Follow-up Discussion of AI Computing Network Requirements

Jesus Escudero-Sahuquillo (UCLM)

Jose Duato (RSAC)

Who am I?

- **Associate Professor** at UCLM, Spain
- **Research and development in interconnection networks for 18 years** at different institutions: UCLM (Spain), Oracle (Norway), and UPV (Spain):
  - Solutions intended for specific network technologies (InfiniBand, Omni-path, BXI, Datacenter networks, etc.), while others could be quickly adopted.
  - Main R&D lines: congestion control, QoS, routing, and network topologies.
- Participated in **previous IEEE 802.1Q, NENDICA, and IETF** meetings (2018 and 2019) to support the Qcz amendment on CI, the congestion management applied to Lossless Ethernet:
  - https://www.ieee802.org/1/files/public/docs2018/cz-escuderosahuquillo-CIAnalysis-response-0518-v01.pdf
  - https://www.ieee802.org/1/files/public/docs2018/cz-escudero-sahuquillo-ci-internetworking-0718-v1.pdf
  - https://datatracker.ietf.org/meeting/105/materials/slides-105-hotrfc-7-strategies-to-drastically-improve-congestion-control-in-high-performance-data-centers-next-steps-for-rdma-00
  - https://mentor.ieee.org/802.1/dcn/19/1-19-0020-00-ICne-presentation-on-congestion-management-for-ethernet-based-lossless-datacenter-networks.pdf

Motivation

- Review the **major challenges** for the AI Datacenter network
- Discuss the **proposed solutions** and technologies to overcome the described challenges
- Analyze the **standardization opportunities** of the proposed solutions

“intelligent, high-performance data center networks enabling both HPC and mega data center workloads will be adopted in the industry soon”

T. Hoefler et al.: ***The Convergence of Hyperscale Data Center and High-Performance Computing Networks,*** *in Computer, vol. 55, no. 7, pp. 29-37, July 2022, doi: 10.1109/MC.2022.3158437*

