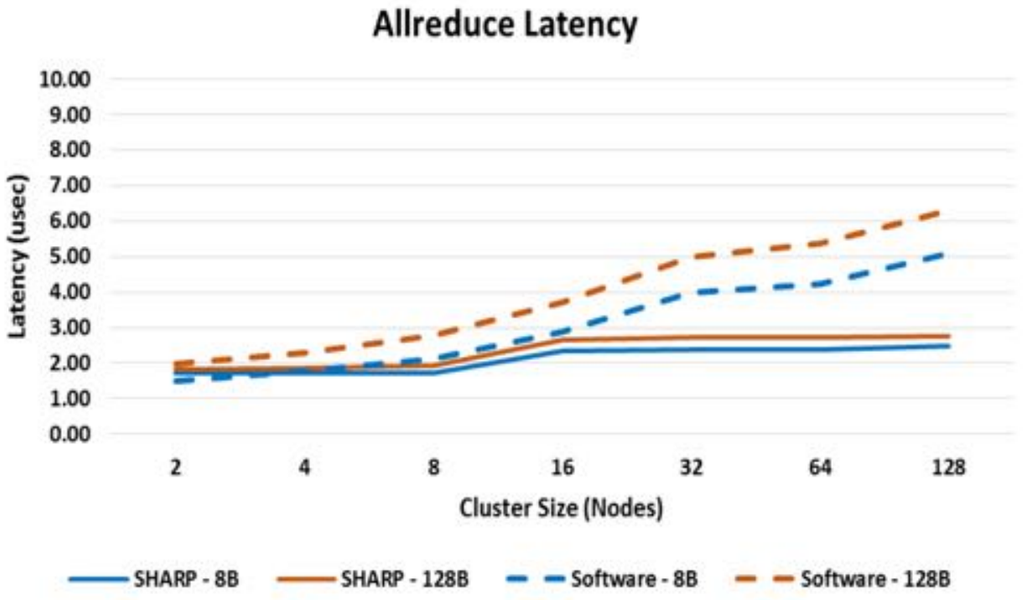
Scalable Hierarchical Aggregation and Reduction Protocol

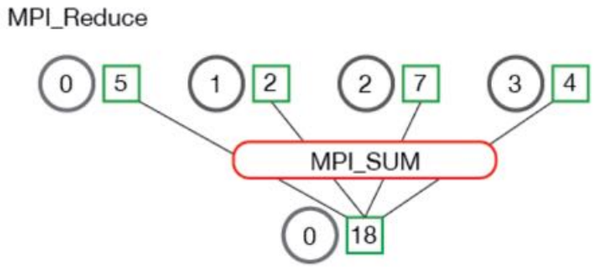
Scalable Hierarchical Aggregation and Reduction Protocol (SHARP)

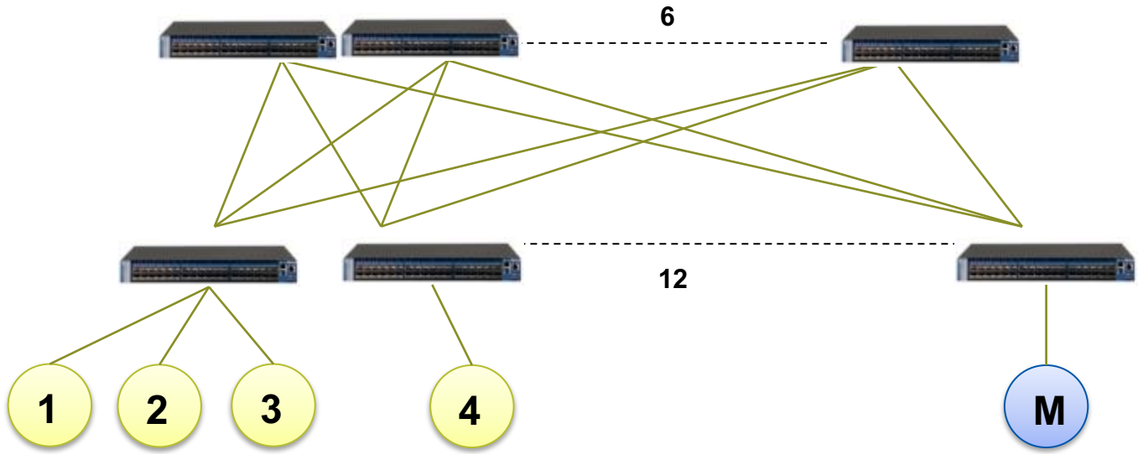
- Reliable Scalable General Purpose Primitive
  - In-network Tree based aggregation mechanism
  - Large number of groups
  - Multiple simultaneous outstanding operations
- Applicable to Multiple Use-cases
  - HPC Applications using MPI / SHMEM
  - Distributed Machine Learning applications
- Scalable High Performance Collective Offload
  - Barrier, Reduce, All-Reduce, Broadcast and more
  - Sum, Min, Max, Min-loc, max-loc, OR, XOR, AND
  - Integer and Floating-Point, 16/32/64 bits



SHARP Allreduce Performance Advantages



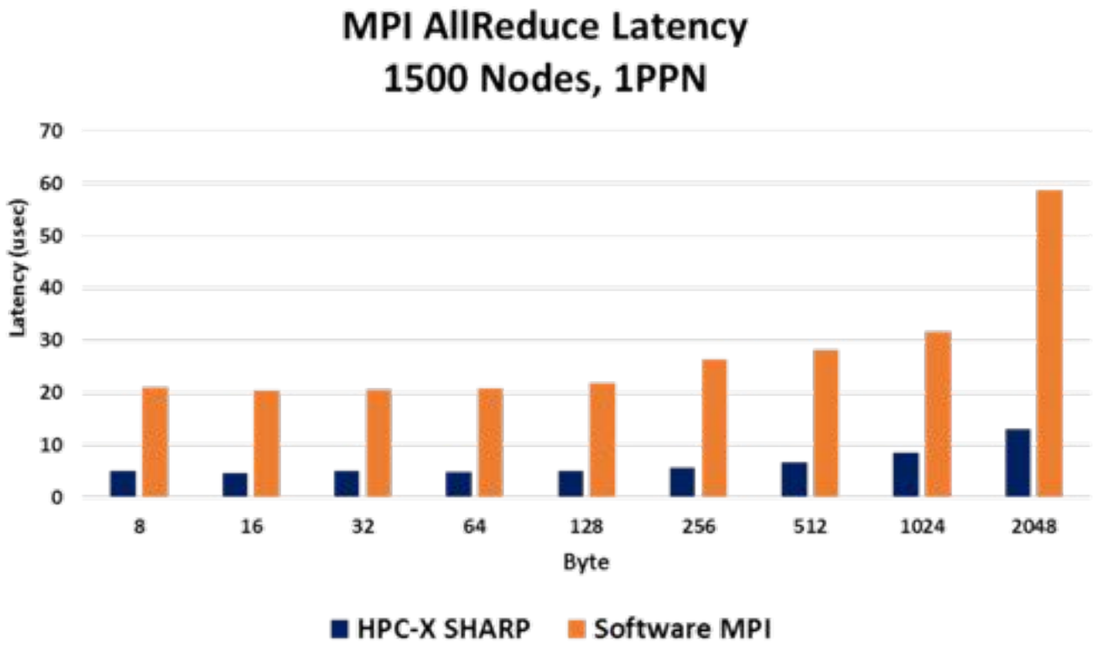


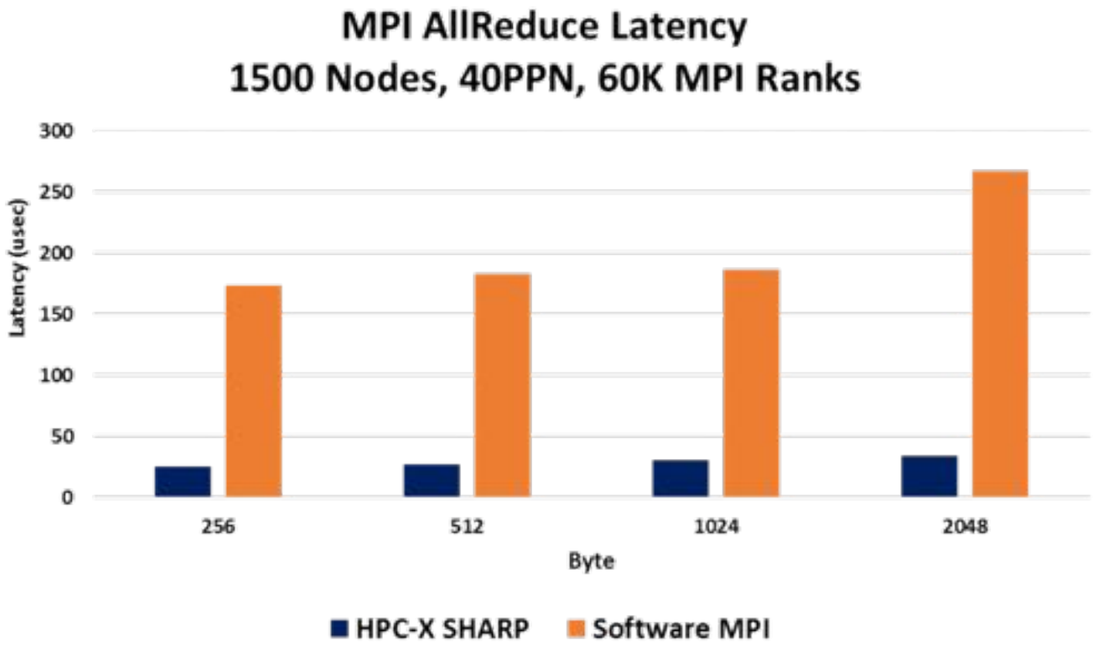


SHARP enables 75% Reduction in Latency
Providing Scalable Flat Latency

SHARP AllReduce Performance Advantages

1500 Nodes, 60K MPI Ranks, Dragonfly+ Topology



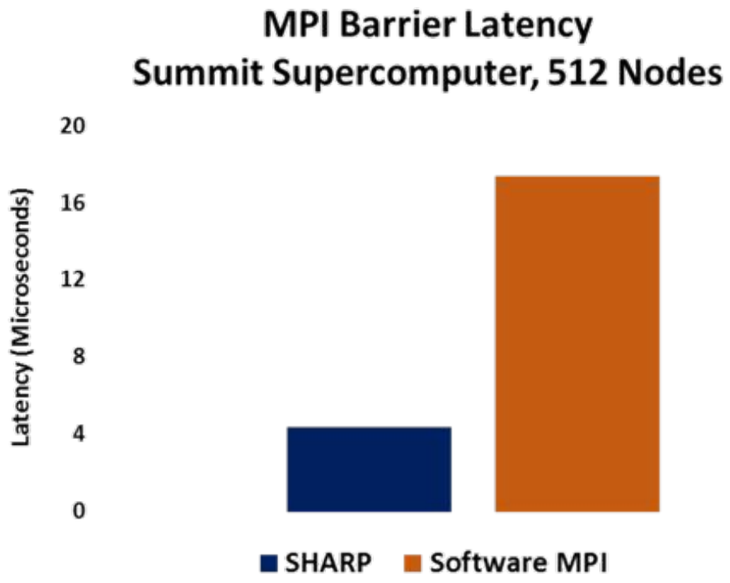


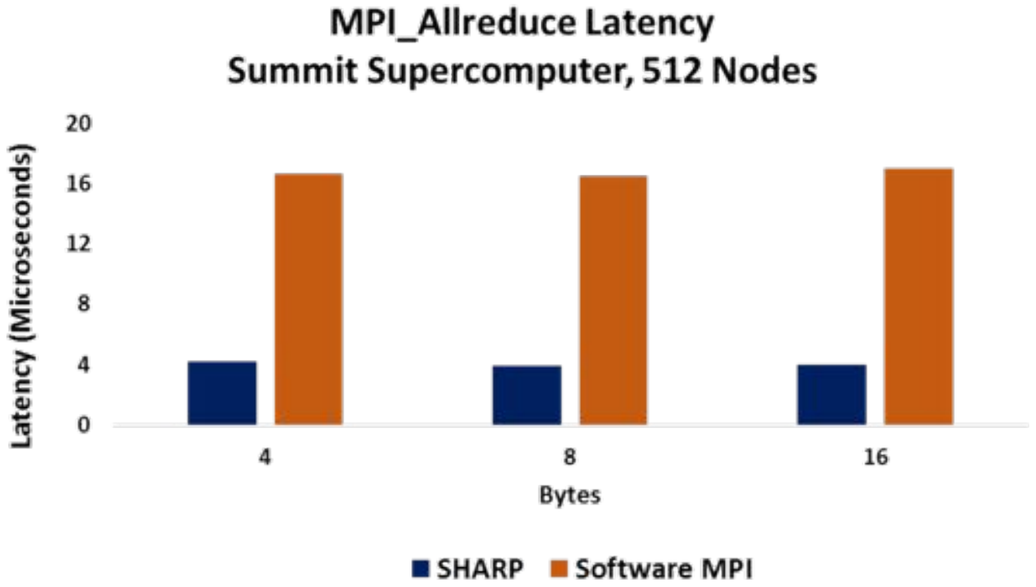
Scalable Hierarchical Aggregation and Reduction Protocol

SHARP Enables Highest Performance

SHARP AllReduce Performance Advantages

Oak Ridge National Laboratory – Coral Summit Supercomputer





SHARP

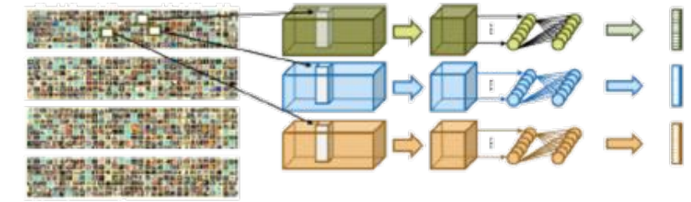
Scalable Hierarchical Aggregation and Reduction Protocol

