SNIA NSF | NETWORKING STORAGE

Intro to Incast, Head of Line Blocking, and Congestion Management

Live Webcast
June 18, 2019
10:00 am PT

Today's Presenters

Tim Lustig
Mellanox

J Metz
SNIA Board of Directors
Cisco

Sathish Gnanasekaran
Brocade/Broadcom

John Kim
SNIA NSF Chair
Mellanox

SNIA-At-A-Glance

SNIA Legal Notice

SNIA | NSF | NETWORKING STORAGE

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Agenda

- Why This Presentation?
- Ethernet
- Fibre Channel
- InfiniBand
- Q&A

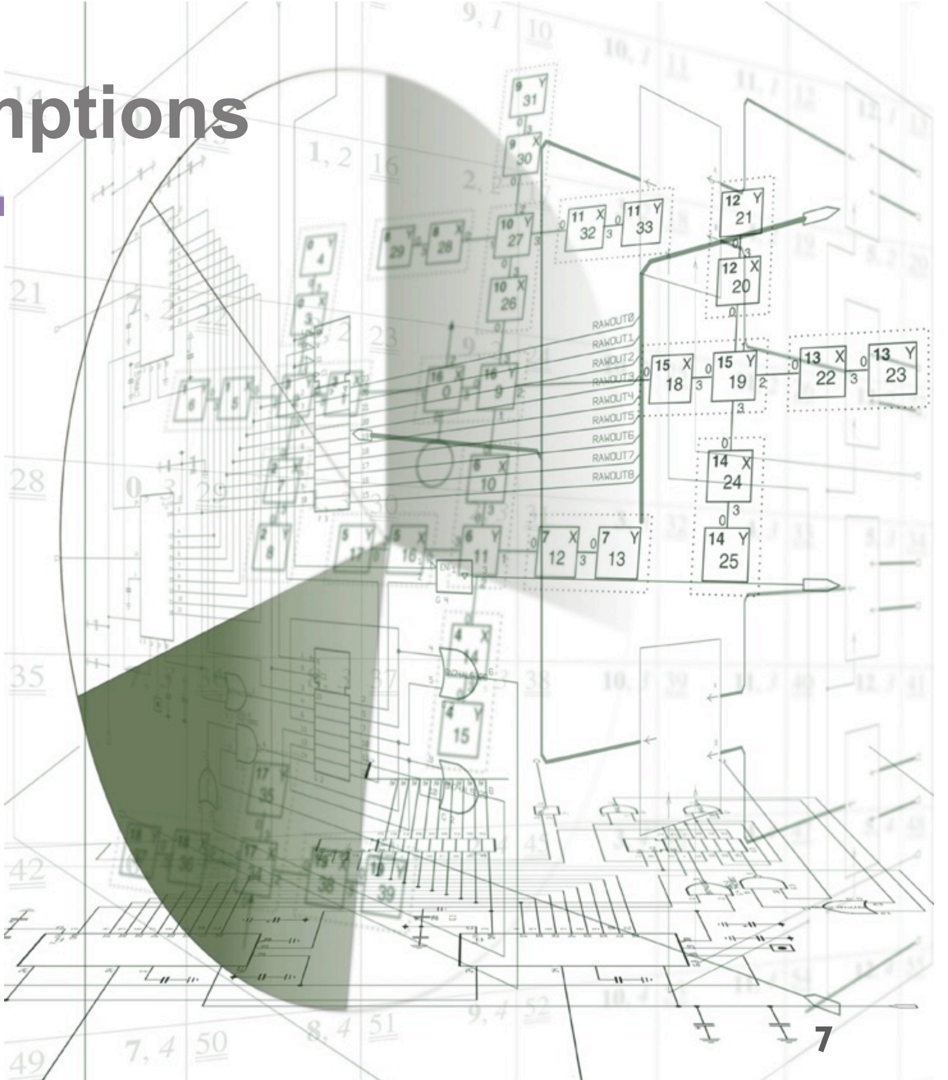
Why This Presentation?

- *All* networks are susceptible to congestion
- Advances in storage technology are placing unusual burdens on the network
- Higher speeds increase the likeliness of congestion
- Planning becomes more important than ever

*Gosling, James. 1997. The 8 fallacies of distributed computing.

Fixing The Wrong Assumptions

- Things that people get wrong about the ***network****
  - The network is reliable
  - Latency is zero
  - Bandwidth is infinite
  - The network is secure
  - Topology doesn't change
  - There is one administrator
  - Transport cost is zero
  - The network is homogeneous



*Gosling, James. 1997. The 8 fallacies of distributed computing.