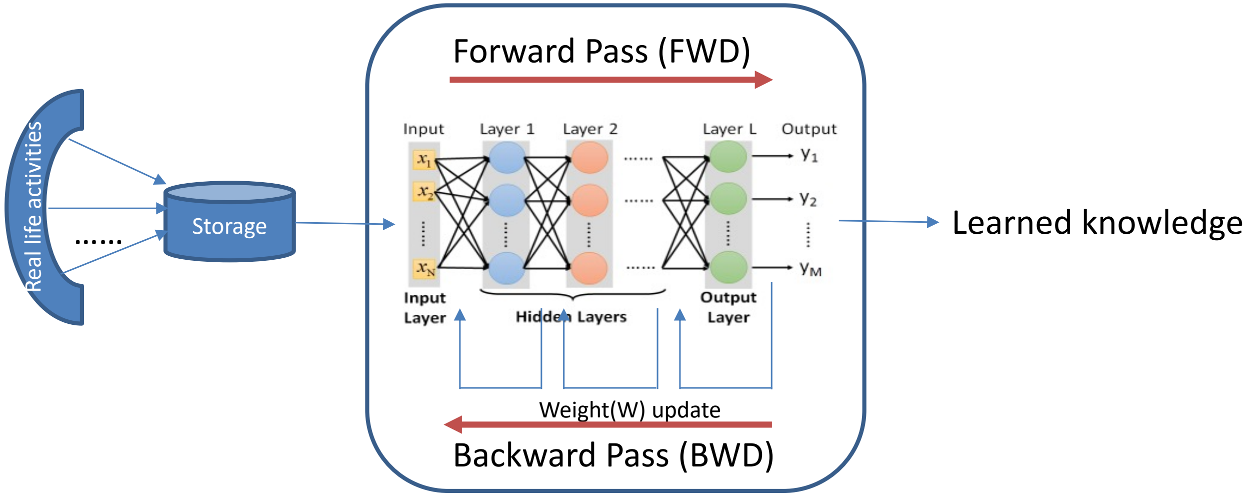
Expected Demand

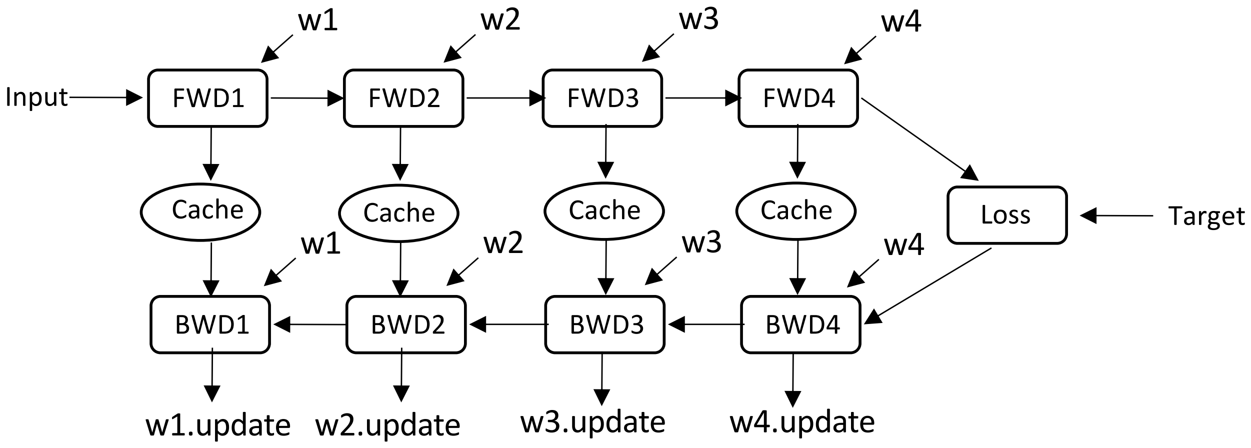
- The last decade has witnessed a very rapid expansion of many DNN-based AI solutions
- Regardless of where they are deployed, cloud datacenters are massively used for AI training
- The release of ChatGPT in Nov 2022 has garnered unprecedented attention, and triggered the recent boom of large language models (LLMs).

Model	**Falcon_40B**	**GPT3_175B**	**GPT4_1.8T**
Token Number	1 T	300 B	13 T
Training Time	2 months	34 days	100 days

- Huge datacenters are exclusively devoted to AI training and inference, and more are planned
- Expected size is on the order of 200K+ servers

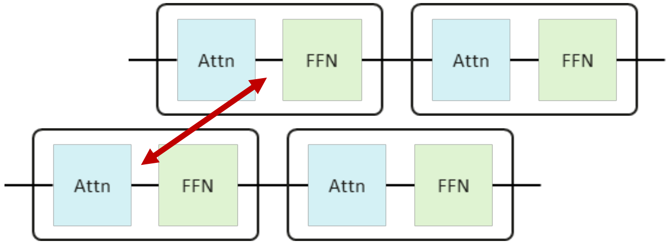
DNN-based AI Training





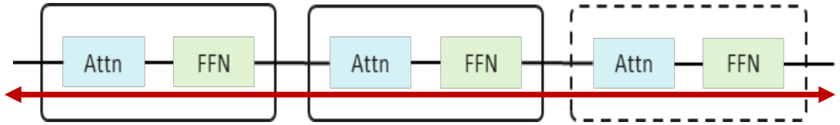
Parallelism in AI Training