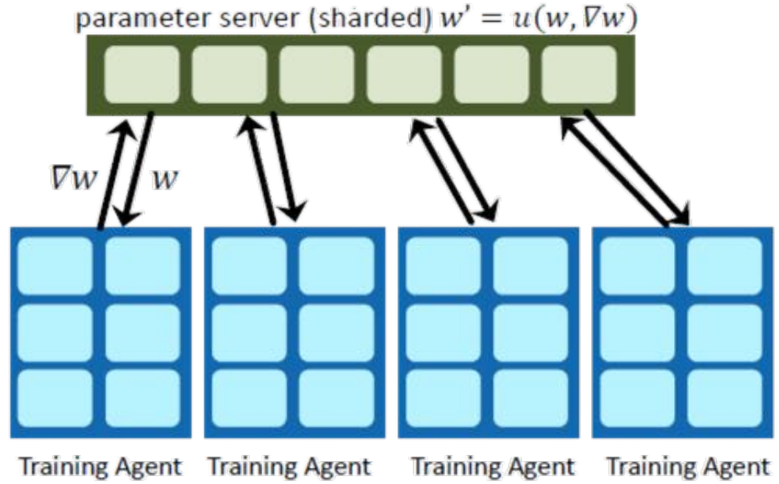
SHARP Enables Highest Performance

SHARP Accelerates AI Performance

The CPU in a parameter server becomes the bottleneck

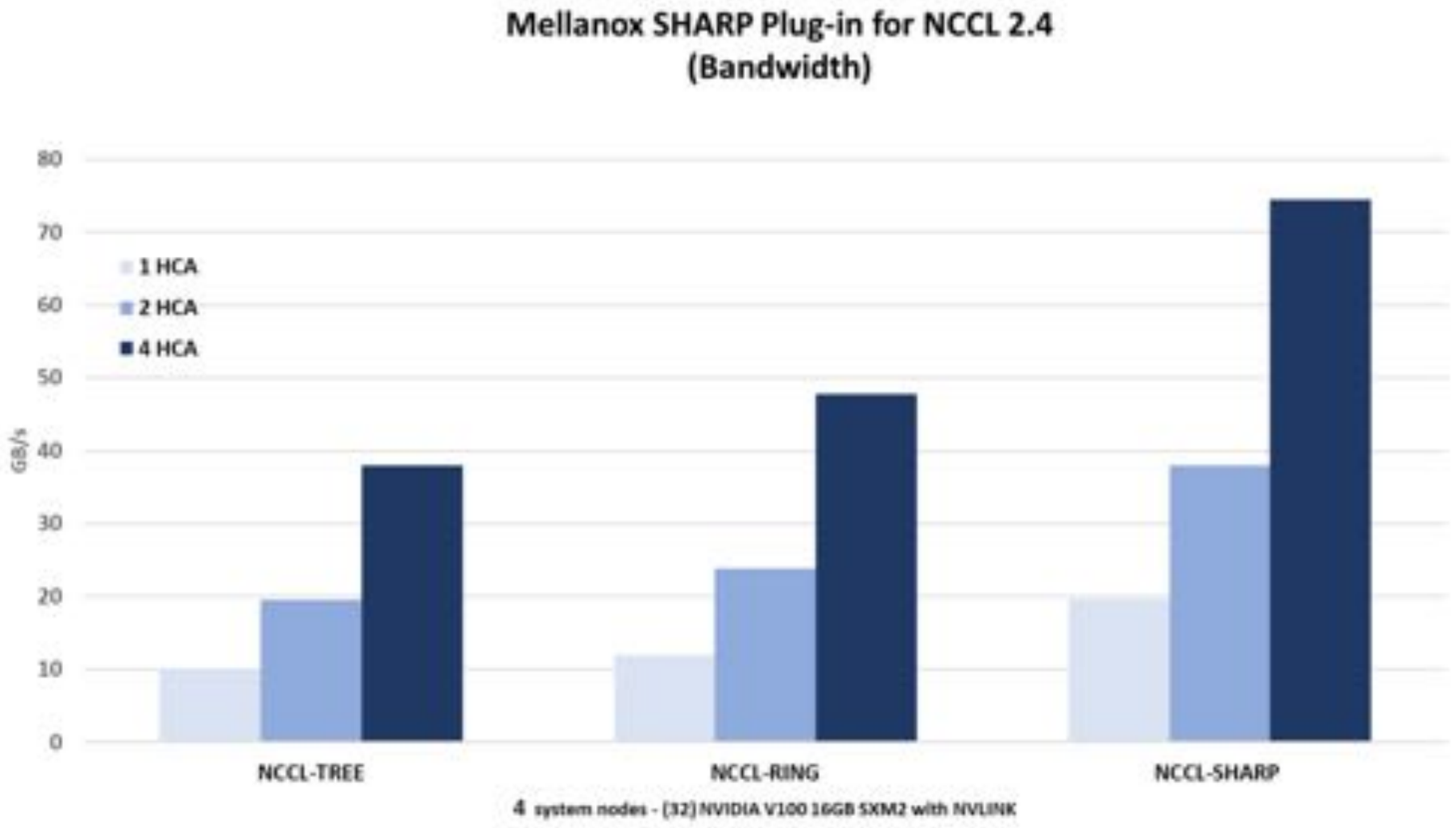
SHARP Scalable Hierarchical Aggregation and Reduction Protocol



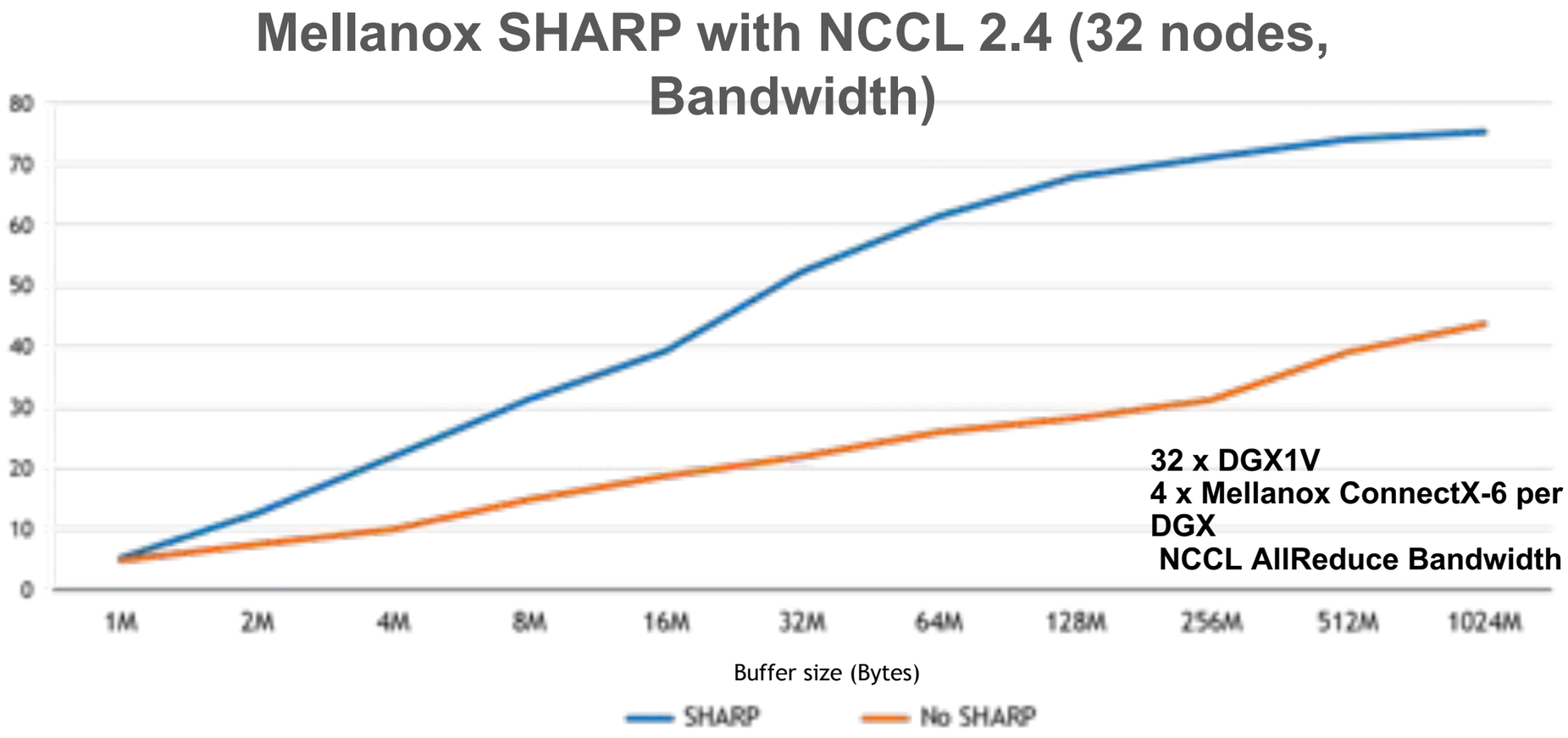


Performs the Gradient Averaging
Replaces all physical parameter servers
Accelerate AI Performance

SHARP Delivers Highest Performance for AI



SHARP Delivers Highest Performance for AI

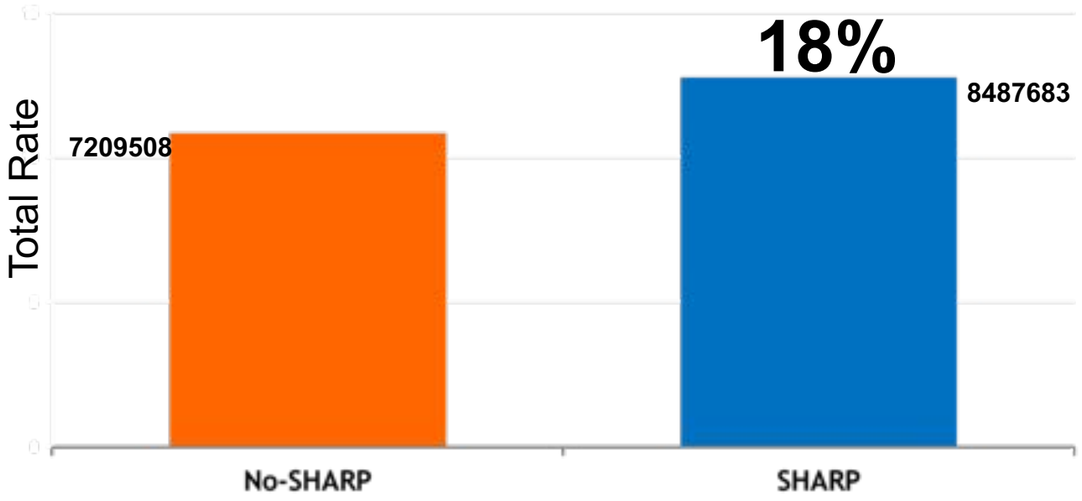


SHARP
Scalable Hierarchical Aggregation and Reduction Protocol

SHARP Delivers up to 2.4X Higher Performance

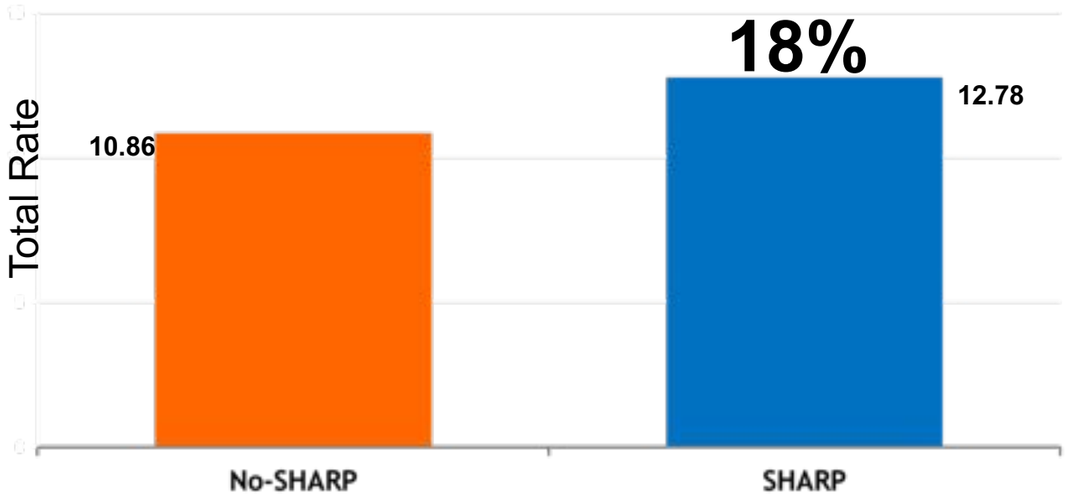
SHARP Delivers Highest Performance for AI

GNMT MLPerf Benchmark
Neural Machine Translation



24xDGX1V + 4xMellanox ConnectX-6
GNMT MLPerf 0.6 benchmark: Batch Size=32, Overlap=0.15

VAE Benchmark
Variable Auto-Encoder



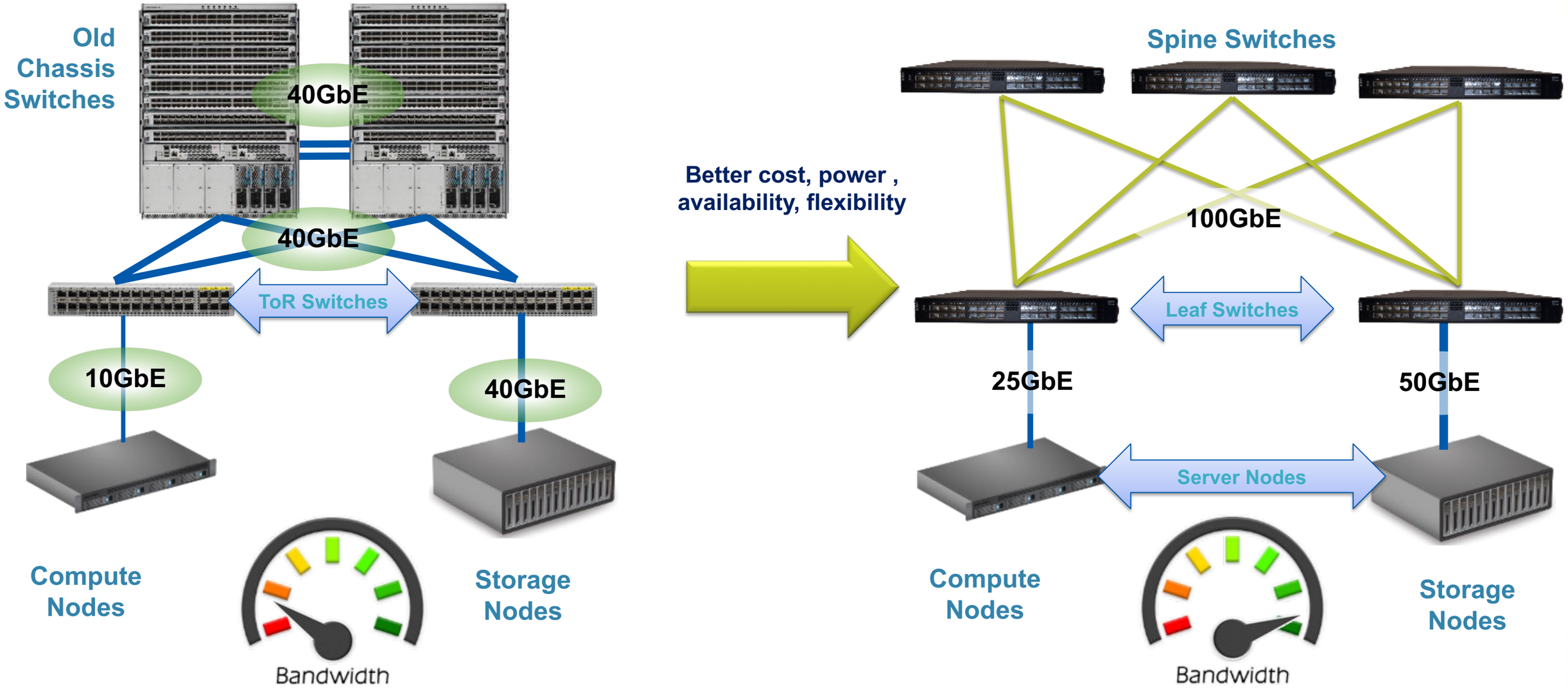
32xDGX1V + 4xMellanox ConnectX-6
VAE benchmark: Model=3, BS=512