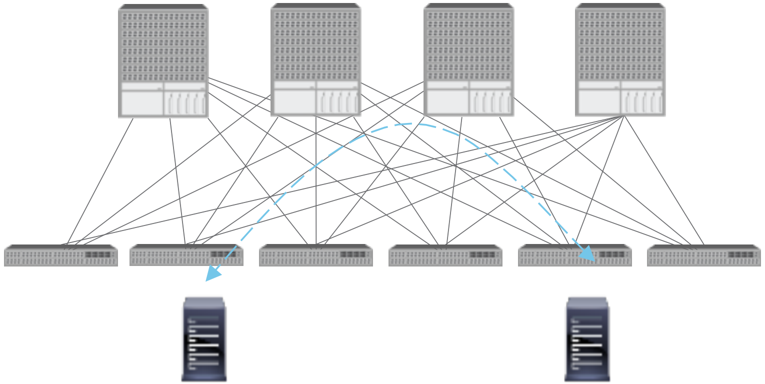
Ethernet

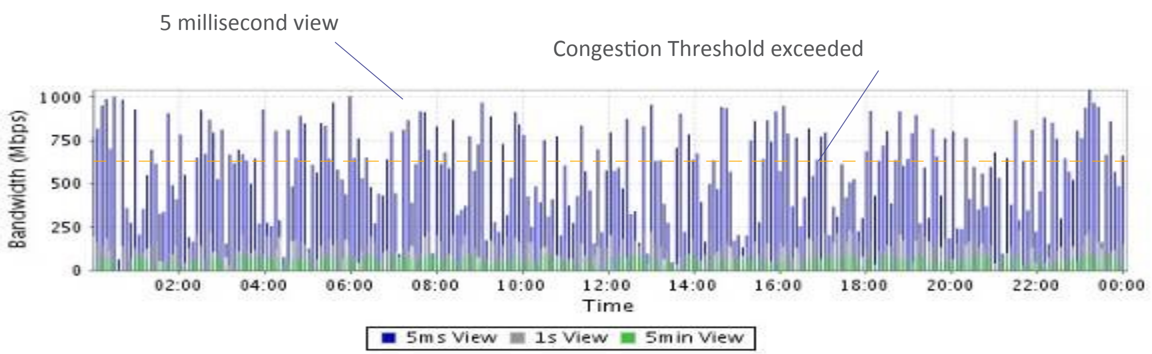
J Metz

Application-Optimized Networks

SNIA NSF | NETWORKING STORAGE

- You do not need to *and* should not be designing a network that requires a lot of buffering
- Capacity and over-subscription is not a function of the protocol (NVMe, NAS, FC, iSCSI, CEPH) but of the application I/O requirements



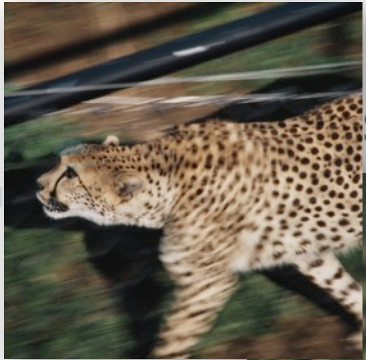


Data Centre Design Goal: Optimizing the balance of end to end fabric latency with the ability to absorb traffic peaks and prevent any associated traffic loss

Variability in Packet Flows

NETWORKING STORAGE

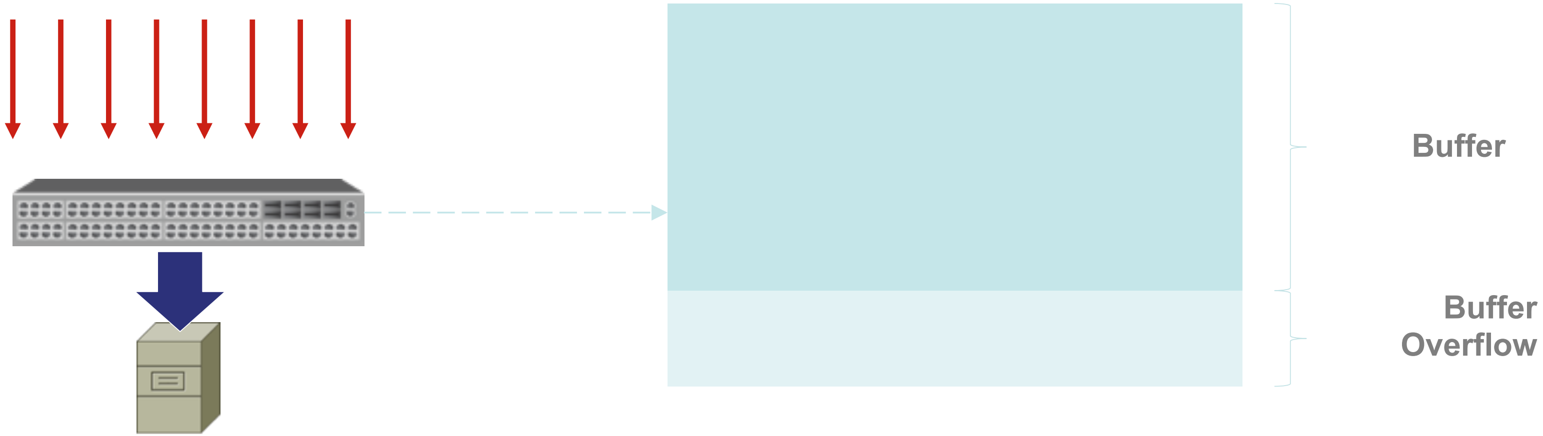
- Small Flows/Messaging
  - *(Heart-beats, Keep-alive, delay sensitive application messaging)*
- Small – Medium Incast
  - *(Hadoop Shuffle, Scatter-Gather, Distributed Storage)*
- Large Flows
  - (HDFS Insert, File Copy)
- Large Incast
  - *(Hadoop Replication, Distributed Storage)*