- Data parallelism
  - Massive parallelism: Batches are independent from each other



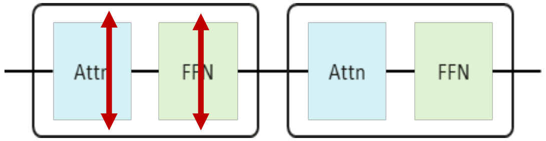
DP illustration in NN

- Pipeline parallelism
  - Implemented when model does not fit into CPU/GPU memory
  - It is indeed two pipelines in opposite direction, where each pair of stages (one from each pipeline) need to share memory
  - Implemented with a multicore CPU/GPU with half the cores devoted to each of the pipelines



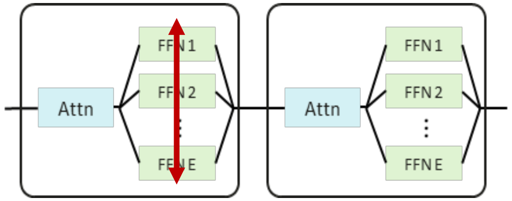
PP illustration in NN

- Tensor parallelism
  - Samples processed in batches (matrix-matrix instead of matrix-vector)
  - Tensor parallelism is critical to maximize data reuse, increasing performance and energy efficiency
  - Benefits of tensor parallelism are maximized through scale-up technologies