SHARP
Scalable Hierarchical Aggregation and Reduction Protocol

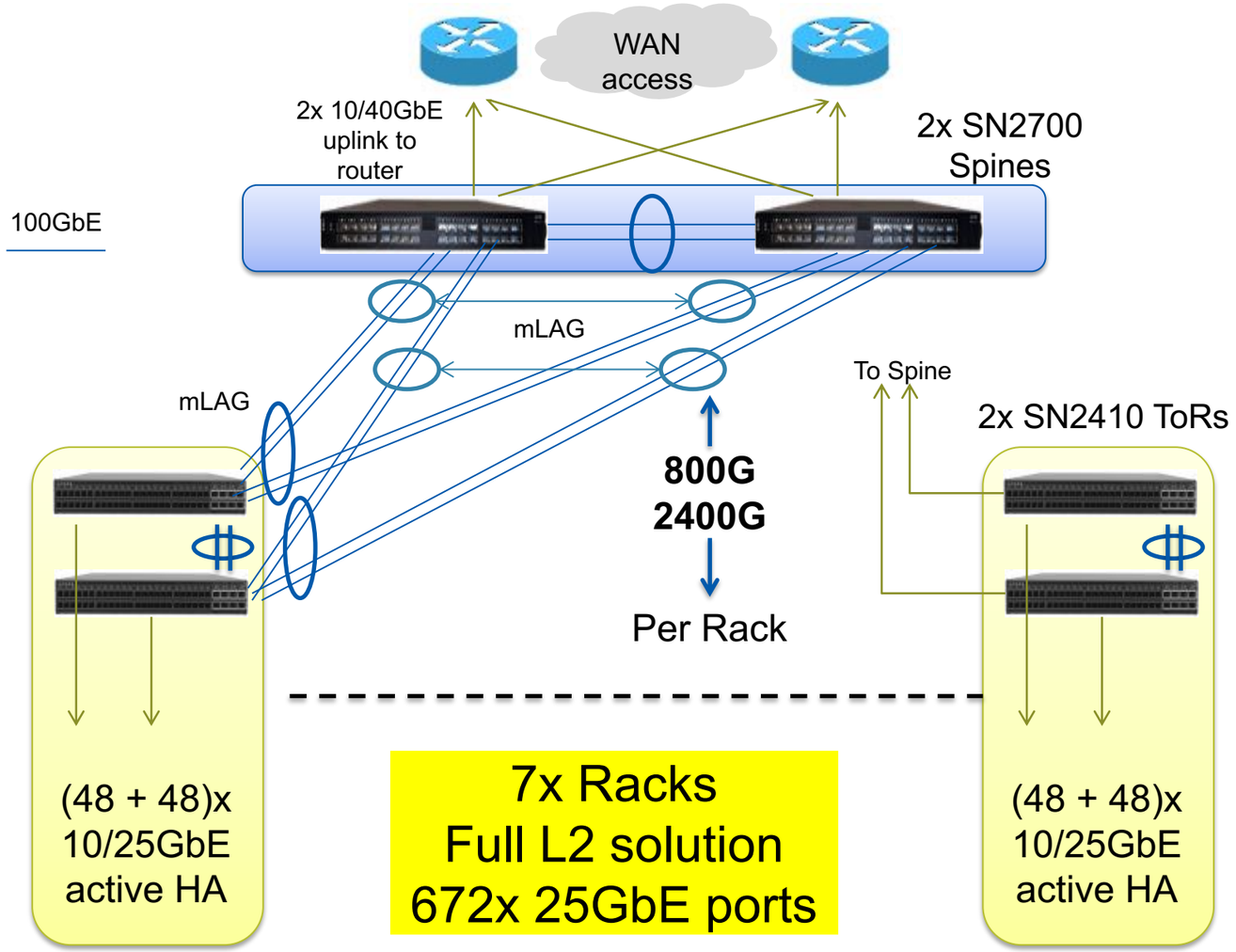
SHARP Delivers Highest Performance

Infiniband or Ethernet Fabric Golden Question!!?

Modern Leaf-Spine Networks in Ethernet



Small/Medium AI Cloud Deployment 25GbE



- Pure L2 Network; full HA and no-SPoF
- Ideal for small/medium private cloud
- Phase-1: Start with as small as 1 Rack and 2x SN2410
- Phase-2: Add 2x SN2700 spines and build up-to 7x Racks in pure L2 domain
- mLAG on ToRs and spines for full active-active HA
- (2+2)x 100GbE or (4+4)x 50GbE ports available per rack for high performance/"fat" storage nodes
- 48+48x 25GbE for compute/hyper-converged infrastructure

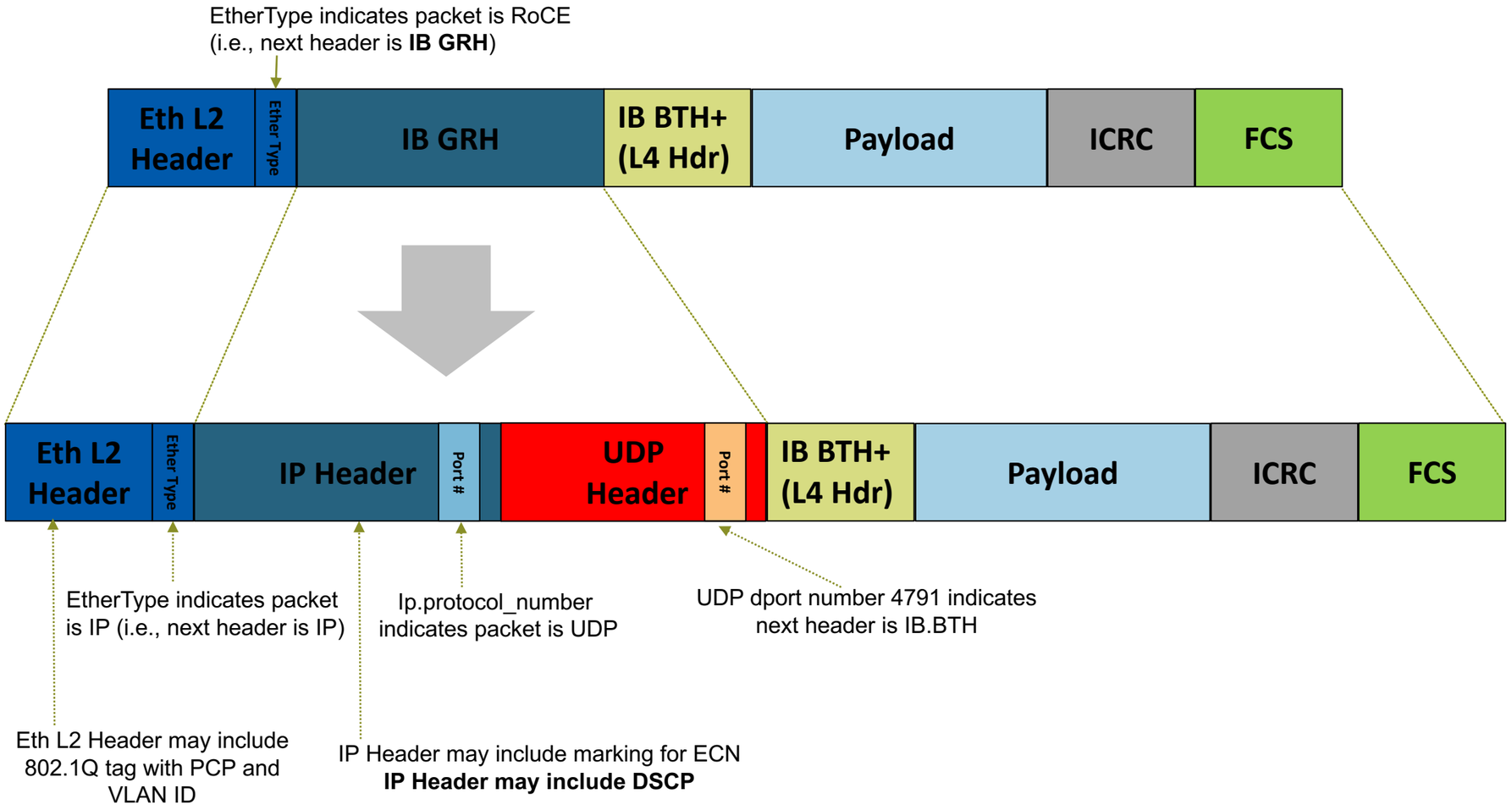
RoCE in a Nutshell

- **What is RoCE (RDMA over Converged Ethernet)?**
  - [Wikipedia] **RoCE is a network protocol that allows RDMA over an Ethernet network**. It does this by encapsulating an InfiniBand transport packet over Ethernet. There are two RoCE versions, RoCE v1 and RoCE v2. RoCE v1 is an Ethernet link layer protocol and hence allows communication between any two hosts in the same Ethernet broadcast domain. RoCE v2 is an internet layer protocol which means that RoCE v2 packets can be routed.
  - RoCE is a standard for RDMA over Ethernet defined by the IBTA (InfiniBand Trade Association)

- **Why RoCE?**
  - Maintain Ethernet Infrastructure with RDMA capabilities
  - All the benefits of RDMA: Transport Offload, Kernel Bypass, High-Throughput and Low-Latency RDMA

RoCEv1 and RoCEv2 Header Format

- RoCEv1
  - L2
  - GRH
- RoCEv2
  - L3
  - IP/UDP based
  - Routable



What Makes a Great Ethernet AI Switch?

Simple Configuration

- 1 Command CLI config
- 1 Click GUI config

High Performance

- High PPS & Low latency
- Fair & Predictable performance

Advanced Congestion Control

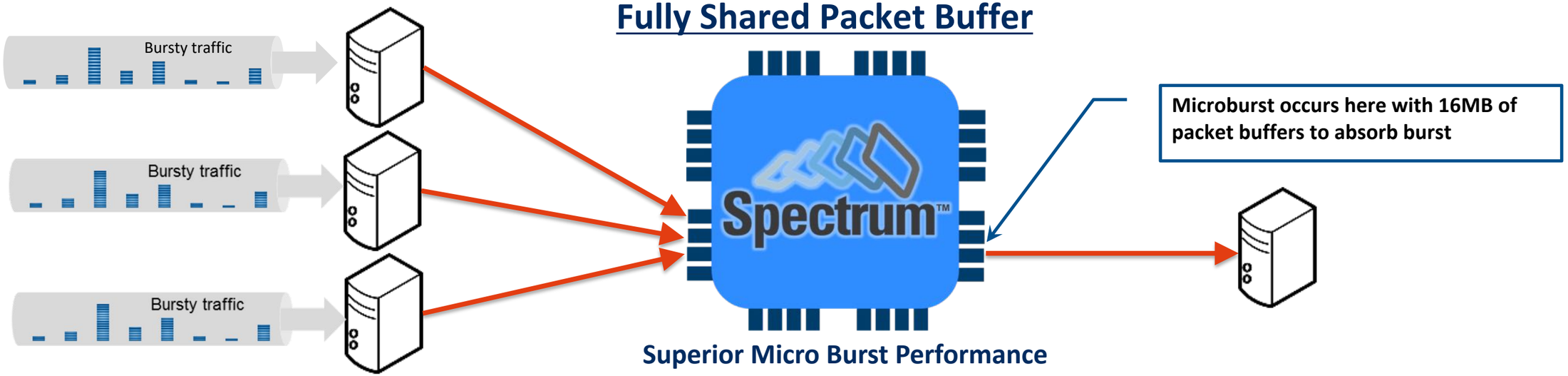
- Early detection and prevention
- RoCE over VXLAN