Understanding Incast

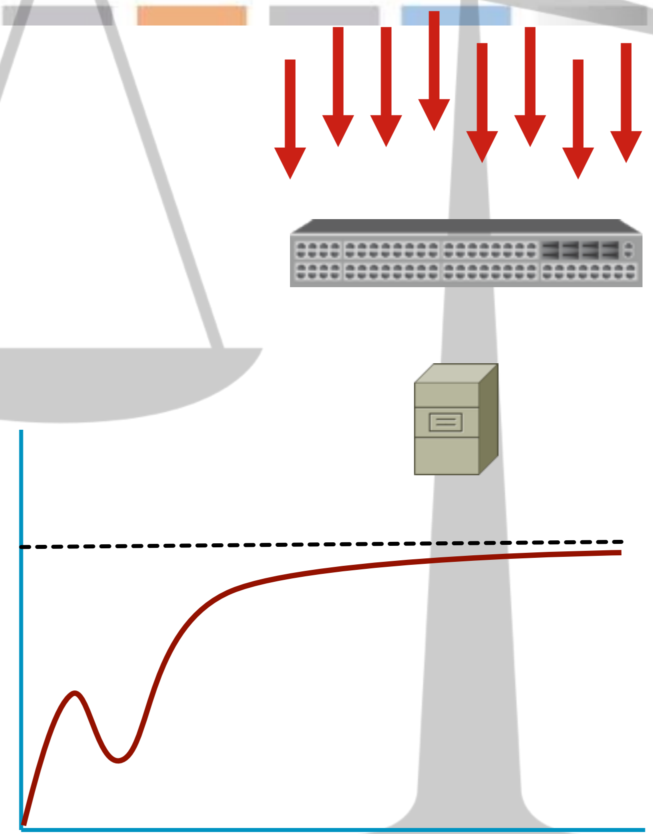
SNIA NSF | NETWORKING STORAGE

- Synchronized TCP sessions arriving at common congestion point (all sessions starting at the same time)
- Each TCP session will grow window until it detects indication of congestion (packet loss in normal TCP configuration)
- All TCP sessions back off at the same time



Incast Collapse

- **Incast collapse** is a very specialized case; not a permanent condition
- It would need every flow to arrive at exactly the same time
- The problem is more the buffer fills up because of elephant flows
  - Historically, buffers handle every flow the same
- Could potentially be solved with bigger buffers, particularly with short frames
- One solution is to have larger buffers in the switches than the TCP Incast (avoid overflow altogether), but this adds latency



Buffering the Data Center

- Large, “elephant flows” can overrun available buffers
- Methods of solving this problem:
  - Increase buffer sizes in the switches
  - Notify the sender to slow down before TCP packets get dropped

Option 1: Increase Buffers

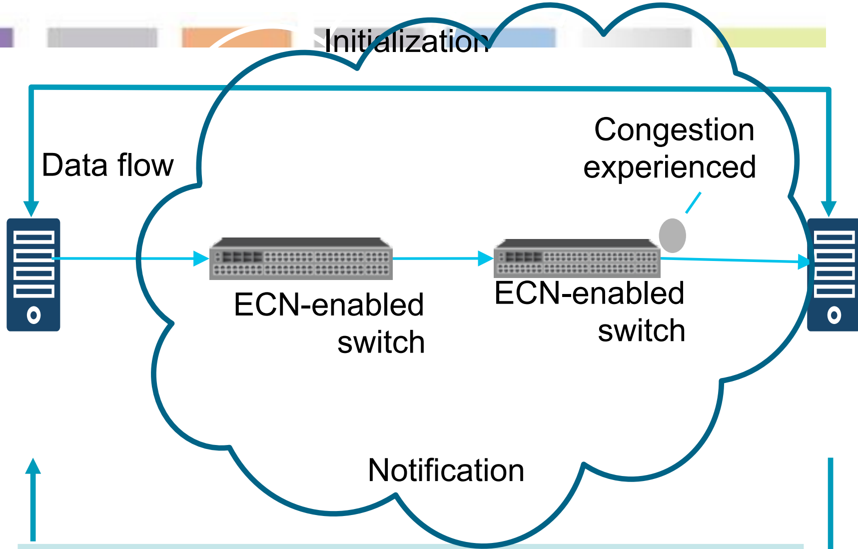
- Leads to “buffer bloat”
- Consequence:
  - As bandwidth requirements get larger, buffer sizes grow and grow
- What happens behind the scenes
  - Large TCP flows occupy most buffer
  - Feedback signals are sent when buffer occupancy is big
  - Large buffer occupancy can’t increase link speed but cause long latency
  - Healthy does of drops are necessary for TCP congestion control
    - Removing drops (or ECN marks) is like turning off the TCP congestion control)

