White Paper

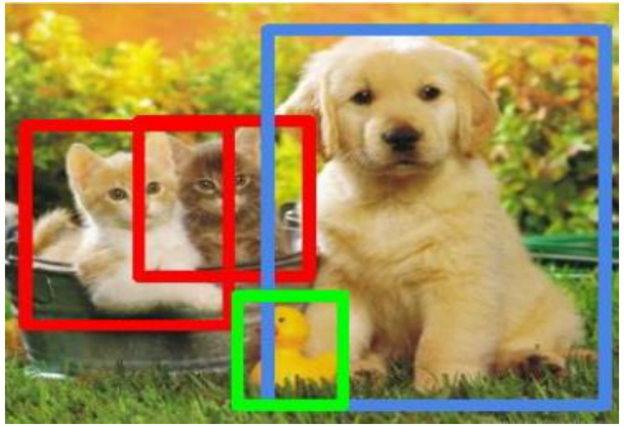
The Lossless Network

in the Data Center

IEEE 802 Industry Connections, November 2017
Paul Congdon

Our Digital Lives are driving Innovation in the DC

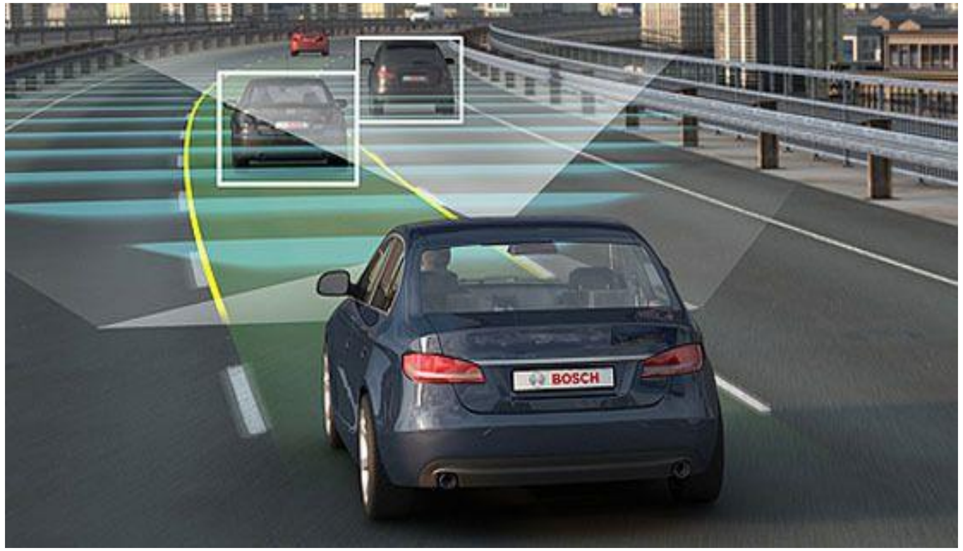
Interactive
Speech
Recognition



Interactive
Image
Recognition

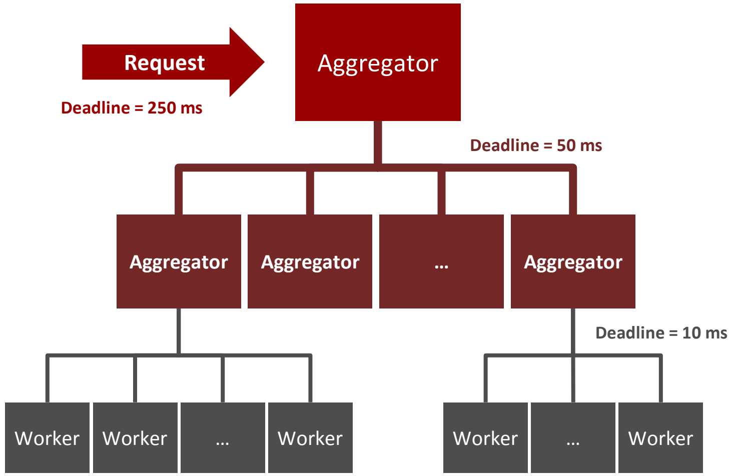
Human / Machine
Interaction

Autonomous
Driving



Three Critical Use Cases

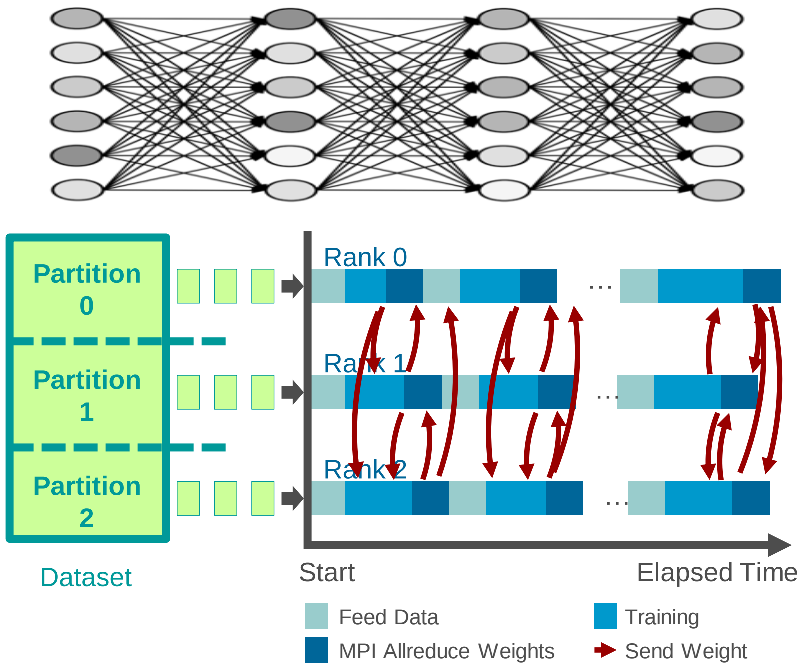
Online Data Intensive (OLDI) Services



Tail Latency is Critical

OLDI applications have real-time deadlines and run in parallel on 1000s of servers. Incast is a naturally occurring phenomenon. Tail latency reduces the quality of results