Extensive Visibility

- Single pane-of-glass
- Real time RoCE Telemetry

Fully Shared Buffers are Superior for RoCE

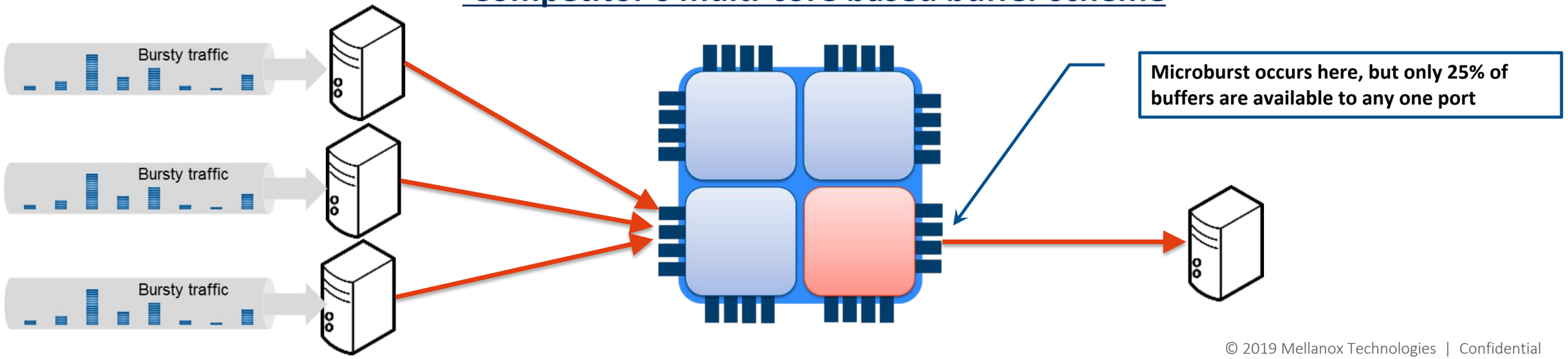
Fully Shared Packet Buffer



Superior Micro Burst Performance

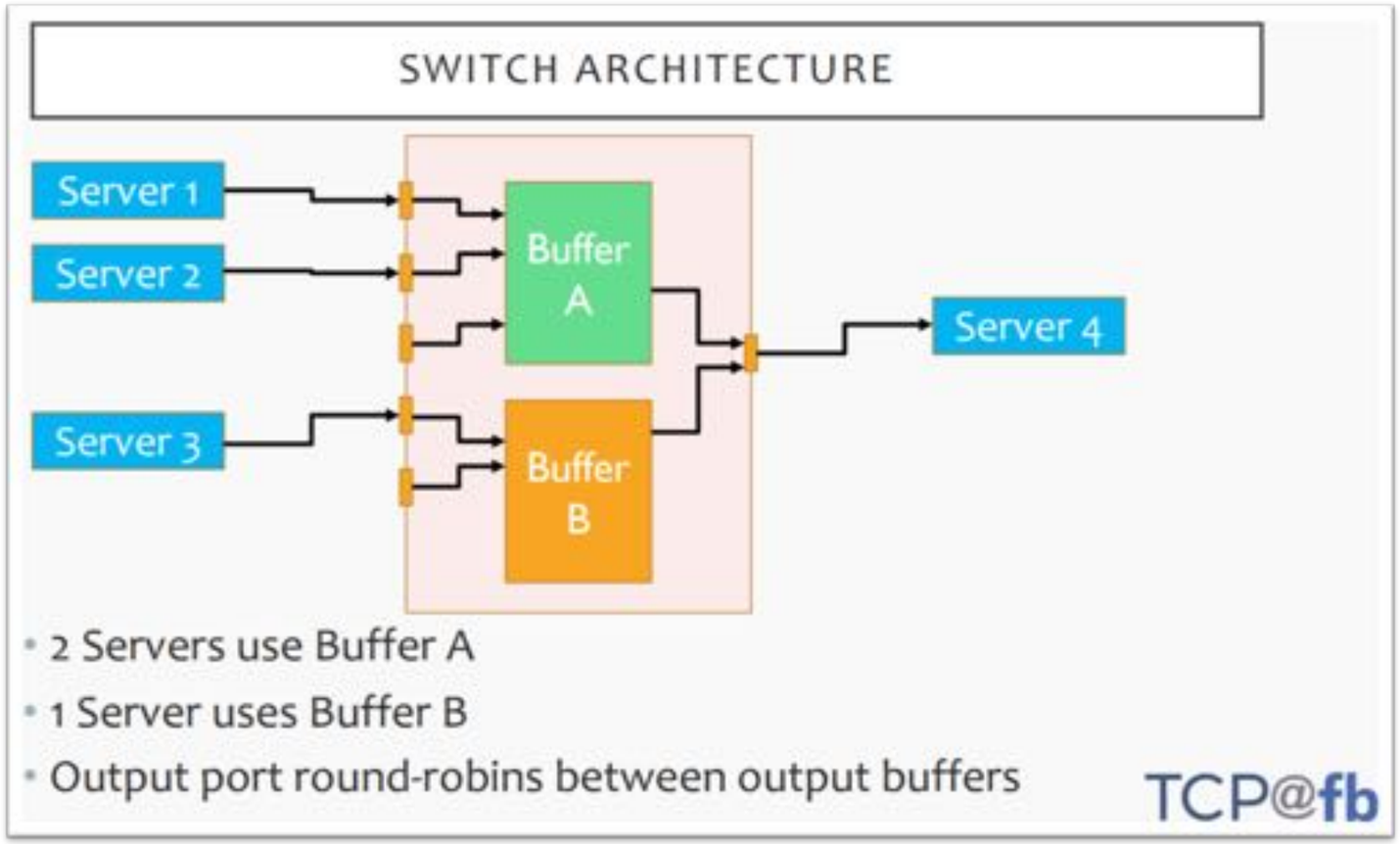
Spectrum's Fully Shared Buffer Provides ***4X effective buffer size***!

Competitor's multi-core based buffer scheme



Unfair & Unpredictable Performance

Source: Facebook Highlighting Fairness Issues



facebook

Unfairness between servers:
Servers 1 & 2 get 25% of bandwidth each
Server 3 gets 50% of all bandwidth

Unfairness between flows:
Flow 1 gets 23.0 Gbps
Flow 2 gets 0.5 Gbps

What Makes a Great RoCE Switch?

Simple Configuration

- One-command “Do RoCE” configuration
- Single pane-of-glass management

High Performance

- High throughput
- Low latency
- Zero packet loss

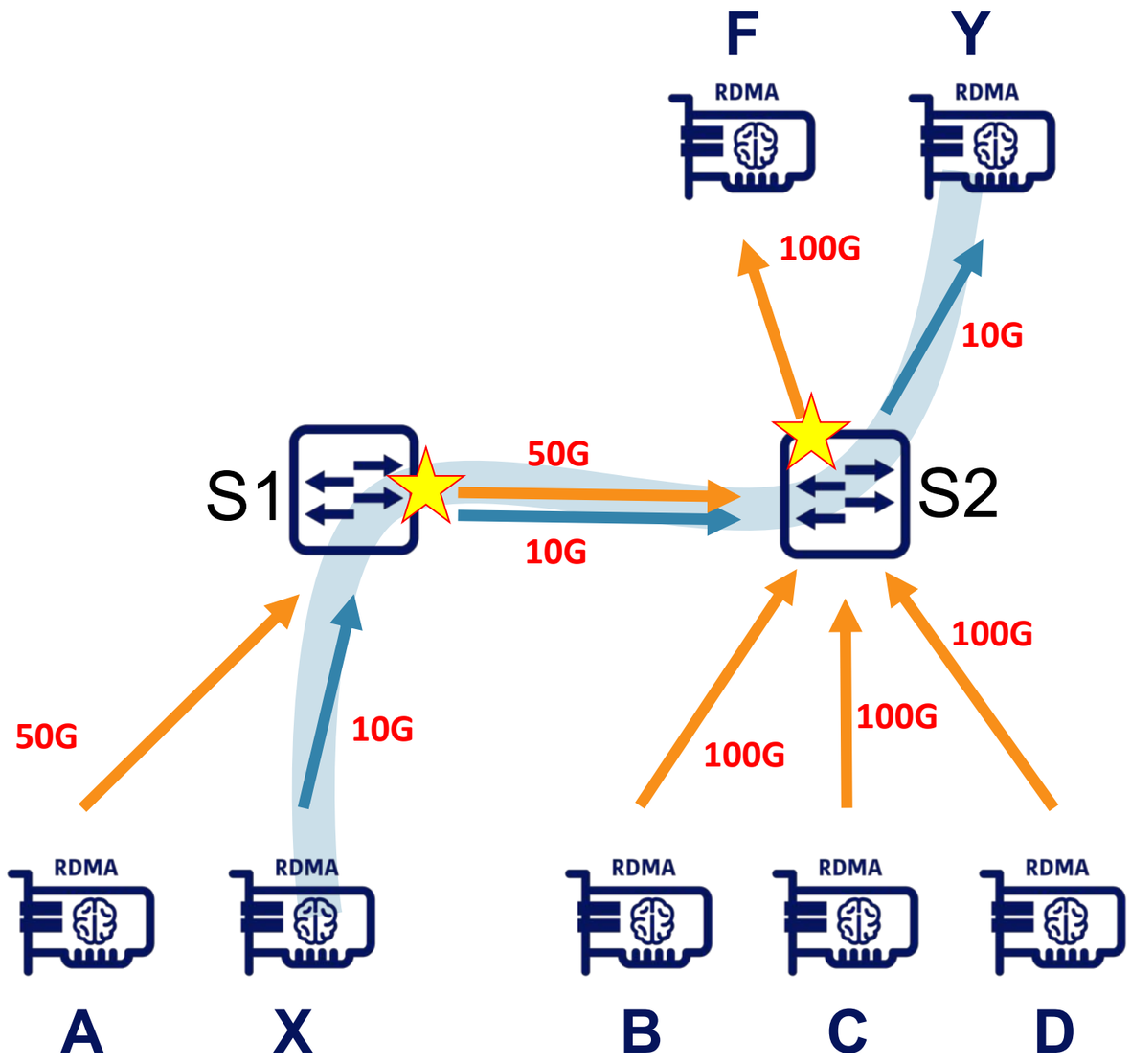
Advanced Congestion Control

- Early detection and prevention
- RoCE over VXLAN

Extensive Visibility

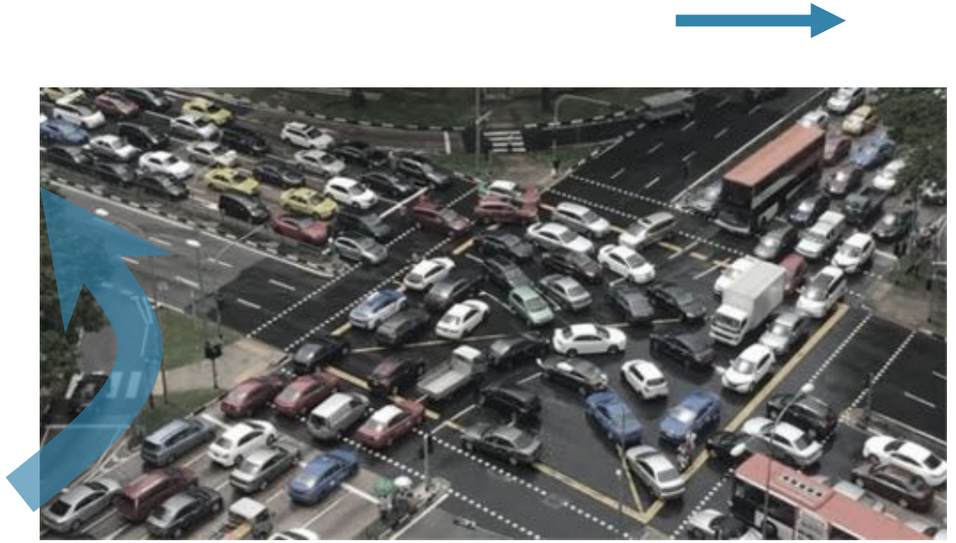
- Real time RoCE Telemetry
- Optimize network utilization
- Problem detection, prevention and troubleshooting