Option 2: Telling the Sender to Slow Down

- Instead of waiting for TCP to drop packets and then adjust flow rate, why not simply tell the sender to slow down before the packets get dropped?
- Technologies such as Data Centre TCP (DCTCP) uses Explicit Congestion Notification, “ECN”) instruct the sender to do just this
- Dropped packets are the signal to TCP to modify the flow of packets being sent in a congested network

Explicit Congestion Notification (ECN)

- IP Explicit Congestion Notification (ECN) is used for congestion notification.
- ECN enables end-to-end congestion notification between two endpoints on a IP network
- In case of congestion, ECN gets transmitting device to reduce transmission rate until congestion clears, without pausing traffic.



DiffServ Field Values
0x00 – Non ECN Capable
0x10 – ECN Capable Transport (0)
0x01 – ECN Capable Transport (1)
0x11- Congestion Encountered

Data Center TCP (DCTCP)

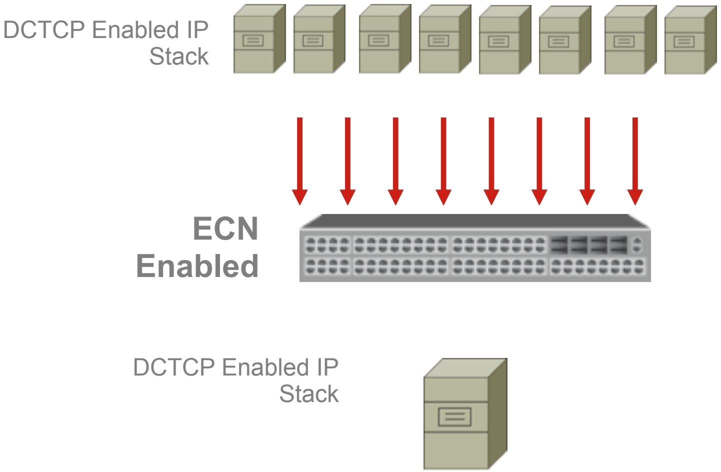
- Congestion indicated quantitatively (reduce load prior to packet loss)
- React in proportion to the extent of congestion, not its presence.
- Reduces variance in sending rates, lowering queuing requirements.

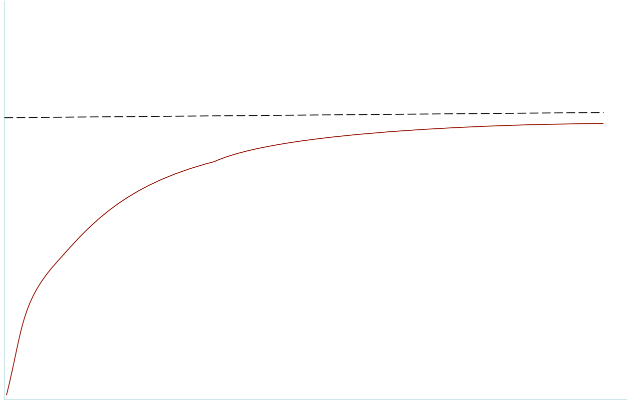
ECN Marks	TCP	DCTCP
1 0 1 1 1 1 0 1 1 1	Cut window by 50%	Cut window by 40%
0 0 0 0 0 0 0 0 0 1	Cut window by 50%	Cut window by 5%

- Mark based on instantaneous queue length.
- Fast feedback to better deal with bursts.