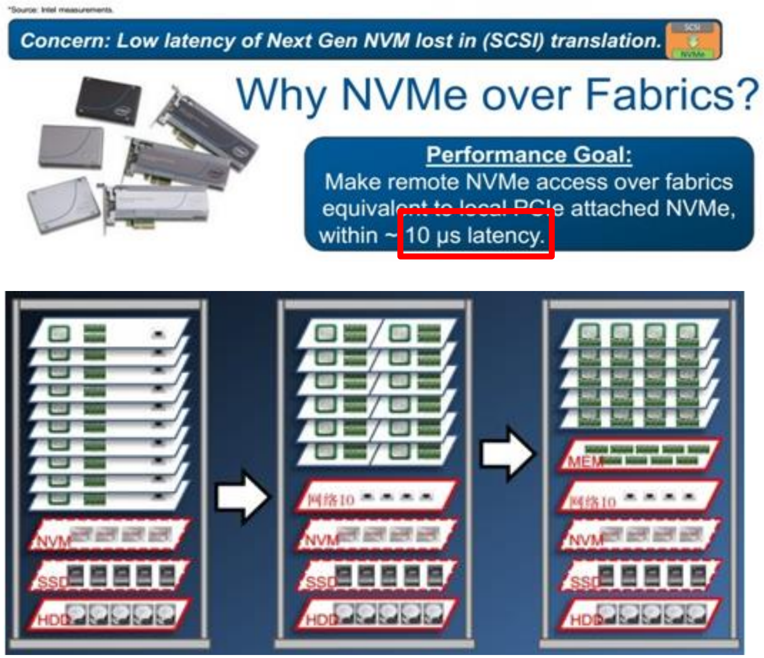
Deep Learning



Training Scale is Network Limited

Massively parallel HPC applications, such AI training, are dependent on low latency and high throughput network. Billions of parameters. Scales out is limited by network performance.

NVMe over Fabrics



*Source: Intel measurements.

Concern: Low latency of Next Gen NVM lost in (SCSI) translation.

Why NVMe over Fabrics?

Performance Goal:
Make remote NVMe access over fabrics equivalent to local PCIe attached NVMe, within ~ 10 μs latency.

SCSI
NVMe
MEM
网络10
NVM
SSD
HDD

Loss and Latency Sensitive

Disaggregated resource pooling, such as NVMe over Fabrics, use RDMA and run over converged network infrastructure. Low latency and loss are critical.

We are dealing with massive amounts of data and computing

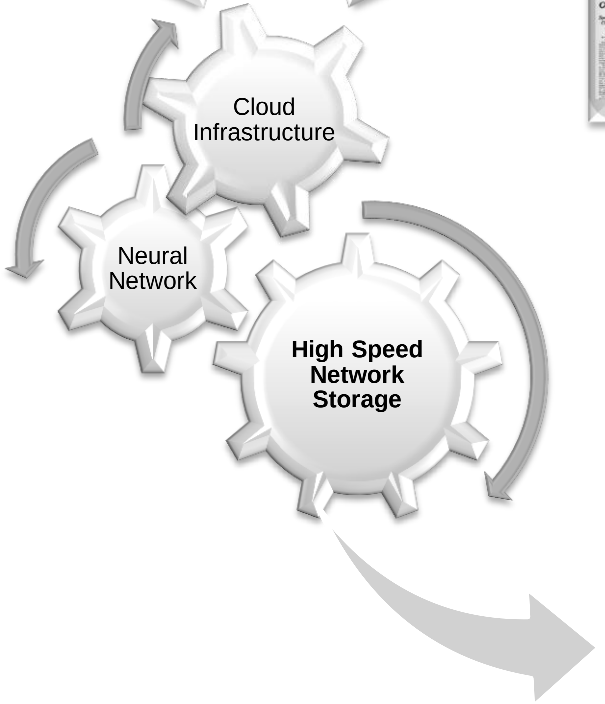
facebook
1B+ USERS
30+ PETABYTES

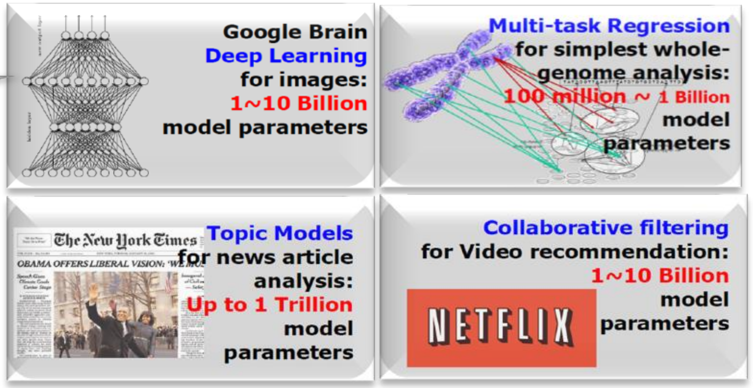
32 million pages
WIKIPEDIA
The Free Encyclopedia

YouTube
100+ hours video uploaded every minute

twitter
645 million users
500 million tweets / day

Divide and Conquer





Google Brain
Deep Learning
for images:
1~10 Billion
model parameters

Multi-task Regression
for simplest whole-genome analysis:
100 million ~ 1 Billion
model parameters

The New York Times
Topic Models
for news article analysis:
Up to 1 Trillion
model parameters