TP illustration in NN

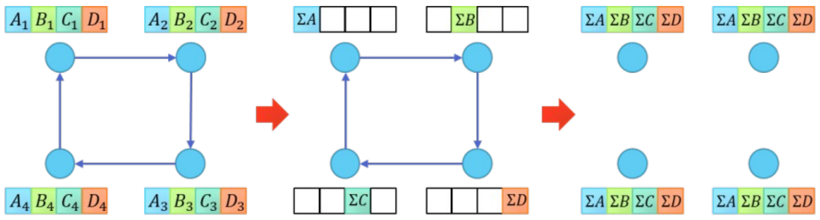
- Expert parallelism
  - Multiple experts are used to expand AI model parameters. Normally only one of a few of them will be running.



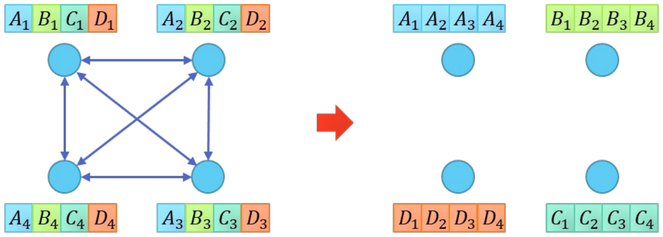
EP illustration in NN

Collective Communication in AI Training

- Collective communication is defined as communication that involves a group of processors. It used to be in MPI, including one to many, many to one or many to many communications.
- Modern distributed AI training relies on parallelism, that requires collective communication to achieve high performance.
- AllReduce and AlltoAll are typical collective communication operations in AI training.


A_1 B_1 C_1 D_1
A_2 B_2 C_2 D_2
A_4 B_4 C_4 D_4
A_3 B_3 C_3 D_3
ΣA
ΣB
ΣC
ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD

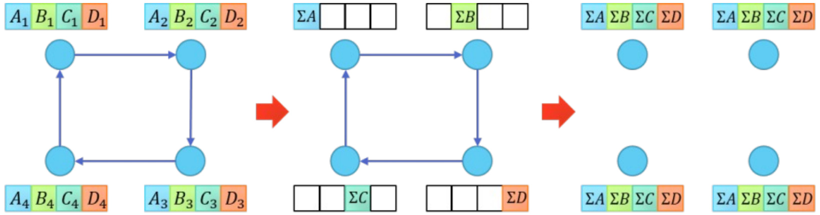
AllReduce



AlltoAll

Viable implementations - Topology and Collective Communication Optimizations

- A ring can be simply embedded into a switch
  - Multi-port NICs or multiple NICs per server may be needed to achieve the required bandwidth
  - Attaching servers to the same switch also helps reducing latency (assuming that the required number of servers does not exceed the number of ports)
- The reduce phase of AllReduce can be implemented in software (possibly, with support in the NIC) in log time with a fat tree
  - Recursive reduce. A tree is required for each reduction, but many reductions occur in parallel
  - The communication is faster if different servers collect the results for different reductions


A_1 B_1 C_1 D_1
A_2 B_2 C_2 D_2
A_4 B_4 C_4 D_4
A_3 B_3 C_3 D_3
ΣA
ΣB
ΣC
ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD
ΣA ΣB ΣC ΣD

- The broadcast phase of AllReduce requires a topology with full bisection bandwidth (fat tree)

Network requirements for AI datacenters

Let's consider a realistic scenario:

- The datacenter may not be exclusively devoted to AI training → several applications can be mixed with very different communication requirements.
- Task-to-server allocation and collective communication may not be fully optimized.
- Most importantly, for 200K+ servers, components will frequently fail

Network requirements for AI datacenters

What happens in this scenario?