DCTCP and Incast Collapse

- DCTCP will prevent Incast Collapse for long lived flows
- Notification of congestion via ECN prior to packet loss
- Sender gets informed that congestion is happening and can slow down traffic
- Without ECN, the packet could have been dropped due to congestions and sender will notice this via TCP timeout





Fibre Channel

Sathish Gnanasekaran

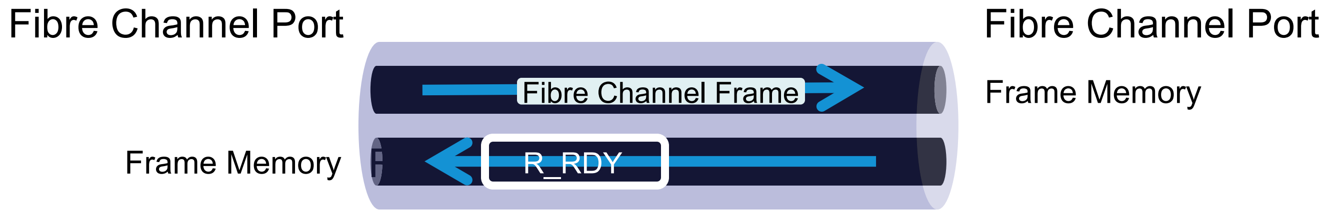
Congestion

- Offered traffic load greater than drain rate
  - Receive port does not have memory to receive more frames
- Non-lossless networks
  - Receiver drops packets, End-points retry
  - Retries cause significant performance impact
- Lossless Networks
  - Receiver paces transmitter, transmitter sends only when allowed
  - Seamlessly handles bursty traffic
  - Sustained congestion spreads to downstream ports causing significant impact

Fibre Channel

Credit Accounting

- When a link comes up, each side tells each other how much frame memory it has
  - Known as "buffer credits"
- Transmitters use "buffer credit" to track the available receiver resources
  - Frames are sent up to the number of available buffer credits
- Receivers tell transmitters when frame memory becomes available
  - Available buffer credit is signaled by the receiver ("receiver ready" – R_RDY)
  - Another frame can be sent



Fibre Channel

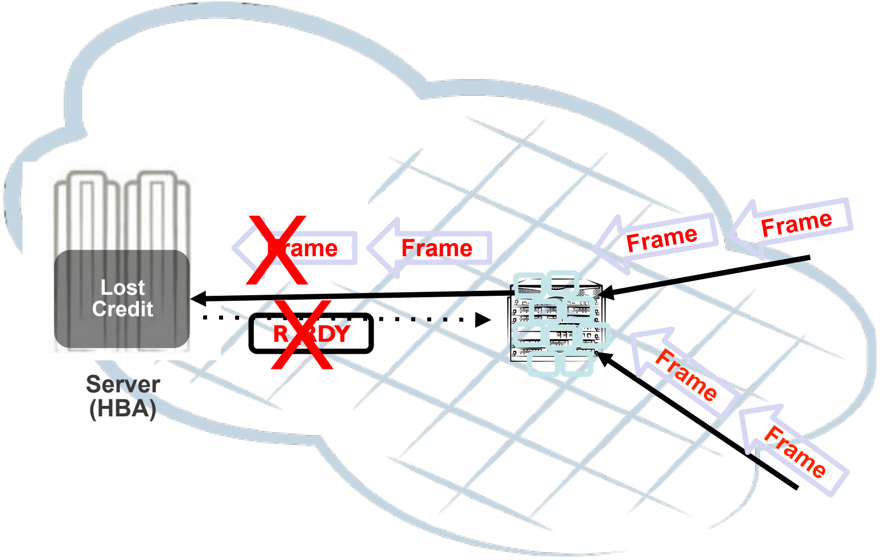
- Fibre Channel is a **credit based**, **lossless** network
  - Not immune to congestion

- Fibre Channel network congestion has **three causes**
  - **Lost Credit** occurs when the link experiences errors
  - **Credit Stall** occurs when frame processing slows or stops
  - **Oversubscription** occurs when the throughput demand exceeds link speed

Congestion Cause
Lost Credit

- **Lost Credit due to link errors**
  - Frames are not received
  - Credits are not returned
- **Credit tracking out of synch**
  - Transmitter sees fewer credits
  - Receiver sees fewer frames
- **Slows transmission rate**
  - Transmitter waits longer for credit on average
  - A **link reset** is required to recover