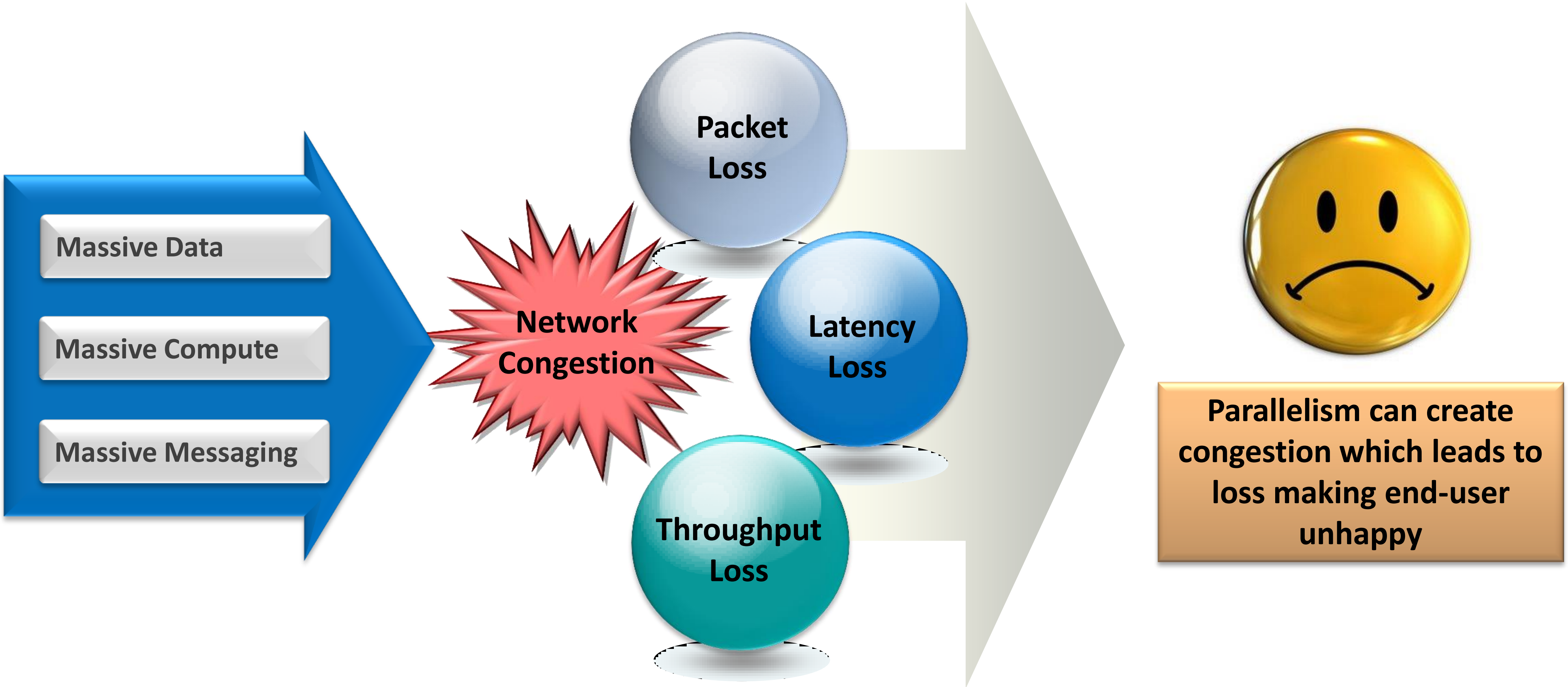
Collaborative filtering
for Video recommendation:
1~10 Billion
model parameters
NETFLIX

Requirements:

- **Fast-scalable storage**
- **Parallel applications and data**
- **Cloud-ified Infrastructure**

Real-time Natural Human/Machine Response

Congestion Creates the Problems



Massive Data

Massive Compute

Massive Messaging

Network Congestion

Packet Loss

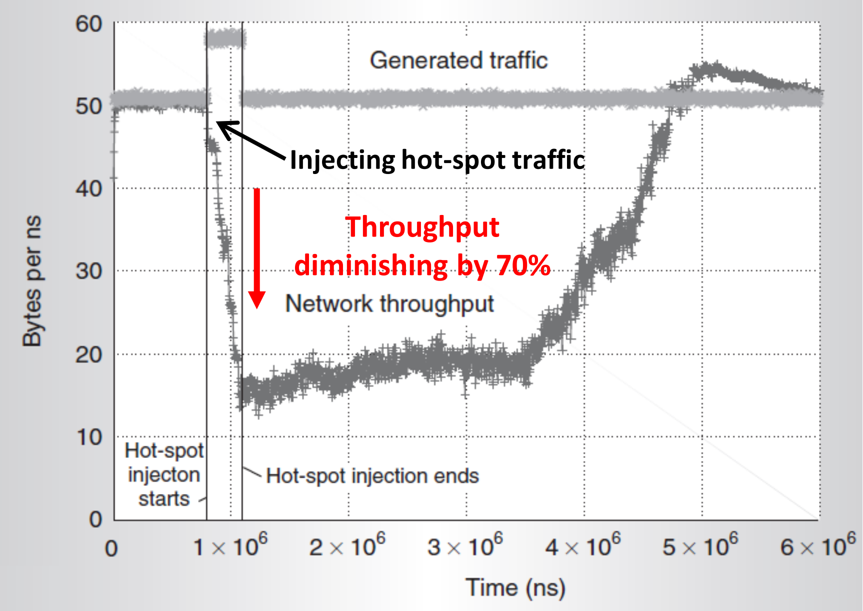
Latency Loss

Throughput Loss

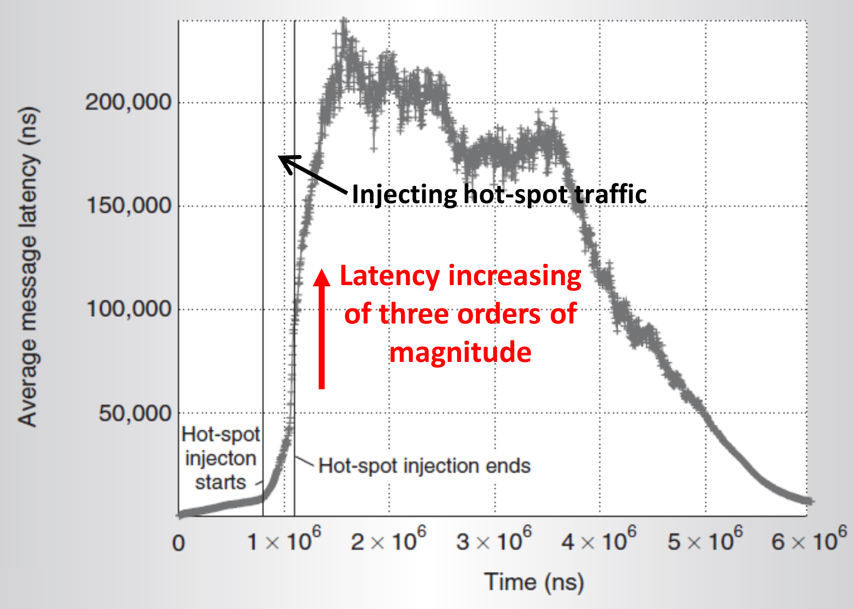
Parallelism can create congestion which leads to loss making end-user unhappy

The Impact of Congestion in Lossless Network

- The impact of congestion on network performance can be very serious.
- As shown in paper (Pedro J. Garcia et al, IEEE Micro 2006)[1]:



Network Throughput and Generated Traffic



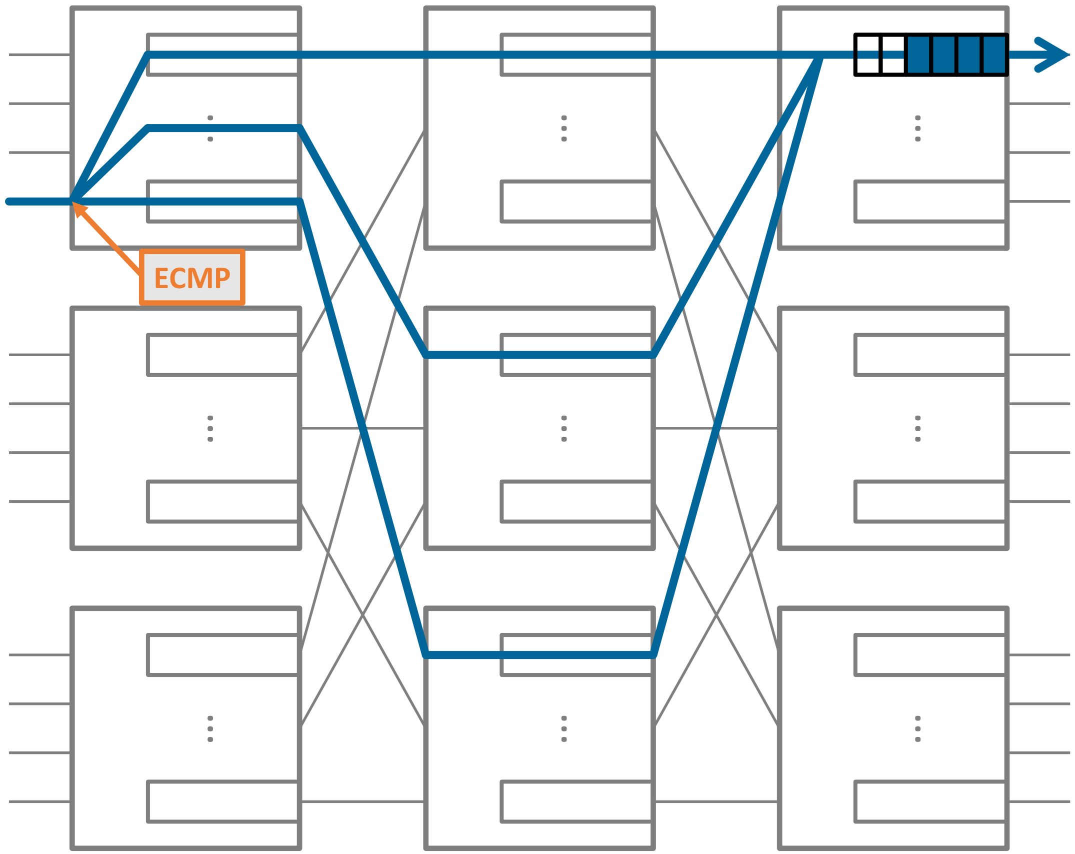
Average Packet Latency

Network Performance Degrades Dramatically after Congestion Appears

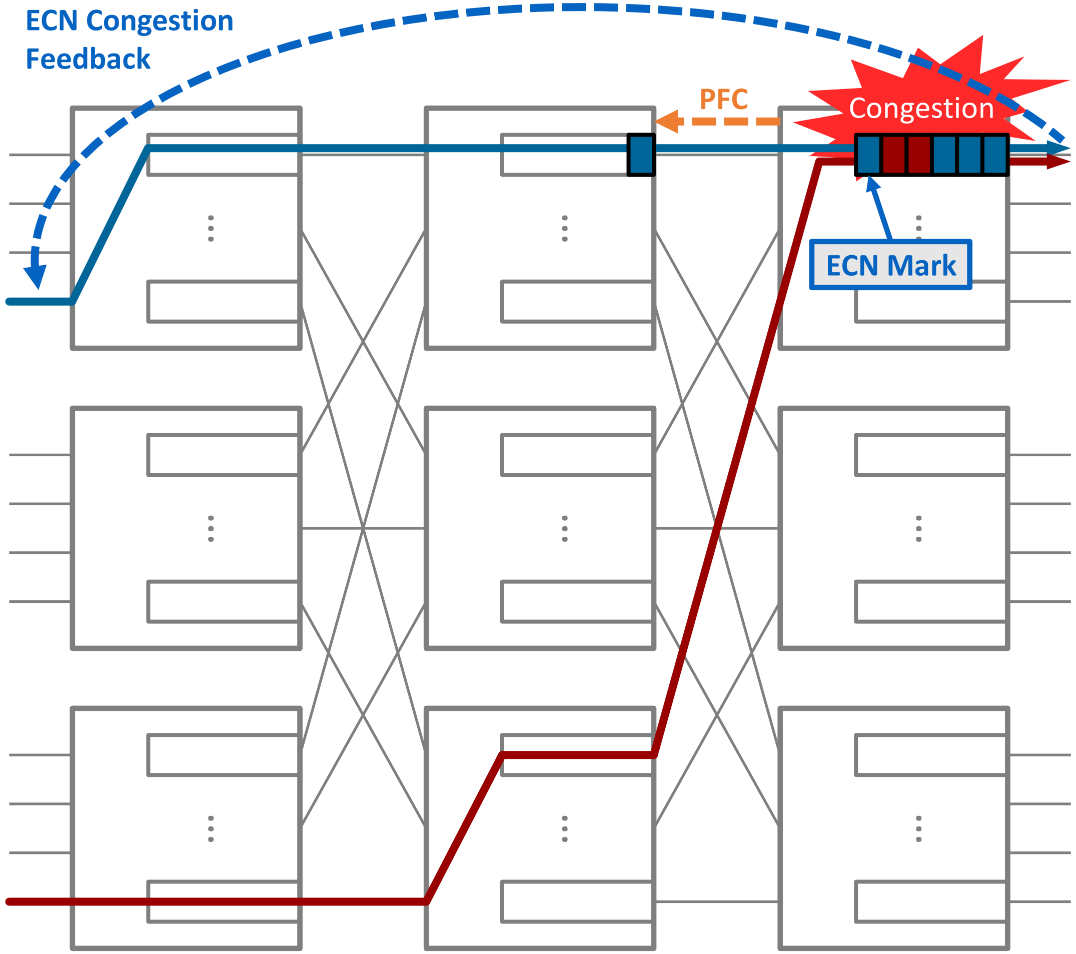
[1] Garcia, Pedro Javier, et al. "Efficient, scalable congestion management for interconnection networks." *IEEE Micro* 26.5 (2006): 52-66.