- Application mix:
  - Not all traffic is based on collective communications
  - Network congestion and Head-of-line (HoL) blocking will occur
- Allocation and communication may not be optimized:
  - Unbalanced resource utilization
  - Likely, network congestion and HoL blocking
- Components will frequently fail:
  - Solutions are required: combination of hot swap, automatic path migration (APM), and checkpointing
  - Those solutions (especially APM) will unbalance traffic

Viable implementations to meet AI training

- Load balancing:
  - Load-aware packet-level load balancing mechanisms will significantly help to eliminate bottlenecks and fully utilize existing bandwidth
  - It is mandatory when implementing APM to balance traffic among the remaining healthy paths
- Adaptive routing with congestion control:
  - Adaptive routing may be used together with load balancing to alleviate in-network congestion further, especially when produced by faulty components
  - Adaptive routing should only be used for in-network congestion, but never for incast congestion
  - Thus, incast congestion still requires congestion control
  - Incast congestion will likely occur during AllReduce

Viable implementations to meet AI training

Limitations of load balancing:

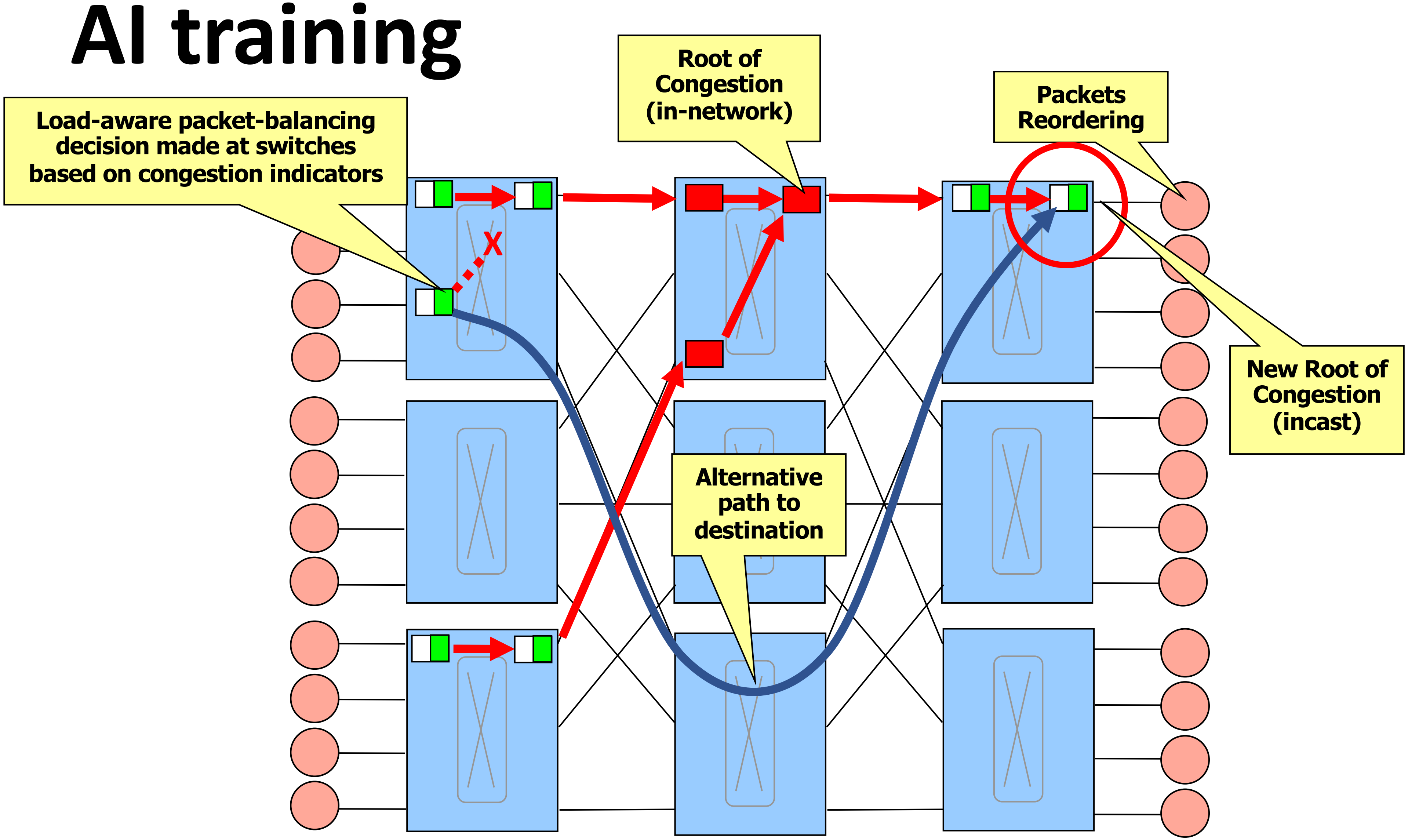
- Technique to **avoid in-network congestion**.
- Ineffective approaches can do the opposite.
- Load balancing **selects a path by hashing the flow identity fields** in the routed packet such that all packets from a particular flow traverse the same route.
- Equal Cost Multi-Path (ECMP) routing: **Flow granularity is a problem** that may cause elephant flows to traverse and occupy a route in the network for a longer time.
- Solution: Load-aware packet-level load balancing