Lost Credit – Makes us slow down

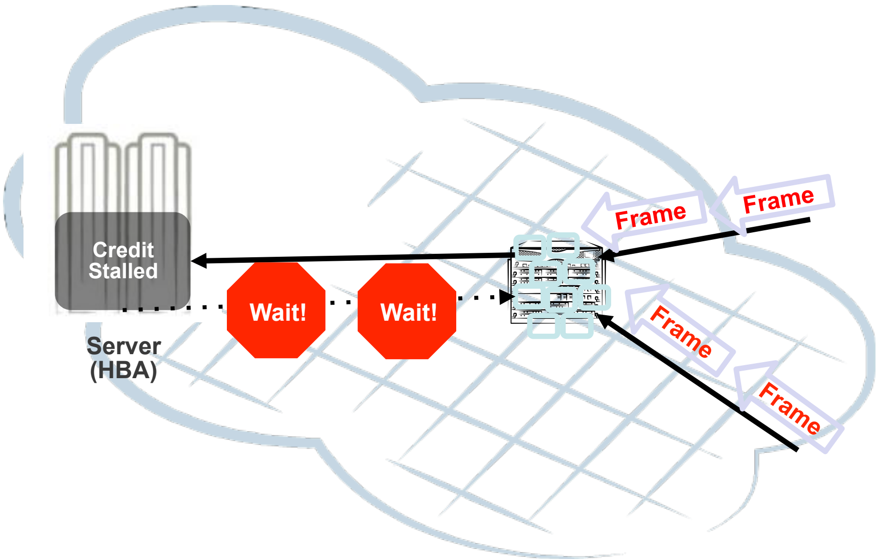


Lost credit causes congestion by reducing frame transmission rate!

Congestion Cause
Device Credit Stall

- **Credit Stall occurs due to rx device misbehavior**
  - Frames not processed
  - Buffers not freed, credits not returned
- Prevents **frame flow**
  - Frames for the device stack up in the fabric waiting for credit
  - Fabric resources are held by flow
- Frame flow slows for other devices
  - Frames for other devices can't move

Credit Stalled Device - Makes us wait



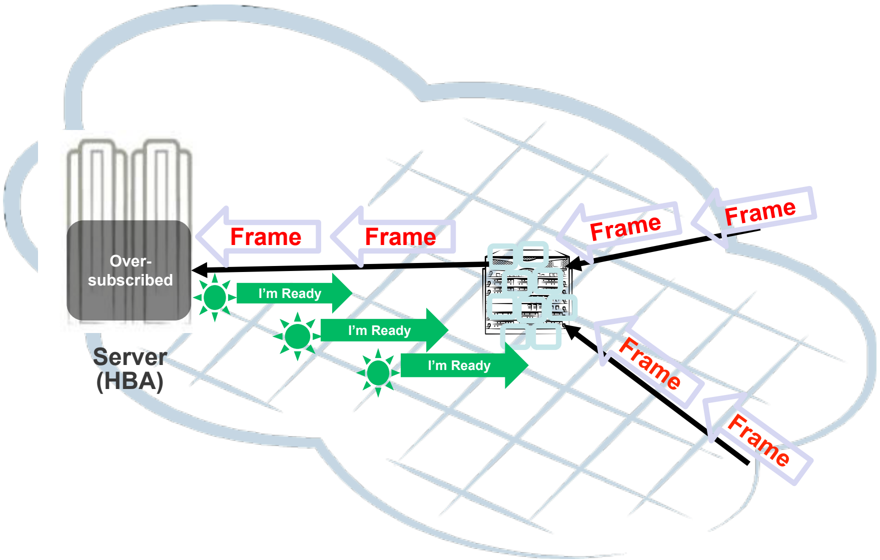
Credit Stalled Devices cause congestion by not sending "I'm Ready!"

Congestion Cause

Port Oversubscription

- **Oversubscription occurs due to resource mismatches**
  - Device asks for more data than link speed
- **Congestion slows upstream ports**
  - Throughput on upstream ports reduced by fan-in to congested port
  - Upstream port congestion worse
- **Slows frame flow**
  - Frame flow slowed due to lower drain rate
  - Affects congestion source as well as unrelated flows

Oversubscribed Device - Asks for too much



Oversubscribed Devices cause congestion by asking for more frames than the interface on the path can handle

Congestion Impact

- **Congestion** results in sub-optimal flow performance

- Sustained congestion radiates to upstream ports
  - Congestion spreads from receiver to all upstream ports
  - Not only affects the congestion source but unrelated flows
  - Can affect significant number of flows

- **Mild to moderate** congestion results in sluggish application performance

- **Severe** congestion results in application failure

Fibre Channel

Congestion Mitigation Solutions

- Detection
  - **Alerts** notify SAN administrators of link and device errors
- Credit Recovery
  - **Link reset** resets the credits on a link
  - **Credit recovery** automatically detects when a credit has been lost and restores it
- Isolation
  - **Port fencing** isolates mis-behaving links and/or devices
  - **Virtual channels** allows “victim” flows to bypass congested flows
  - **Virtual fabrics** allow SAN administrators to group and isolate applications flows efficiently

InfiniBand

John Kim

InfiniBand—The Basics

- InfiniBand is a **credit-based**, **lossless** network
  - **Lossless** because the transmitter cannot send unless the receiver has resources
  - **Credit-based** because credits are used to track those resources
  - Low latency with RDMA and hardware offloads

- InfiniBand network congestion has one main cause
  - **Oversubscription** occurs when the IO demand exceeds the available resources. Can be caused by incast
  - Can also come from hardware failure