Dealing with Congestion today

ECMP – Equal Cost MultiPath Routing

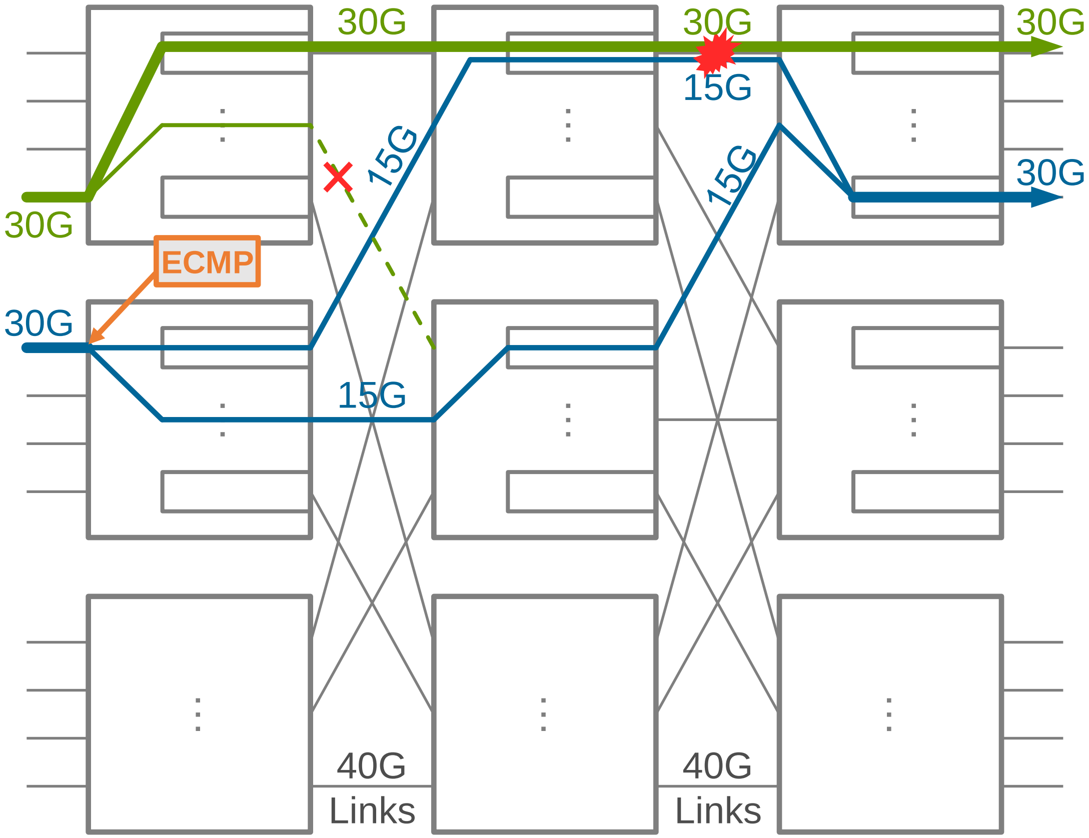


Explicit Congestion Notification (ECN) +
Priority-based Flow Control (PFC)