Scaling RoCE with ECN

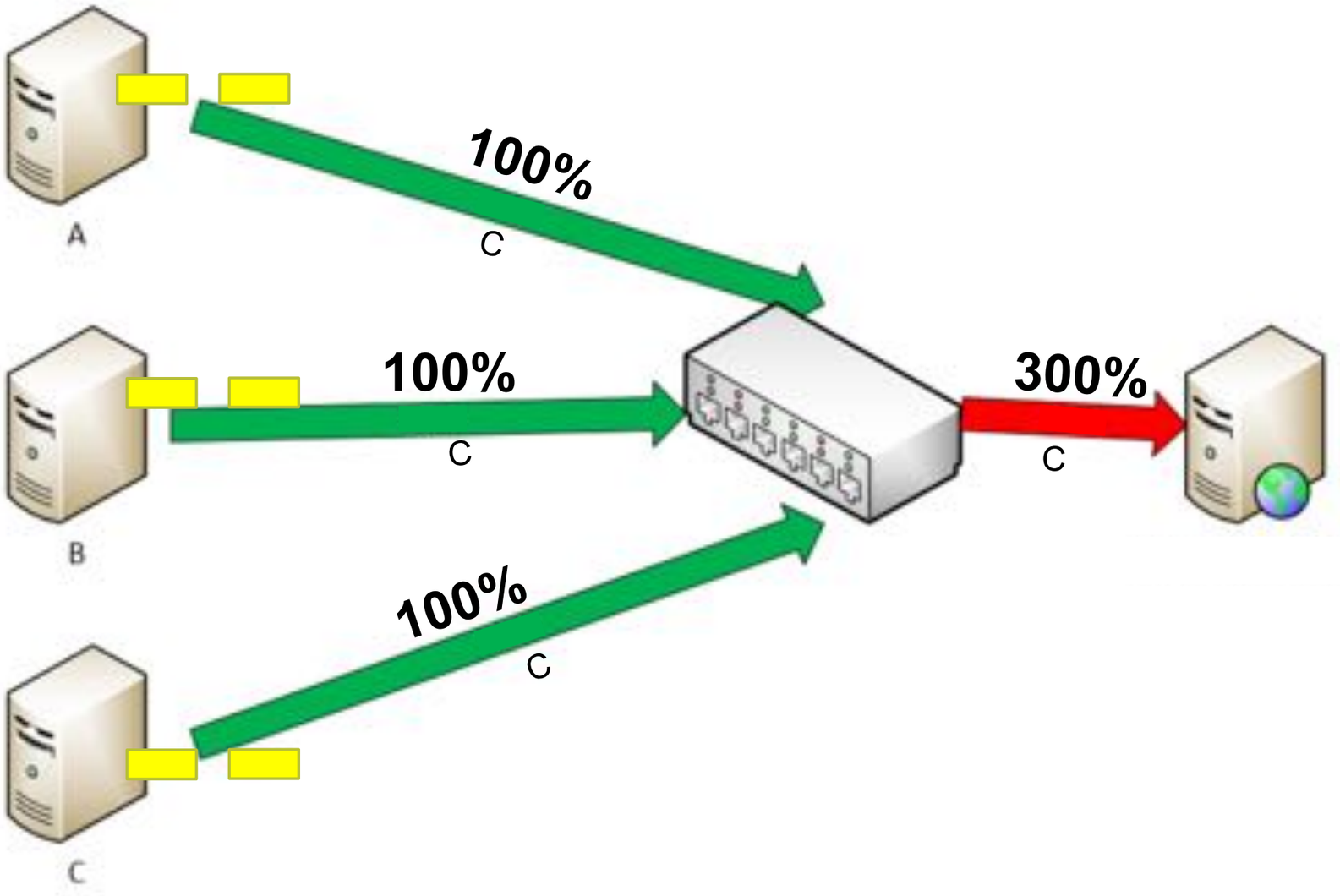


Explicit Congestion Control

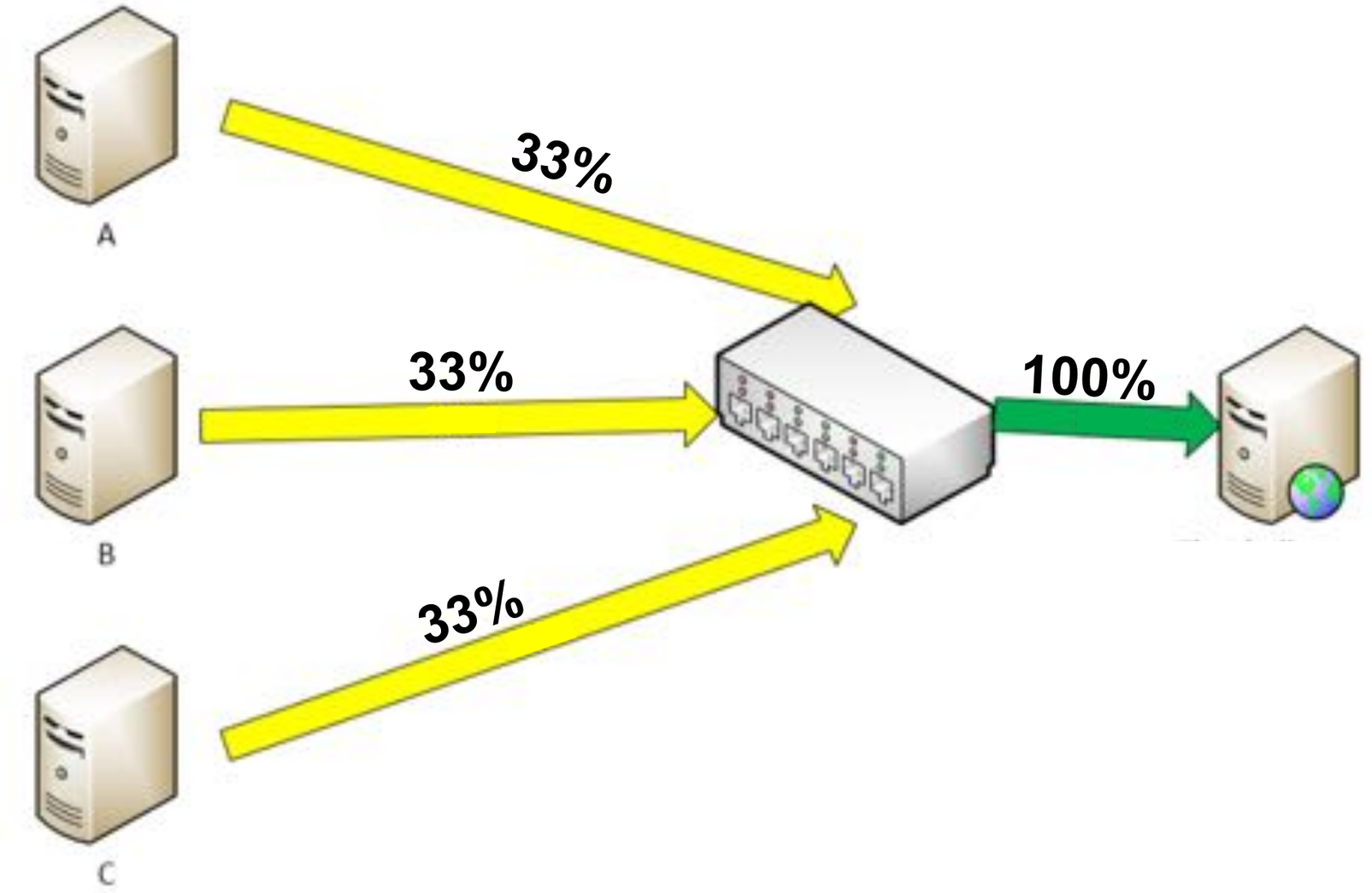
- ECN = Flow Level Congestion Handling
- ECN throttles A+B+C+D
- Victim traffic from X passes



Why congestion control is needed?



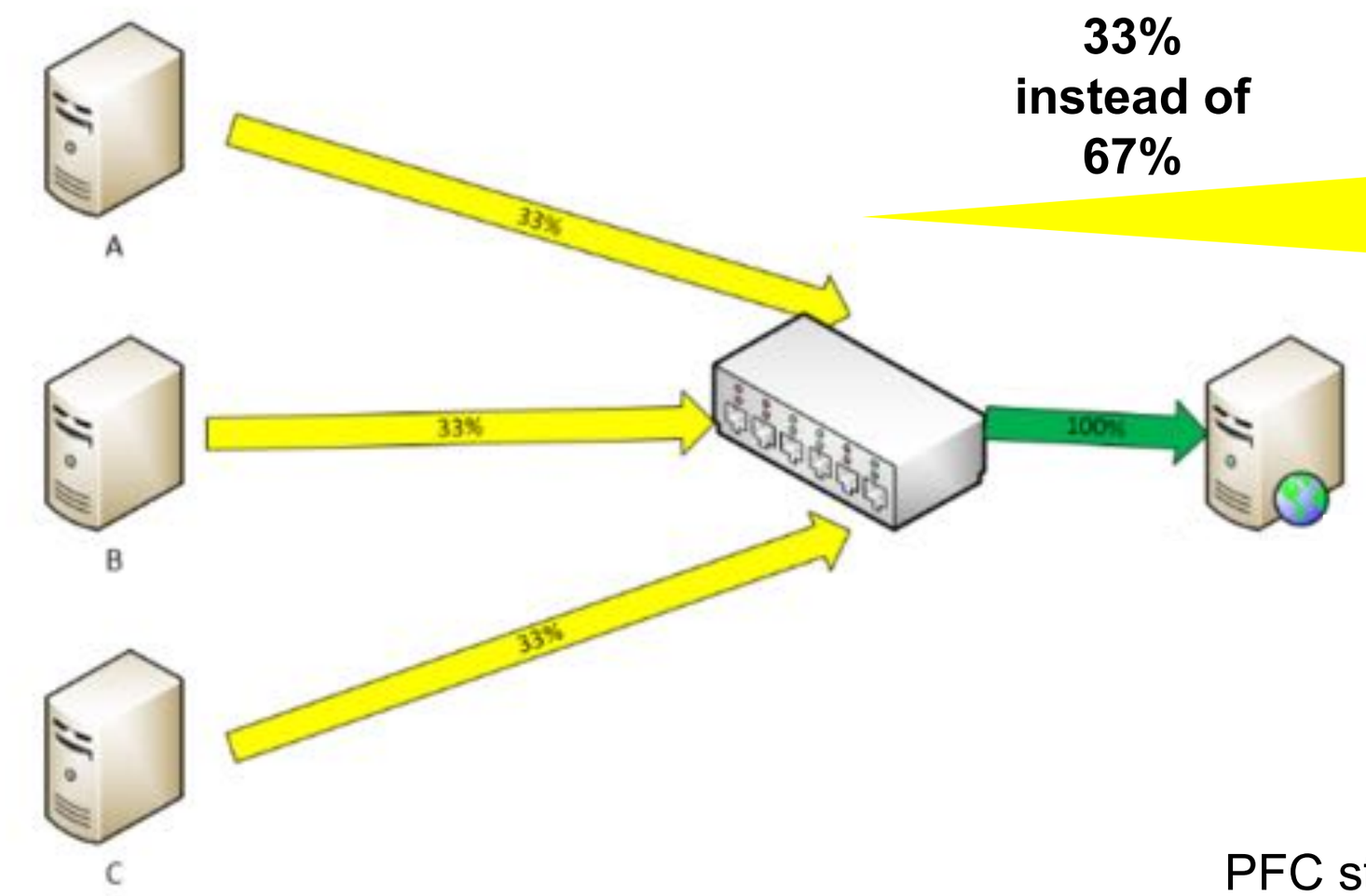
What perfect congestion control achieves?



Flow Control (PFC)

- Link layer protocol
  - Switch to neighboring switch/NIC. NIC to switch.
- When buffer fills up, the receiver sends **pause** frame to the sender.
- When buffer empties up, the receiver sends **unpause** frame to the sender.
- Pausing granularity is per priority
  - 8 priorities can be defined

PFC: Congestion Spreading Problem



33%
instead of
67%

This flow is also paused, since the pause control does not distinguish between flows.

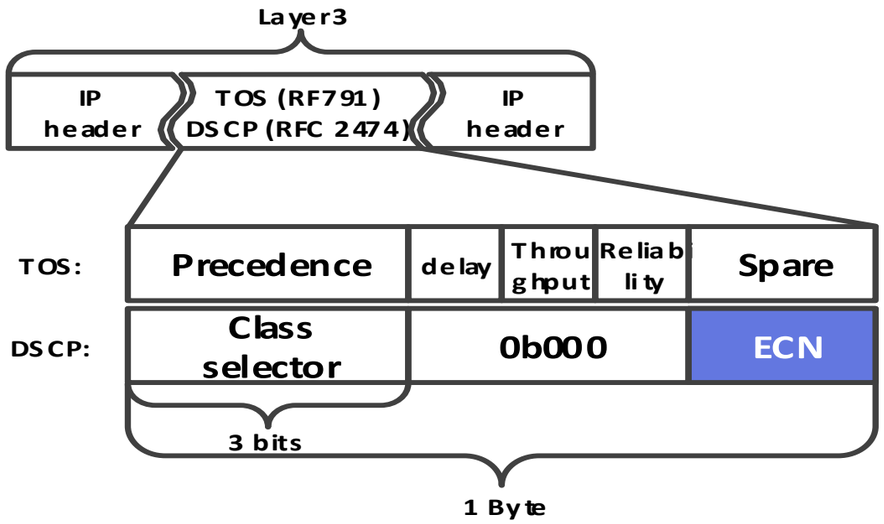
PFC stops links (priorities).
Congestion control stops flows.

Congestion Control and Flow Control

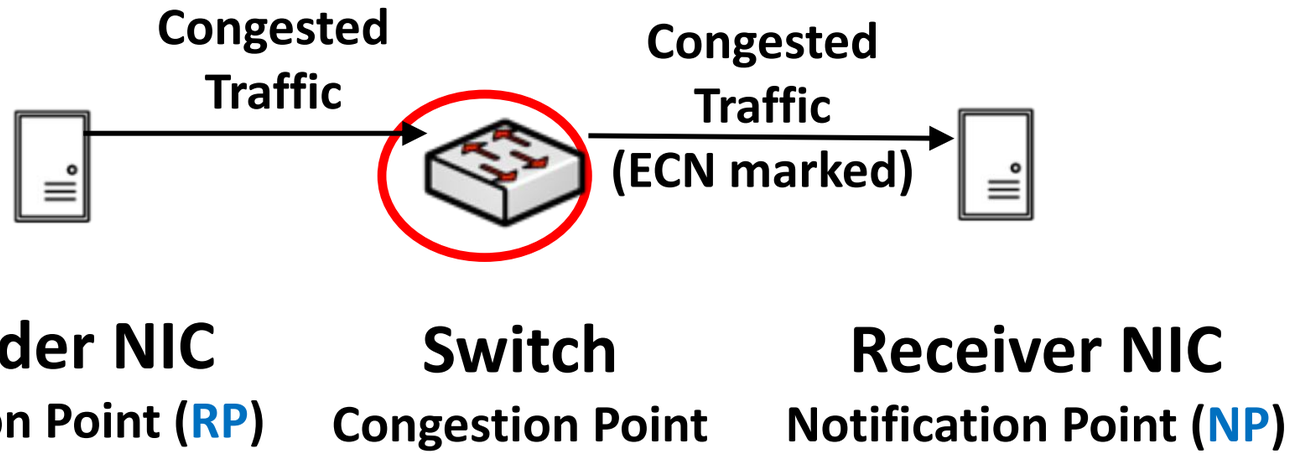
	Without flow control (PFC)	**With** flow control (PFC)
Without congestion control	Low performance due to many packet drops	Congestion might spread - Not recommended for large scale
With congestion control	Resilient RoCE - Easier to configure, but may cause slightly lower performance. - Congestion control alone reduces buffer overflows drops, but cannot prevent it.	Lossless RoCE - Recommended for large scale - Deployed today in production at scale

RoCEv2 Packet Format

ECN field in IP header is used to mark congestion (same as used for TCP)

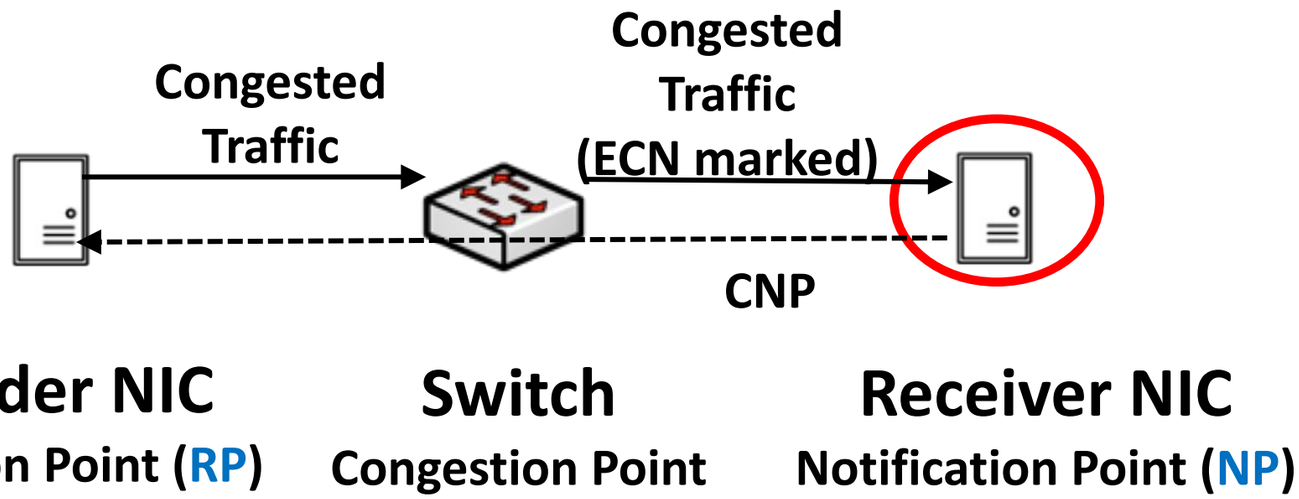


RoCE Congestion Control Algorithm: Congestion Point



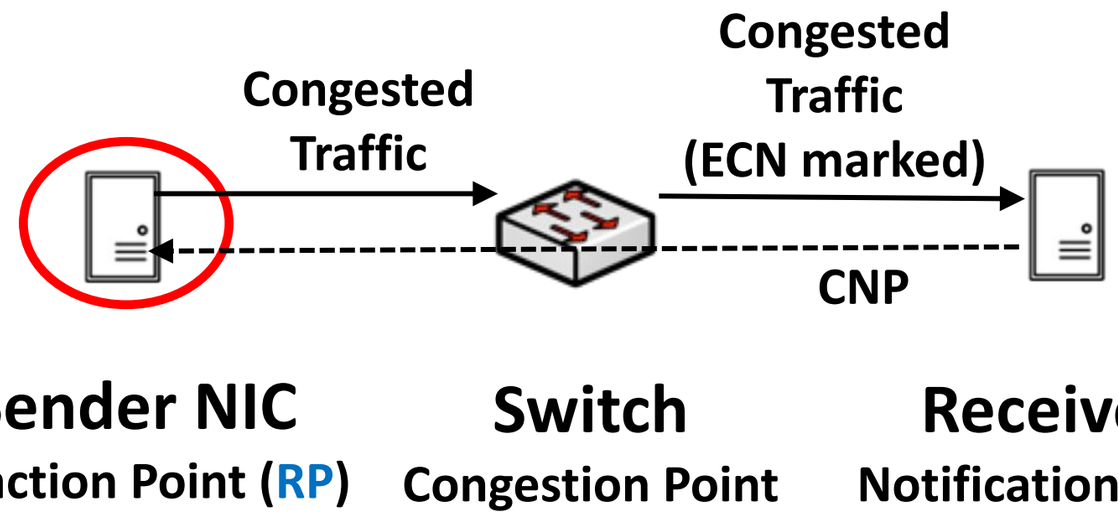
- Congestion Point (switch): marks ECN bits in packet header based on queue length
- Standard functionality supported by all commodity switches
  - also used for TCP