InfiniBand—Congestion

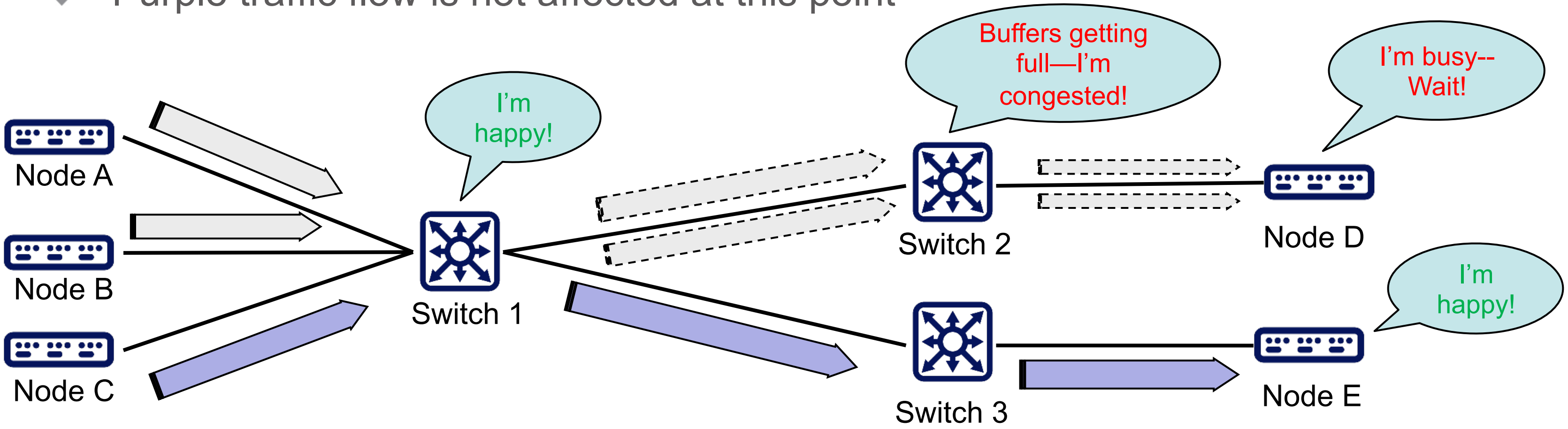
- If one destination is congested, sender waits
  - Since lossless, senders pause rather than drop packets
  - If pause too long, causes timeout problems
- If one switch congested too long, can spread
  - Flows to other destinations can be affected if they share a switch
  - Sometimes called “Head of Line Blocking”
  - Large “elephant” flows can victimize small “mice” flows
  - Not only in InfiniBand—true for other lossless networks

InfiniBand—Congestion

Initial congestion

Congestion Example

- Gray flows from Nodes A and B to Node D cause congestion at Switch 2.
  - Node D is overwhelmed and pauses traffic periodically
  - Switch 2's buffers fill up with incoming traffic
- Purple traffic flow is not affected at this point