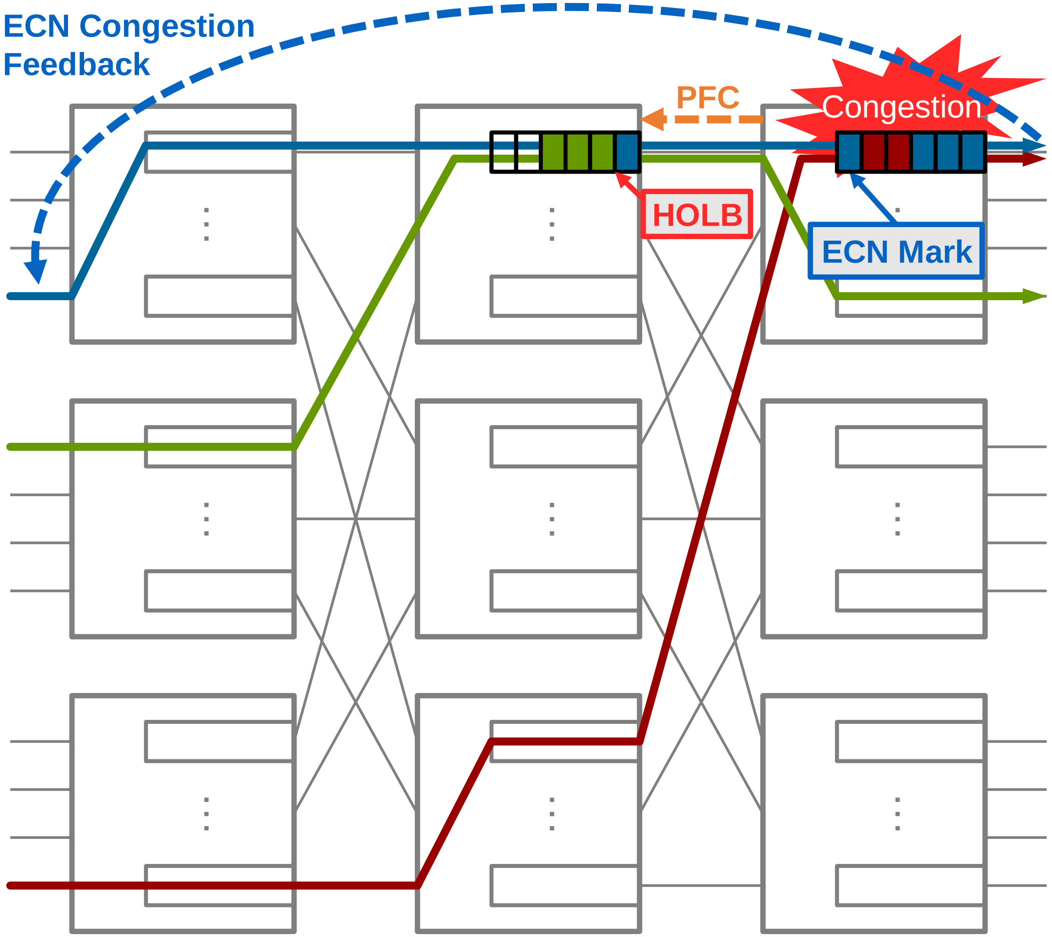


Ongoing challenges with congestion

ECMP Collisions



ECN Control Loop Delay
Head-of-line Blocking



Potential New Lossless Technologies for the Data Center

Goal = No Loss

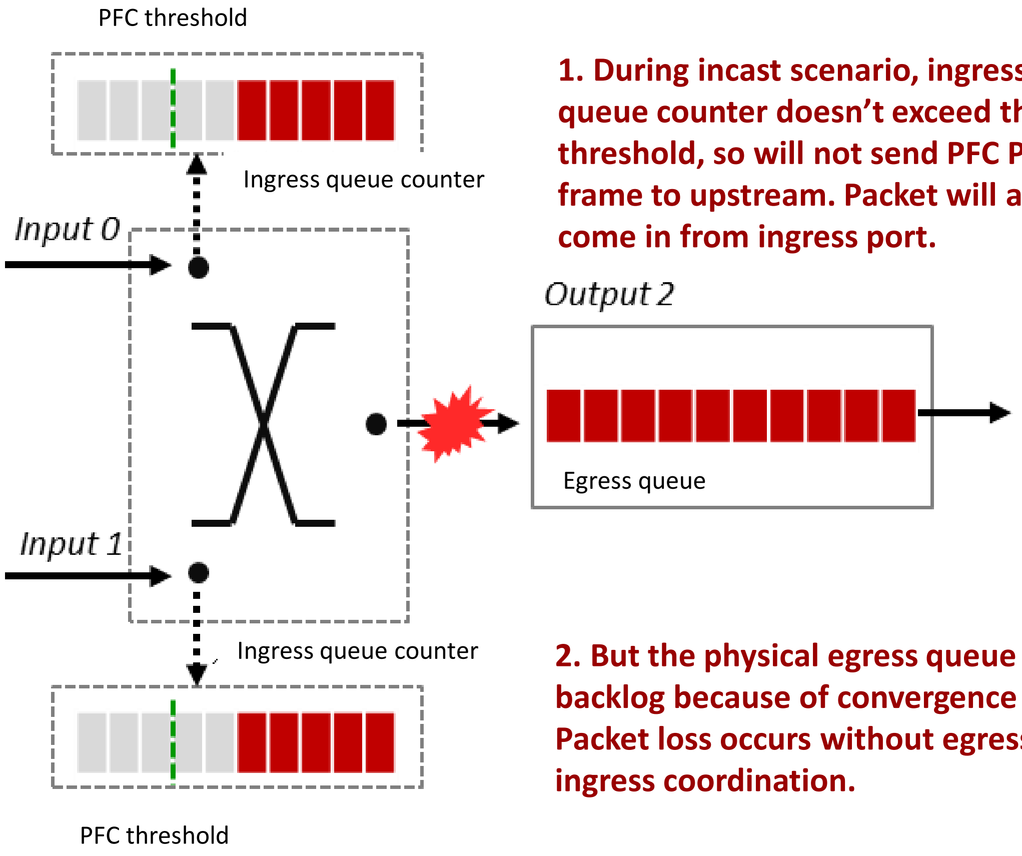
- No Packet Loss
- No Latency Loss
- No Throughput Loss

Solutions

- Virtual Input Queuing - VIQ
- Dynamic Virtual Lanes - DVL
- Load-Aware Packet Spraying - LPS
- Push & Pull Hybrid Scheduling - PPH

VIQ (Virtual Input Queues)： Resolve Internal Packet Loss

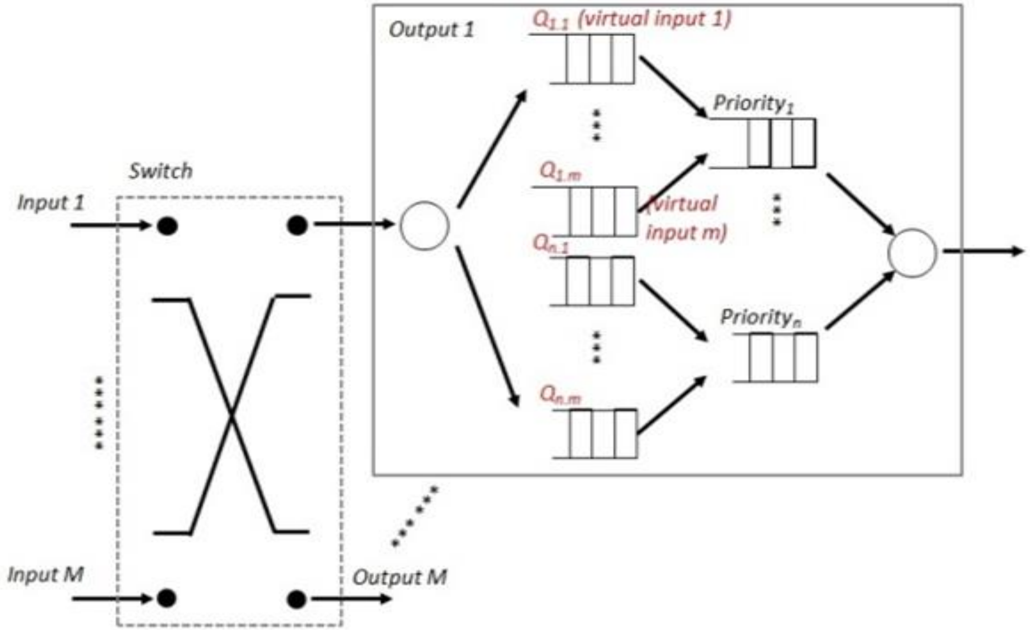
Incast Congestion leading to internal packet loss



1. During incast scenario, ingress queue counter doesn't exceed the PFC threshold, so will not send PFC Pause frame to upstream. Packet will always come in from ingress port.