RoCE Congestion Control Algorithm: Notification Point



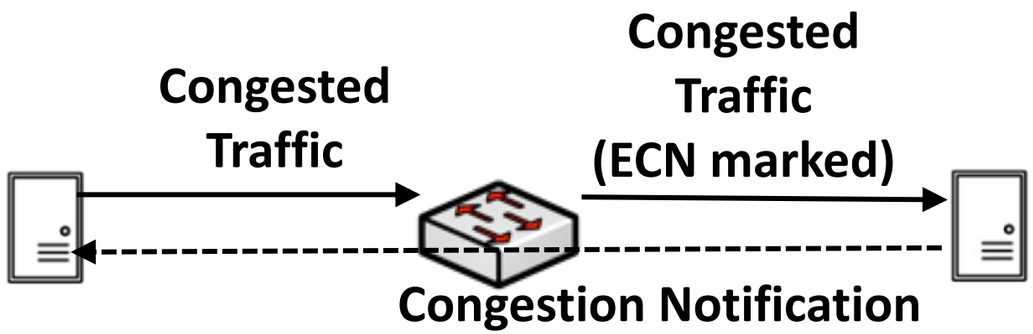
- Notification Point: If ECN-marked packet arrives, sends **CNP** (**C**ongestion **N**otification **P**acket) back to the sender
- CNP identifies a flow (QP)
- CNP generation is implemented by NIC HW
  - HW implementation provides fast response
  - CNP can be delivered via low latency path (guaranteed QoS)

RoCE Congestion Control Algorithm: Reaction Point



- Reaction Point: Throttles sending rate based on Congestion Notification Packets (CNPs) arrival
  - Also based on packet drop (planned)
- Implemented by HW
  - Fast response to congestion notification

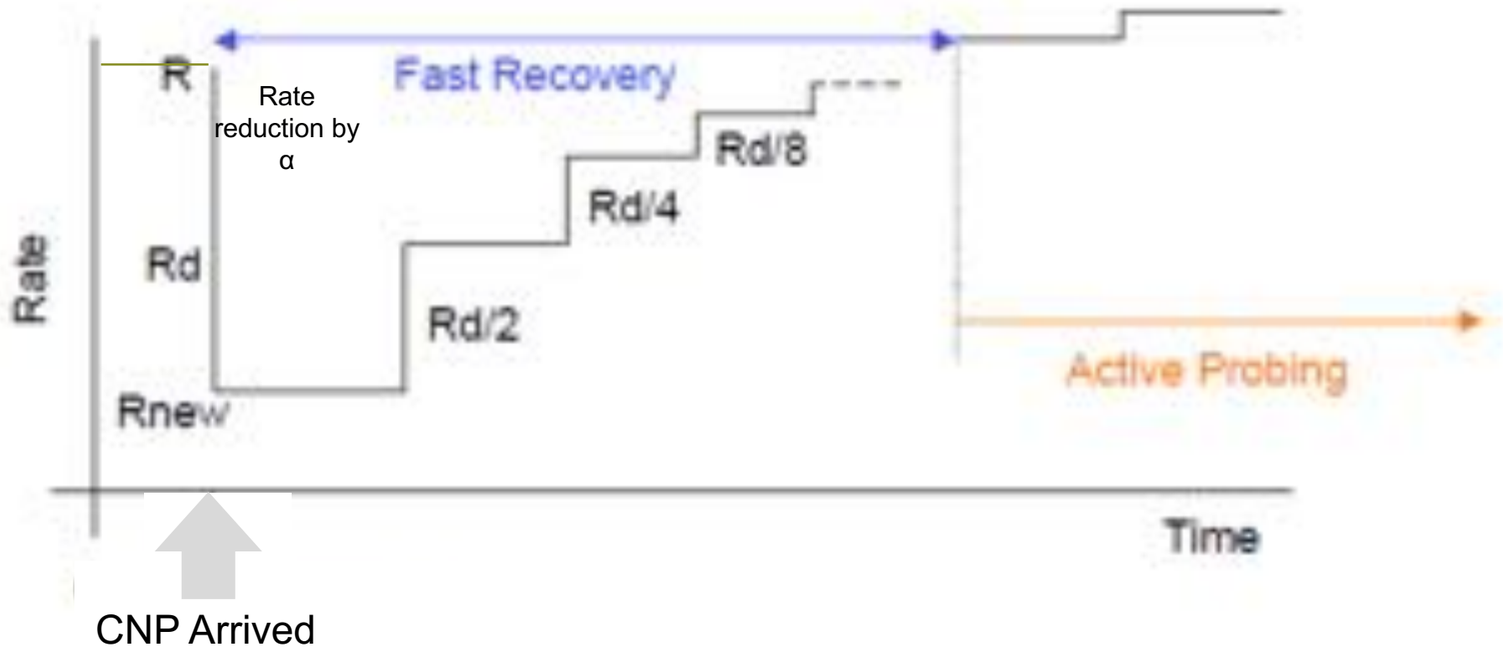
Hardware Based Congestion Control



- The novelty in ConnectX4: complete HW-based congestion control .
- Much faster than SW-based congestion control
  - HW based: 10's nanosec.
    - Immediately on the entire posted queue
    - Does not require SW intervention
  - SW/FW based: 100's microsec and more
    - Might be much longer due to length of posted queue
- Fast reaction to congestion notification minimizes the network congestion time
  - Congested switch buffers are less likely to overflow.

Rate Change Behaviour

- Maintains dynamic congestion estimation parameter (α, 0<α<1)
- When CNP arrives the rate reduced by α/2
- Fast Recovery phase : the rate is increased by half a way to the rate before congestion.
- Active Probing phase: the rate is increased constantly.



$$R_T = R_C,$$
$$R_C = R_C(1-\alpha/2),$$
$$\alpha = (1-g)\alpha + g,$$

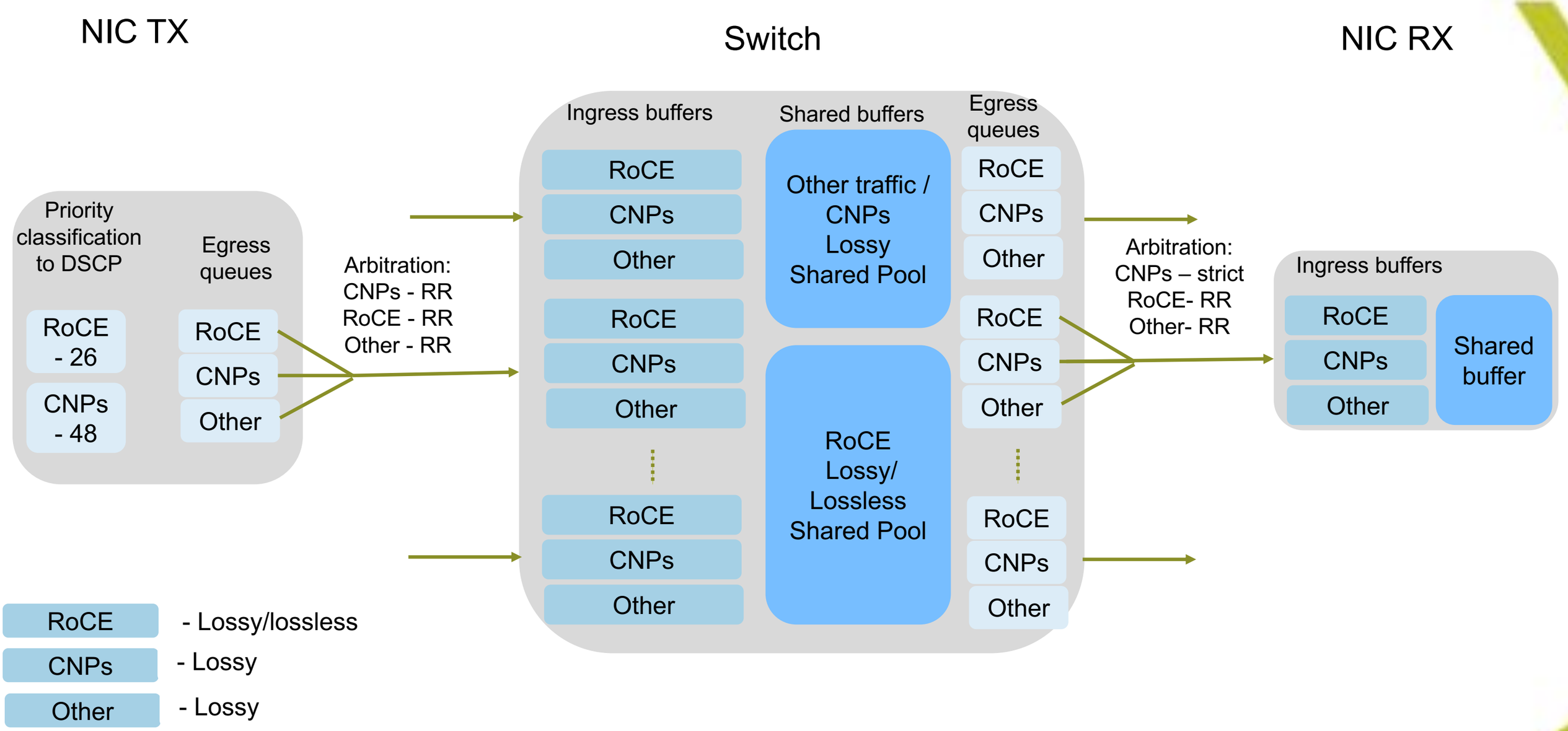
$$R_C = (R_T + R_C)/2,$$

$$R_T = R_T + R_{AI},$$
$$R_C = (R_T + R_C)/2,$$

Continuous update if no CNP arrived: $\alpha = (1-g)\alpha,$

Figure from: **Pan et al., QCN: Quantized Congestion Notification An Overview, 2007**

E2E QoS Configuration

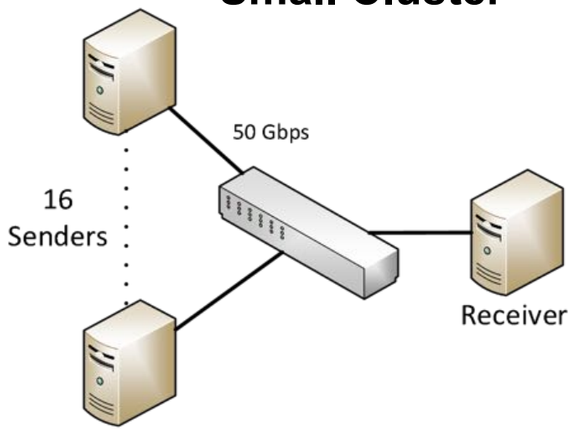


Test Results

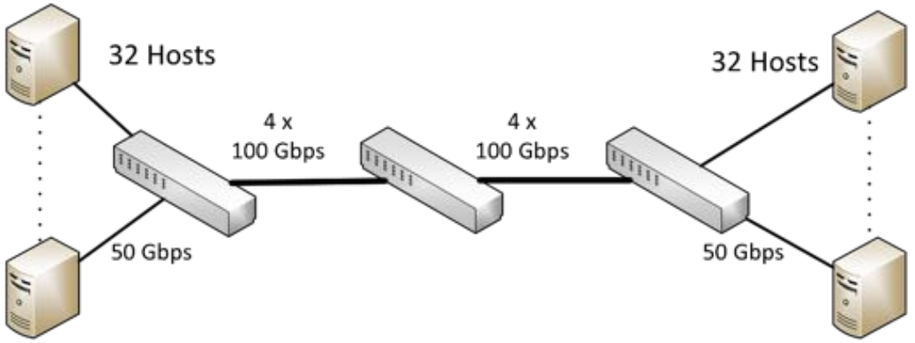
Setup

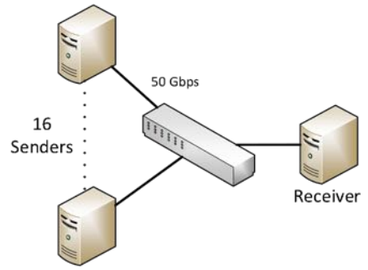
- Traffic Patterns:
  - Many to One
  - All to All
- Traffic/Network Configurations:
  - RoCE over lossless network
  - RoCE over lossy network
  - RoCE + TCP with priority separation
  - RoCE + TCP without priority separation
  - TCP only
- Tool: ib_write_bw / nd_perf
  - Streaming continuous traffic of Write Requests
- Driver: MLNX_OFED v. 4.0-1.6.1.0
- TCP stack: cubic (Linux Red Hat 7.0 defaults)
- Switch: Mellanox Spectrum