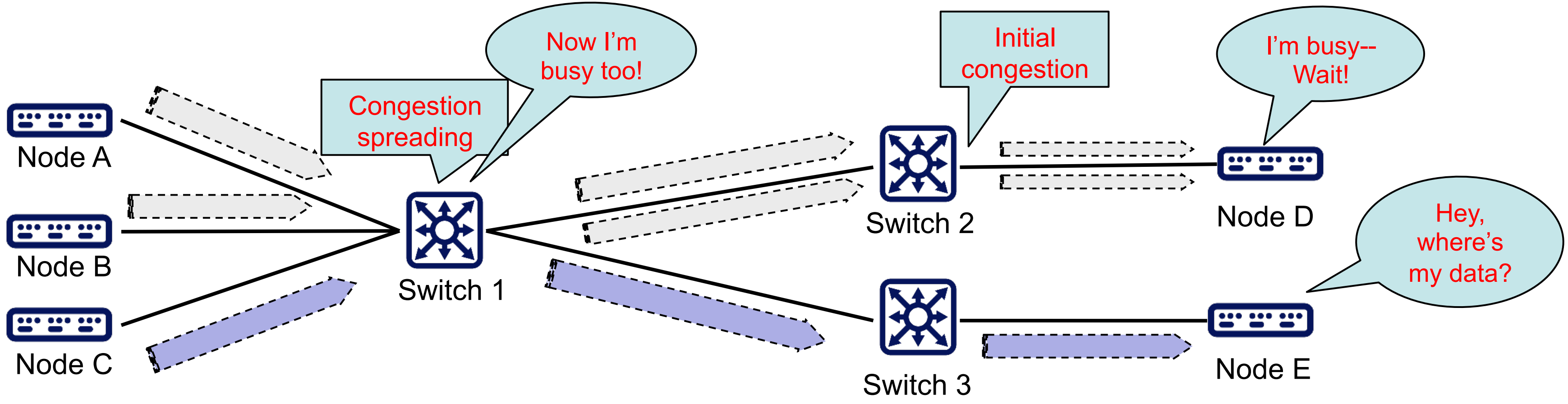


InfiniBand—Congestion

Initial
congestion

If Congestion Lasts Long Enough, It Can Spread

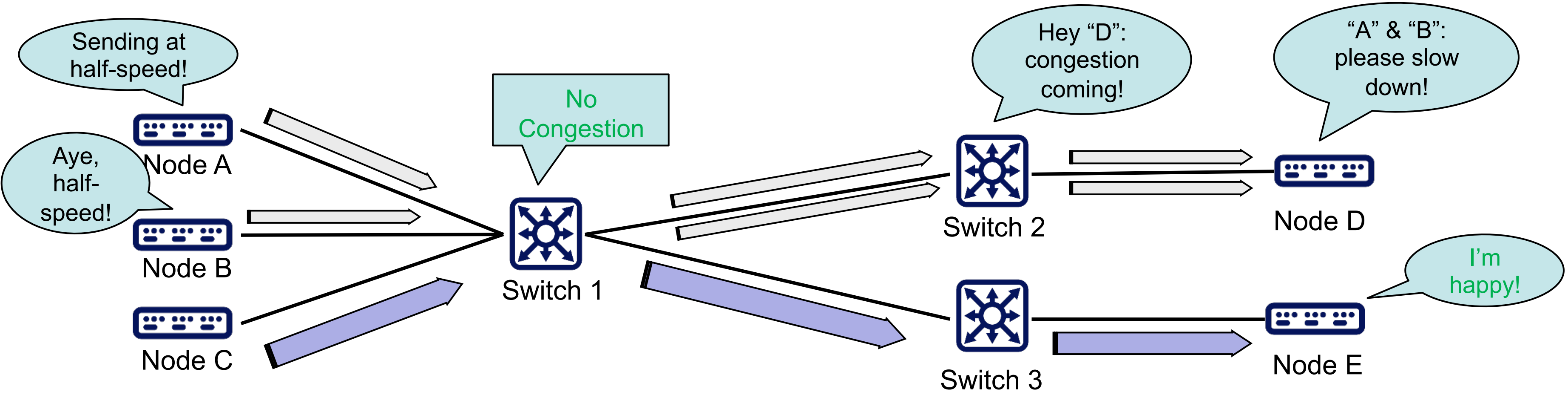
- Switch 2 asks Switch 1 to wait and Switch 1's buffers start to fill up
- Purple traffic flow is now affected—it's a "victim" flow
  - Even though it's between non-congested Nodes C and E



InfiniBand
Congestion Control Option

Initial congestion

- Congestion Control Throttles Traffic
  - Switch alerts destination about potential congestion
  - Destination alerts senders to slow down temporarily
  - Purple traffic flow is no longer victimized by gray flows



InfiniBand

Solving the Congestion Problem

- Overprovisioning
  - More bandwidth reduces chance of congestion
- Congestion Control
  - Similar to Ethernet ECN; hardware-accelerated notifications
- Adaptive Routing
  - Chooses least-congested path, if multiple paths available
- Virtual Lanes
  - Credit-based flow control per lane
  - Congestion in one traffic class does not affect other clases

Summary

- *Advances in storage are impacting networks*
  - *Network congestion differs by network type*
  - *But congestion can potentially affect all storage networks*
- *Issues and Symptoms*
  - *Incast collapse, Elephant flows, Mice flows*
  - *Lost credits, Credit Stall, Oversubscription*
  - *Hardware failures, Head of line blocking, Victim flows*
- *Cures*
  - *Data Centre TCP, Explicit Congestion Notification*
  - *Link reset, Credit recovery, Virtual channels, Port Fencing*
  - *Overprovisioning, Adaptive Routing, Virtual lanes*

After This Webcast

SNIA NSF | NETWORKING STORAGE

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Networking Storage Forum (NSF) website and available on-demand at www.snia.org/forums/nsf/knowledge/webcasts
- A Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-NSF blog: sniansfblog.org
- Follow us on Twitter @SNIANSF

Resources

- FCIA Webcast: Fibre Channel Performance: Congestion, Slow Drain, and Over-Utilization, Oh My!
  - https://www.brighttalk.com/webcast/14967/295141
- Buffers Queues and Caches Explained
  - http://sniaesfblog.org/buffers-queues-and-caches-explained/
- Explicit Congestion Notification
  - https://tools.ietf.org/html/rfc3168
- Performance Evaluation of Explicit Congestion Notification in IP Networks
  - https://tools.ietf.org/html/rfc2884
- Data Center TCP
  - https://tools.ietf.org/html/rfc8257
- Performance Study of DCTCP
  - https://people.csail.mit.edu/alizadeh/papers/dctcp-sigcomm10.pdf

Thank You