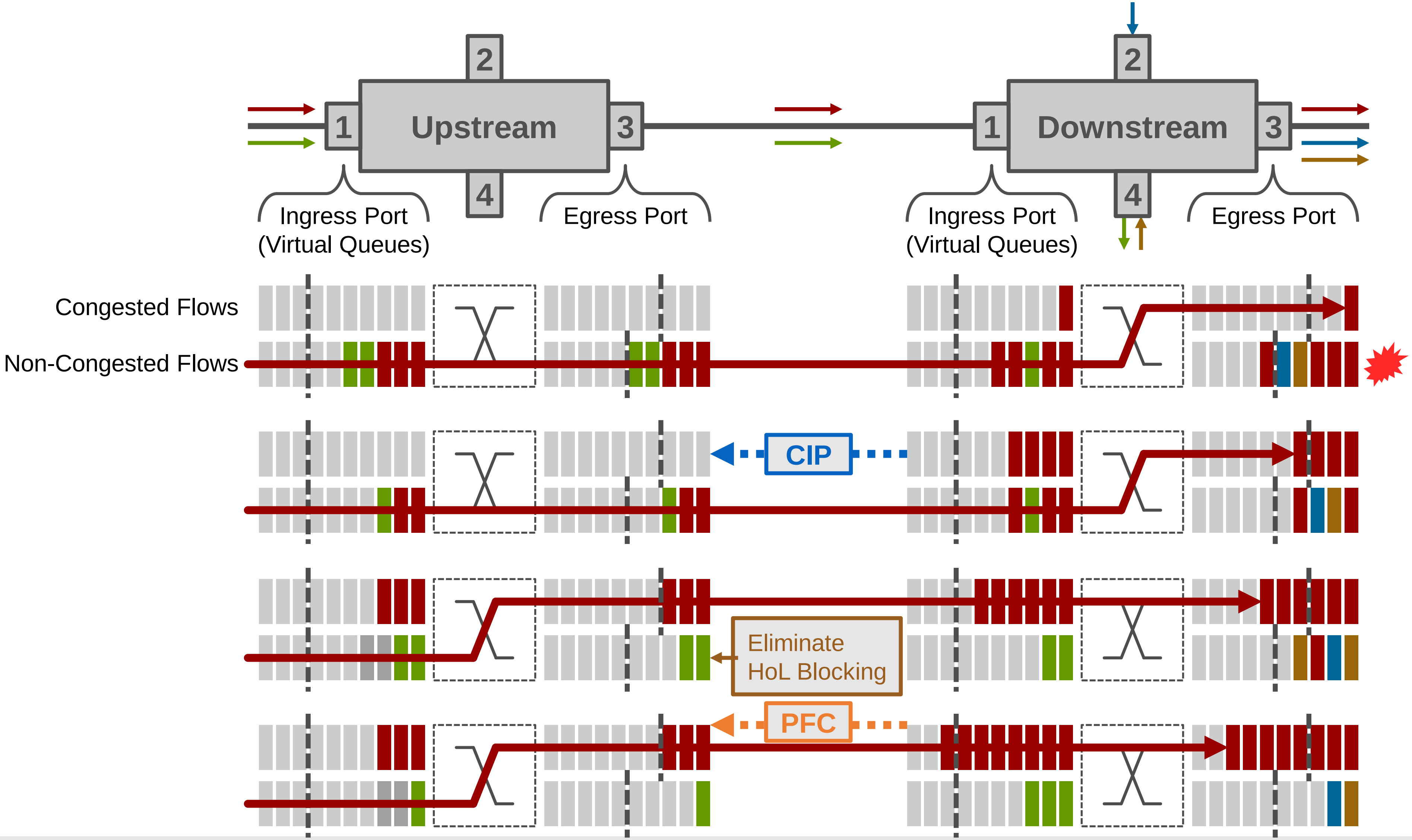
2. But the physical egress queue has backlog because of convergence effect. Packet loss occurs without egress-ingress coordination.

Coordinated egress-ingress queuing



VIQ could be looked as: that on out port, assign a dedicated queue for every in port. Memory changes from sharing to virtually monopolized according to in ports. So that every in port could get fair scheduling. The tail latency of business could be controlled effectively.

DVL (Dynamic Virtual Lanes)