Small Cluster

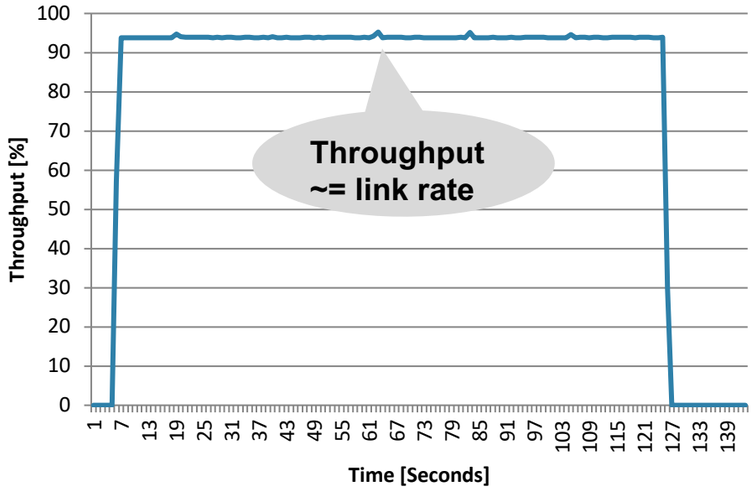


Large Cluster

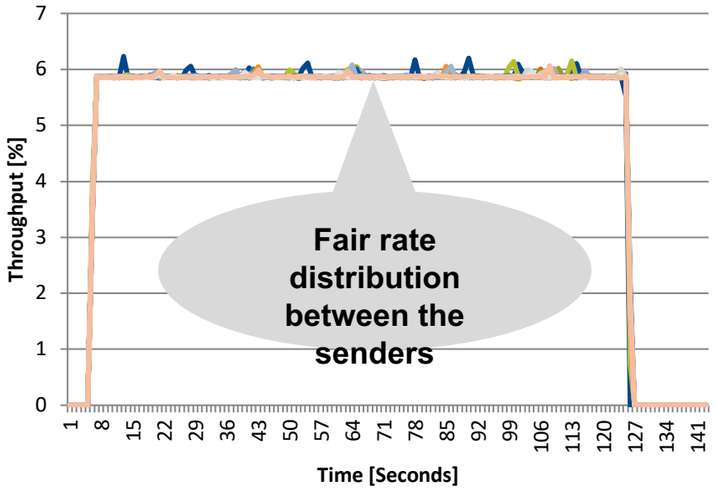




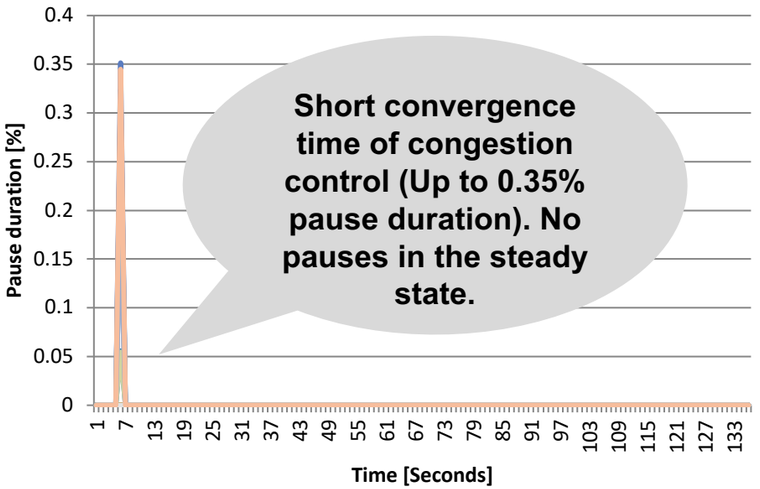
Total Throughput



Throughput per Sender

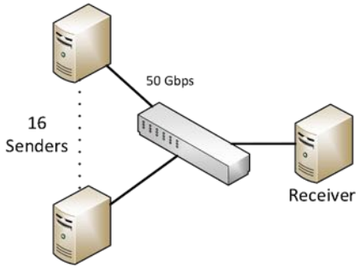


Pause Duration on Host

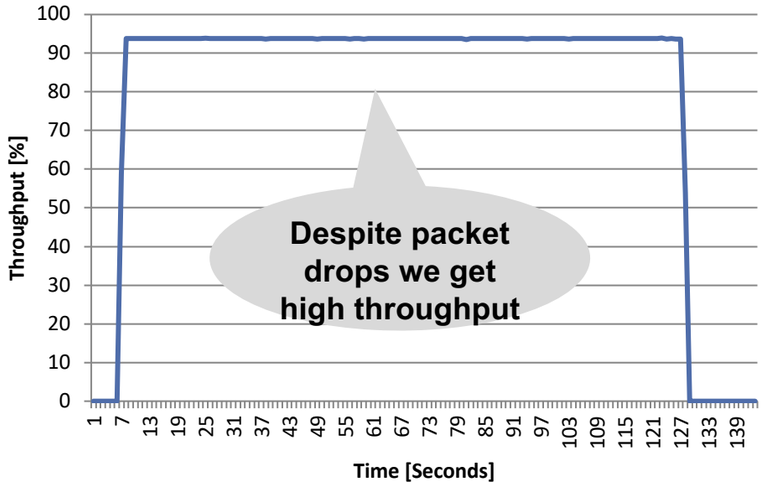


Under High Load – 512QP per sender

Large value was chosen to test system in stress

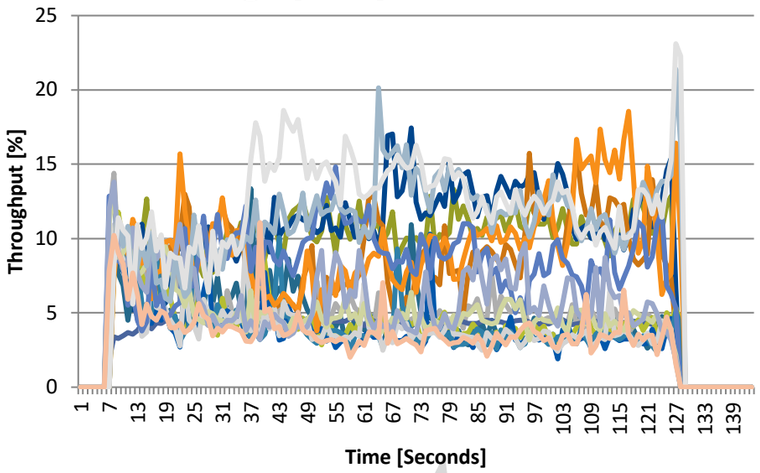


Total Throughput



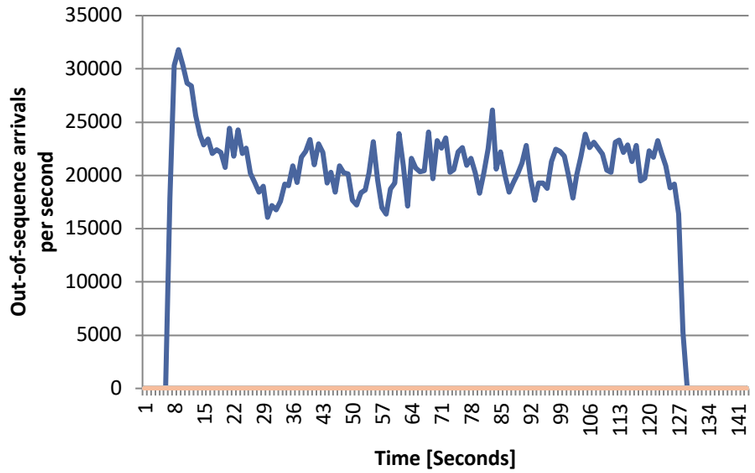
Despite packet drops we get high throughput

Throughput per Sender

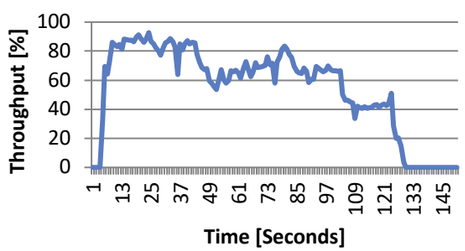
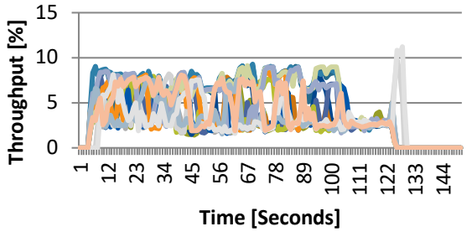
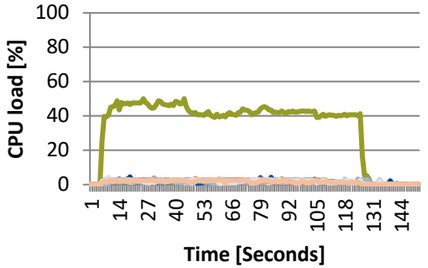
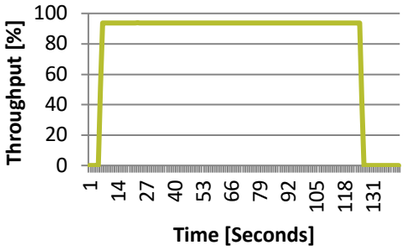
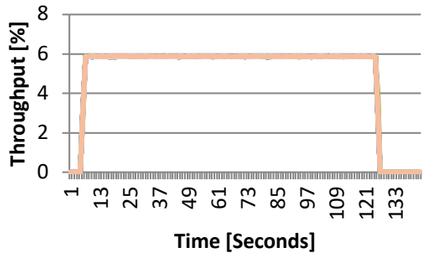
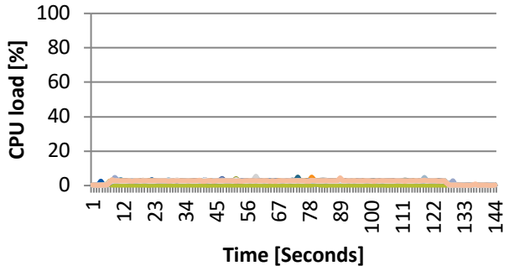


Losses cause unfair distribution. Issue to improve.

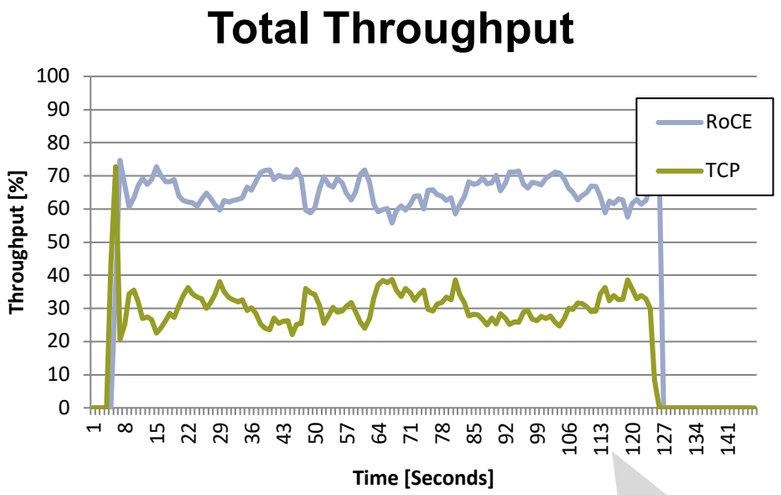
Out of Sequence Events (indicates packet drops)

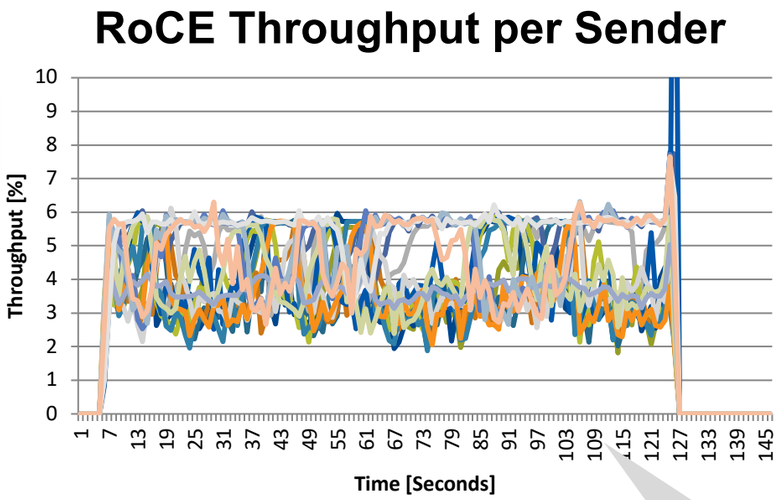


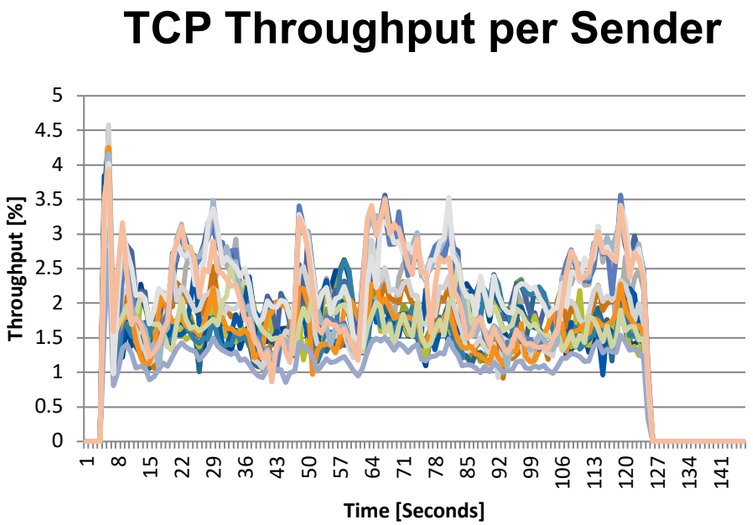
- 16 senders to 1 receiver
- 64 QPs per sender
- Lossy network

	Total Throughput	Throughput per Sender	CPU usage
Experiment 1: TCP			
Experiment 2: RoCE			
Conclusions:	**RoCE achieves almost twice larger throughput than TCP**	**RoCE achieves better fairness and less fluctuations than TCP**	**TCP requires high CPU usage, while RoCE requires negligible CPU usage**

- 16 senders to 1 receiver
- RoCE 64QP / TCP 32 flows
- Lossy network







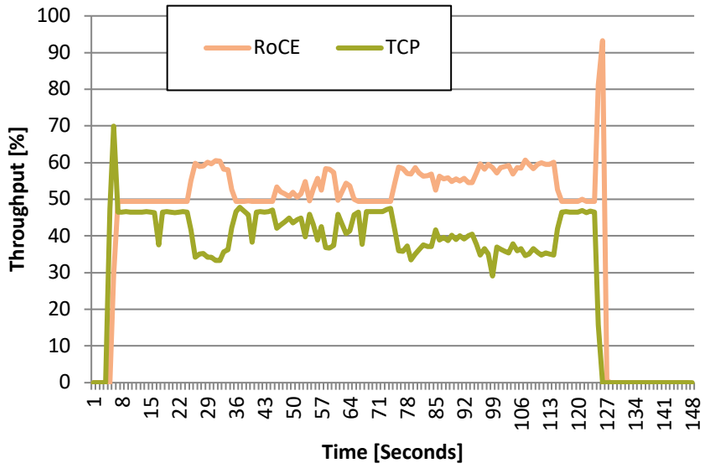
Achieved fairness by ratio of RoCE/TCP flows: more RoCE QPs, more RoCE BW

Background TCP traffic harms RoCE fairness

- 16 senders to 1 receiver

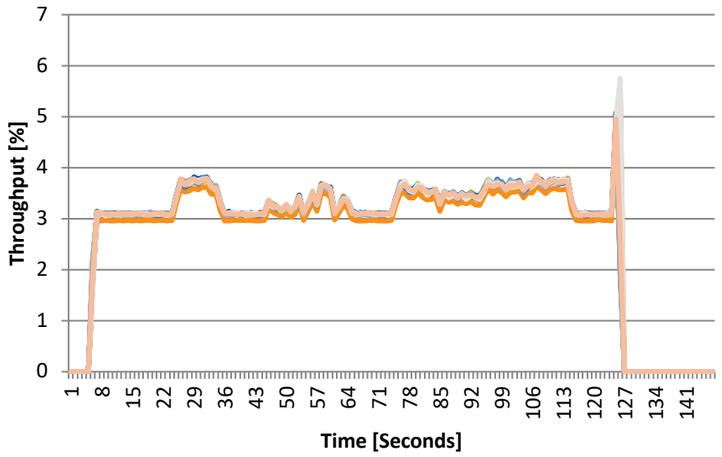
- RoCE on lossless: 64 QPs per sender
- TCP on lossy: 32 flows per sender