Viable implementations to meet AI training

- Reducing the granularity from flows to packets to make better load-balancing decisions.
  - Solution: Load-aware packet-level balancing
- Issues with the uniformity of traffic flow distribution and in-order delivery
  - Solution: Intelligent packet reordering and selective retransmissions
- Balancing congested packets through alternative routes may end up moving congestion roots near end nodes, transforming in-network congestion into incast congestion → **The congestion spreading problem**

Rocher-Gonzalez, J., Escudero-Sahuquillo, J., Garcia, P.J., Quiles, F. ***On the Impact of Routing Algorithms in the Effectiveness of Queuing Schemes in High-Performance Interconnection Networks****. In Proc. of IEEE HoTI 2017.*

Viable implementations to meet AI training


Load-aware packet-balancing decision made at switches based on congestion indicators
Root of Congestion (in-network)
Packets Reordering
X
New Root of Congestion (incast)
Alternative path to destination

In-network congestion in a 3-tier CLOS evolves to incast due to multi-path routing

LB/AR/CC cooperation

- To deal with the **congestion-spreading problem**, we proposed to avoid routing congesting flows through alternative routes
  - Single-path (deterministic) routing is used for congesting flows
  - Multi-path (LB or AR) routing is used for non-congesting flows
- The **evolution of congestion trees** depends on the traffic patterns, network topology, and routing and needs to be thoroughly analyzed [Garcia19Nendica]:
  - It is the basis for efficient HoL-blocking elimination.
- Solution: Multi-path routing combined with CC that distinguishes between in-network and incast congestion

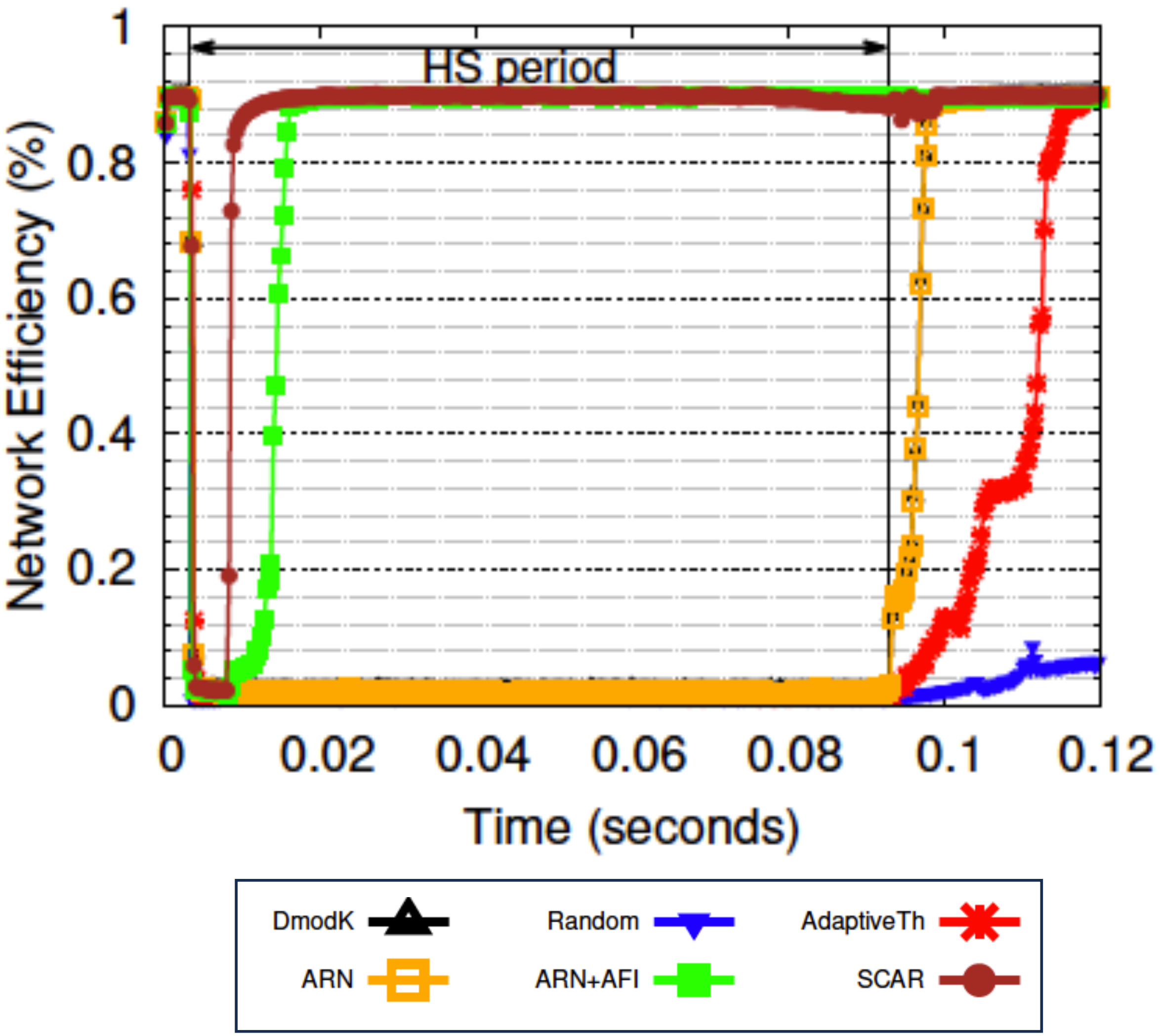
J. Rocher, J. Escudero Sahuquillo, P.J. Garcia, F.J. Quiles and J. Duato: ***A Smart and Novel Approach for Managing Incast and In-Network Congestion Through Adaptive Routing*** (May 10, 2023). Pre-print available at: http://dx.doi.org/10.2139/ssrn.4660017