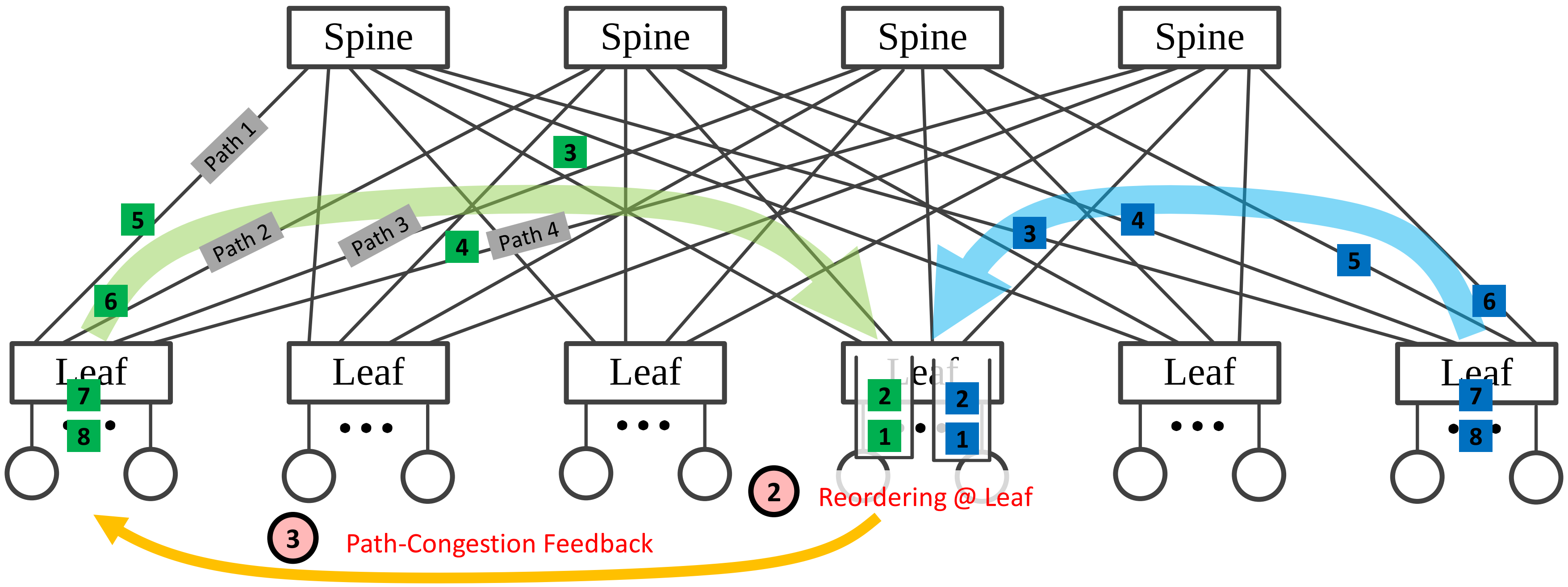


1. Identify the flow causing congestion and isolate locally
2. Signal to neighbor when congested queue fills
3. Upstream isolates the flow too, eliminating head-of-line blocking
4. If congested queue continues to fill, invoke PFC for lossless

LPS (Load-Aware Packet Spraying)

LPS = Packet Spraying + Endpoint Reordering + Load-Aware

Distributed | Finer Granularity | In-Ordering | Congestion-Aware



1 According to path-congestion degree, spray packets over paths

2 Reordering @ Leaf

3 Path-Congestion Feedback

PPH (Push & Pull Hybrid Scheduling)

PPH = Congestion aware edge switch scheduling

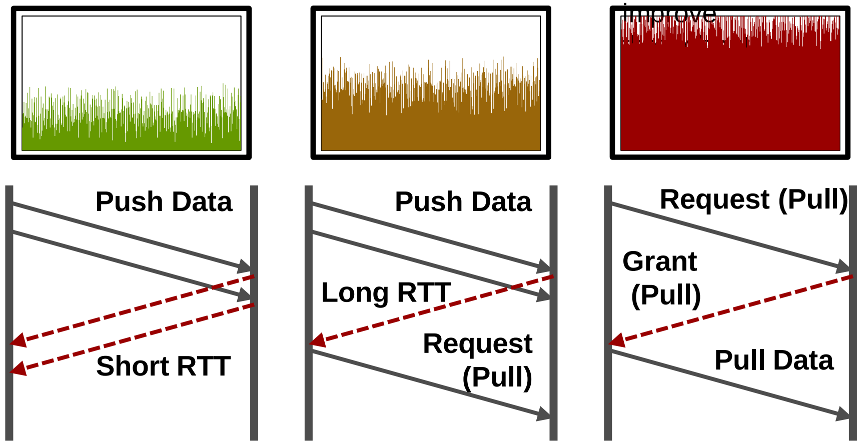
Push when load is light

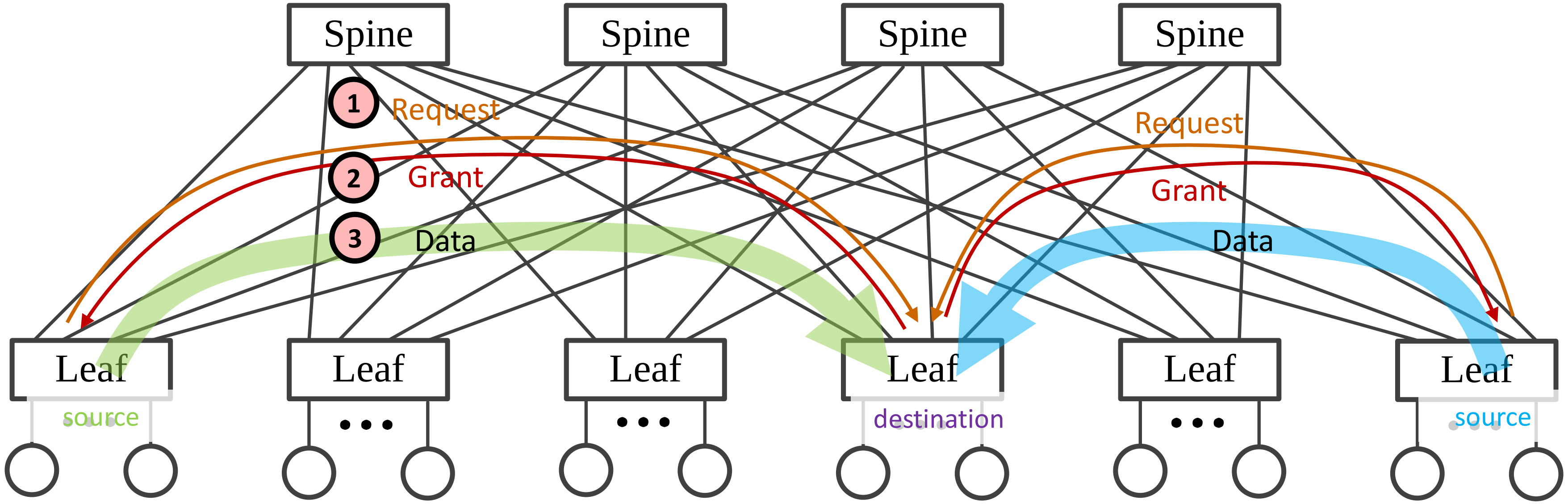
Pull when load is high

Light load: All Push. Acquire low latency.

Light congestion: Open Pull for part of the congested path

Heavy load: All Pull. Reduce queuing delay, improve





Innovation for the Lossless Network

Coping with Congestion		Mitigating Congestion	Innovation
Ingress thresholds unrelated to egress buffer availability. Incast causes internal packet loss.	Coordinated Resources	Coordinate egress availability with ingress demand. Avoid internal switch packet loss	**Virtual Input Queues**
Priority-based Flow Control (Coarse grain). Victim flows hurt by the congested flows	Isolate Congestion	Allow time for end-to-end congestion control. Move congested flows out of the way. Eliminate head-of-line blocking.	**Dynamic Virtual Lane**
Unbalanced load sharing. Elephant flow collisions block mice flows.	Spread the Load	Load-balance flows at higher granularity. Use congestion awareness to avoid collisions	**Load-aware Packet Spraying**
Source Network Destination Unscheduled and network resource unaware many-to-one communication leads to incast packet loss	Schedule Appropriately	Source Network Destination Scheduling decision integrated the information from source, network and destination.	**Push & Pull Hybrid Scheduling**

Thank You