Total Throughput



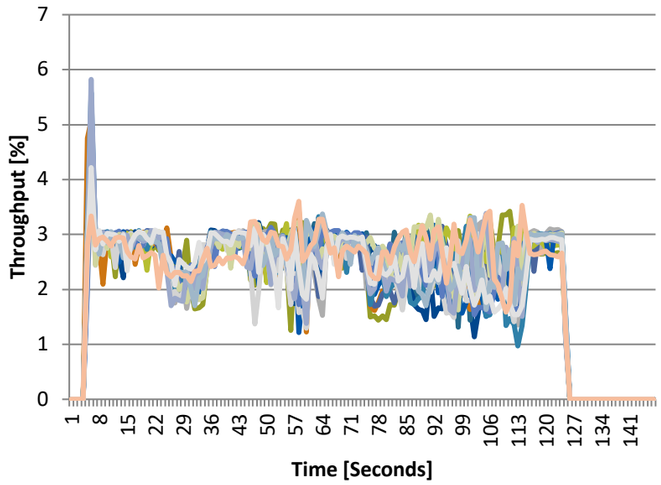
Fair share between two types of traffic: each one with ~half line rate BW
(oscillations due to under-flowing TCP queues)

RoCE Throughput per Sender



Inter-flow fairness in RoCE is preserved

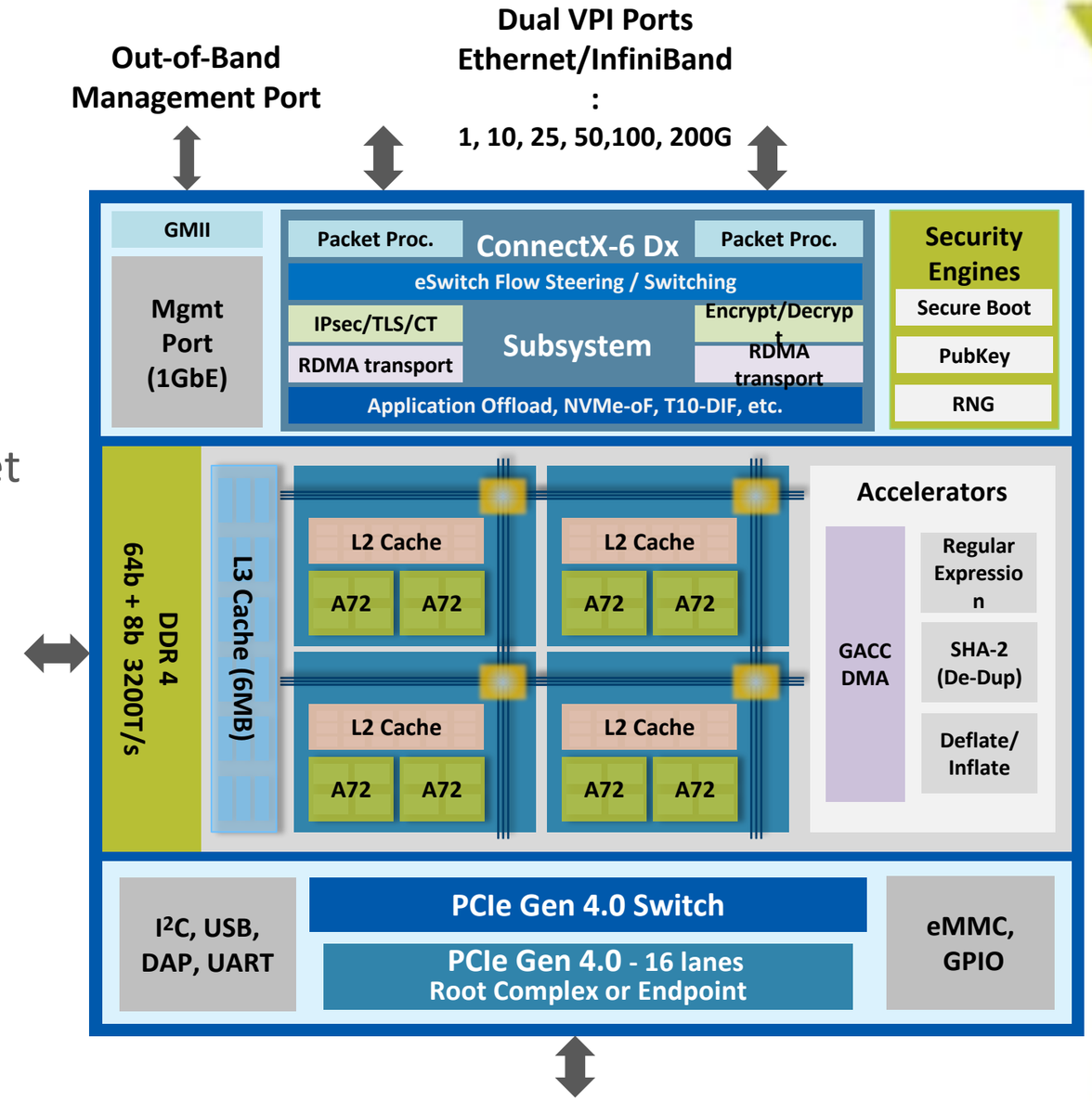
TCP Throughput per Sender



Next Generation System on Chips (Network SoCs)

Block Diagram

- Tile architecture running 8 x Arm ® A72 CPUs
  - SkyMesh™ coherent low-latency interconnect
  - 6MB L3 Last Level Cache
  - Arm frequency : 2GHz - 2.5GHz
  - Up to 32GB DDR4 @3200MT/s w/ ECC
- Up to 200Gb/s port bandwidth, InfiniBand or Ethernet
  - ConnectX-6 based
- Acceleration engines
  - ASAP2 switching and packet processing
  - NVMe SNAP™ storage emulation
  - IPsec/TLS data-in-motion and AES-XTS
  - Data-at-rest crypto accelerations
- Fully integrated PCIe switch
  - PCIe Gen3/4



The Ultimate Co-Processor

High Performance SoC

- Embedded ConnectX-6 Dx adapter
- Single and Dual port Ethernet & InfiniBand 10/25/50/100Gb/s, single port 200Gb/s
- PCIe Gen3/Gen4 x16, total throughput 200Gb/s
- 8 Arm® A72 CPUs @1.5GHz-2.5GHz
- One channel of DDR4 @3200MT/s

Advanced Hardware Accelerations

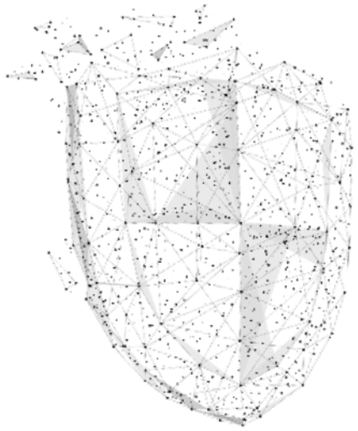
- Networking and virtualization accelerations – RDMA, ASAP^2, VirtIO , SR-IOV
- Security: Crypto (IPsec, TLS, AES-XTS), Isolation, Regular Expression & DPI
- Storage Accelerations – NVMe, (De)Compression, Dedup, RAID, CRC64
- Host agnostic network solution

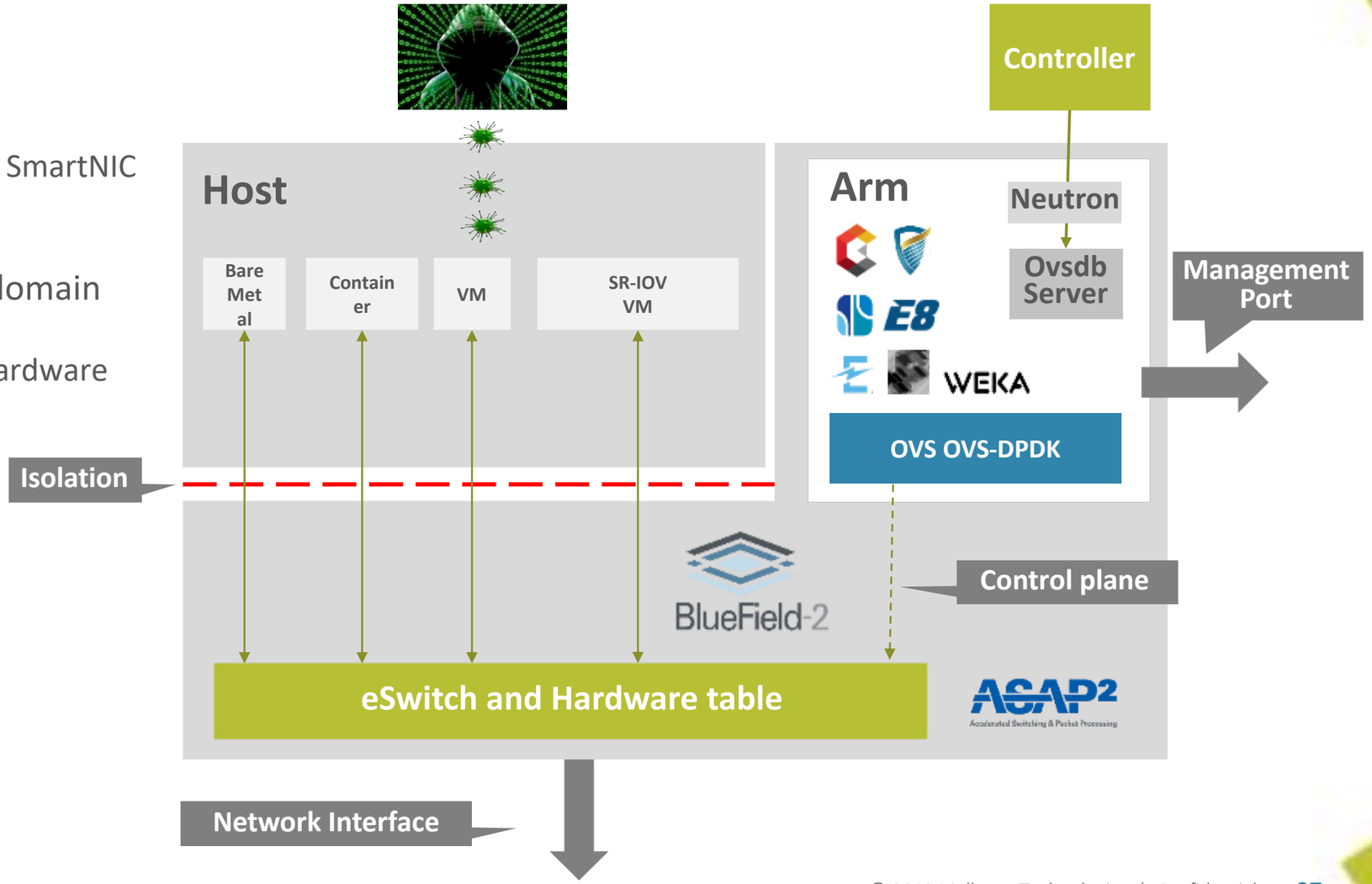
Scalability and Programmability

- For smartNIC and as a controller in system
- User specific application to run over Arm cores
- Integrated control and data planes
- For smartNIC: Security application isolated from main host

Functional Isolation with BlueField-2

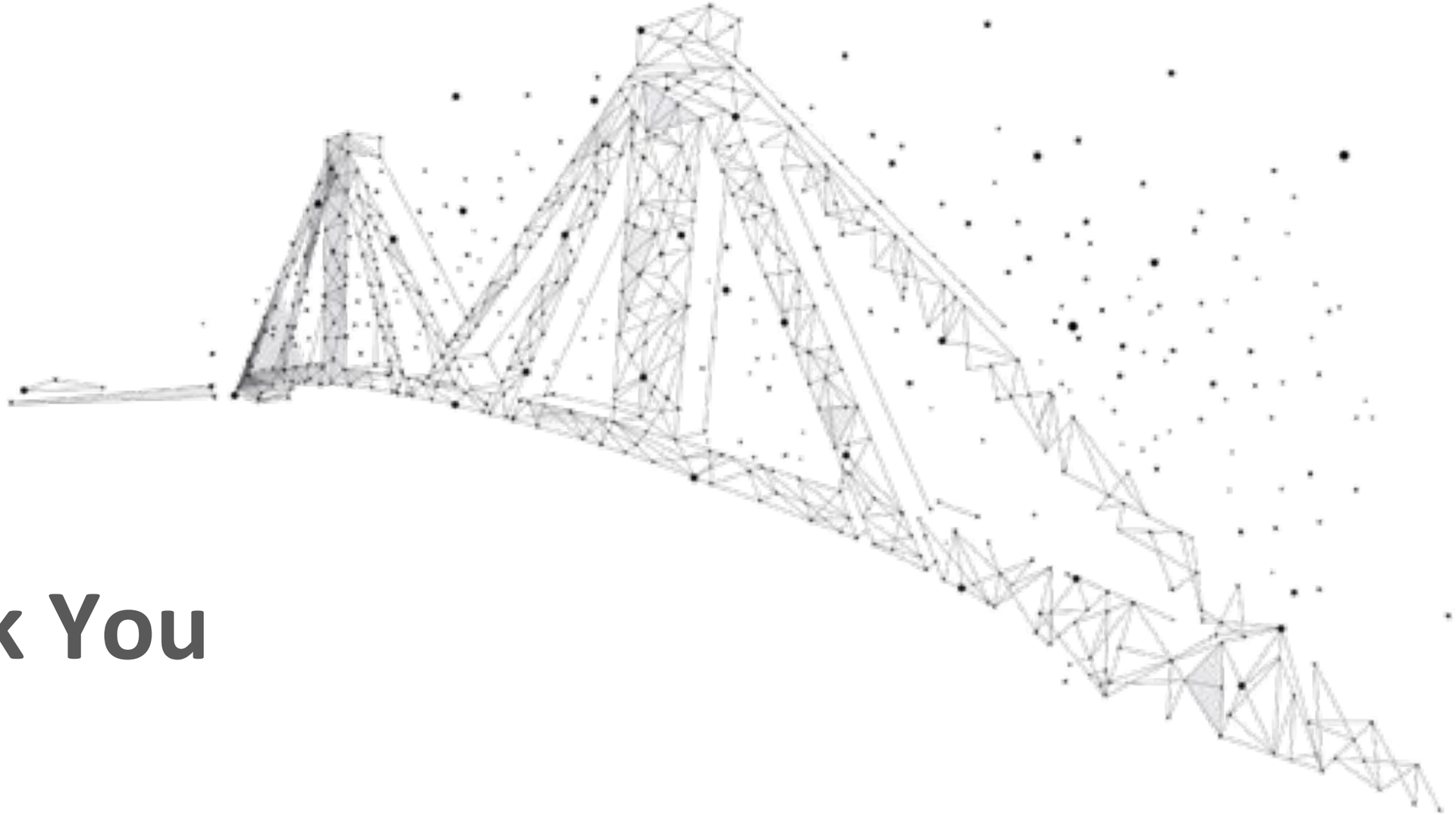
- A Computer in-front of a computer
- Isolation and Offload
  - Infrastructure functions fully implemented in SmartNIC
  - Networking, Security and Storage
- Functionality runs secure in separate trust domain
  - Enforces policies on compromised host
  - Host access to SmartNIC can be blocked by hardware





Boundaries and Distinction between pure Compute, Network and Storage elements is fading..

Intelligent Networks and Processing data ”on the move” will take us to Exascale and beyond..



Thank You