LB/AR/CC cooperation

- Congestion is detected at switches based on queuing occupancy, which triggers adaptive routing.
- Notifications must be sent between switches (as InfiniBand does with ARNs and CI with CNPs).
- Based on notifications, switches use adaptive routing to alleviate in-network congestion or deterministic routing when an incast is notified.
- HoL blocking can be avoided using CI.
- Challenge: AR+CC cooperation with intelligent LB

J. Rocher, J. Escudero Sahuquillo, P.J. Garcia, F.J. Quiles and J. Duato: ***A Smart and Novel Approach for Managing Incast and In-Network Congestion Through Adaptive Routing*** (May 10, 2023). Pre-print available at: http://dx.doi.org/10.2139/ssrn.4660017

LB/AR/CC cooperation

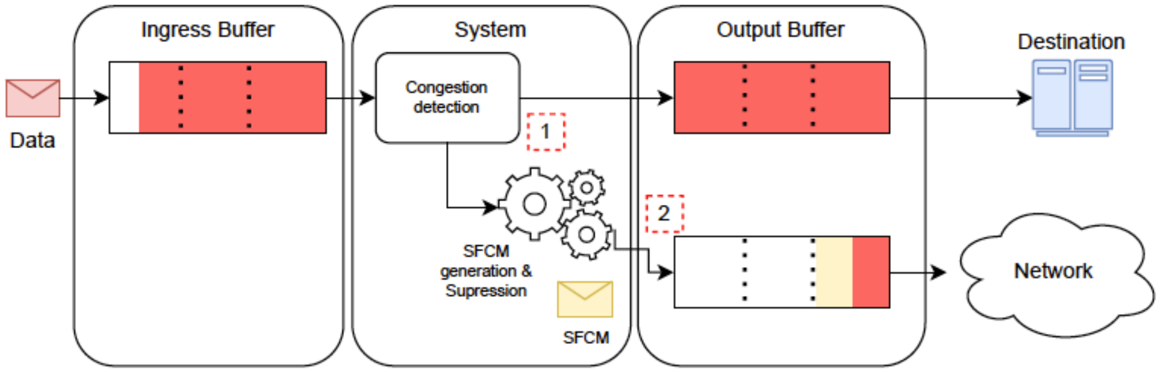


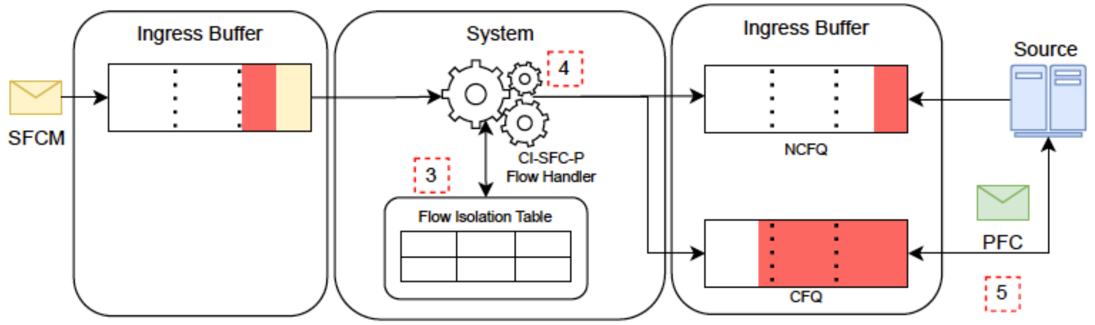
LB/AR/CC cooperation

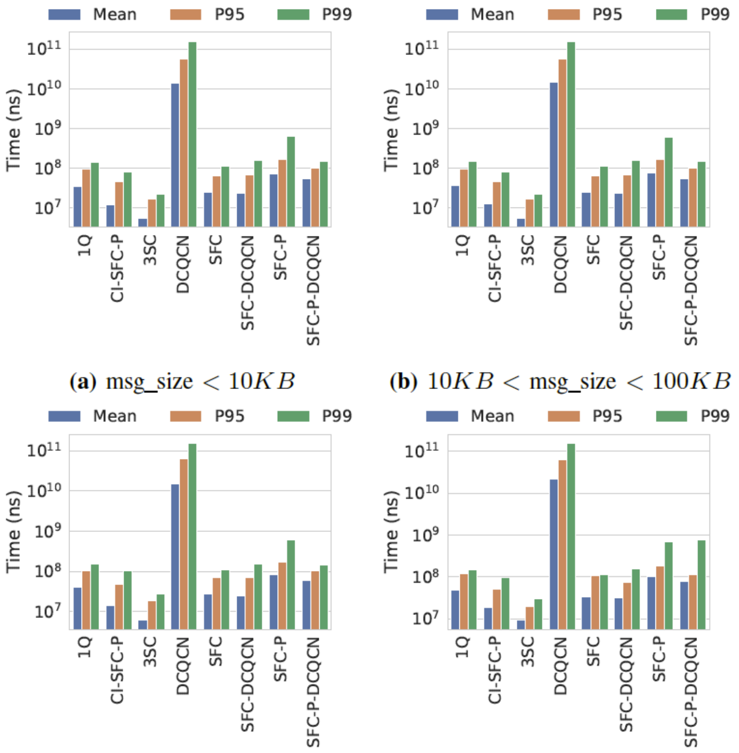
- Congestion Isolation (CI) deals with congesting flows and marks packets so they cannot be routed using adaptive routing
  - It also entirely avoids HoL blocking.
- Non-congesting packets are routed using either LB or AR.
  - Intelligent LB can be used if APM reacts to network failures.
  - Analyze congestion trees' evolution and traffic patterns on the fly to select between LB and AR:
    - LB is better suited for regular, massive traffic.
    - AR is best suited for very random or time-varying traffic.
    - Network load may vary so fast that load-aware LB may need to adapt faster. In that case, AR achieves a very fast local response and quickly avoids rapidly arising congestion scenario.

Other examples of cooperation

- 3SC: Combination of SFC, CI, and DCQCN







(a) msg_size $< 10KB$ **(b)** $10KB <$ msg_size $< 100KB$

Potential standardization opportunities

- Cooperation between protocols, if done correctly, benefits network performance.
- CI is in the standard. SFC standard is in progress. LB, AR, and CC are implemented with vendors but are not included in the standard yet.
  - Even LB with intelligent reordering and selective retransmissions can be used to cooperate
- CC/AR coordination is possible using fast status feedback of link/port/queue.