Source Priority Flow Control (SPFC)

towards Source Flow Control (SFC)

IETF 112, ICCRG

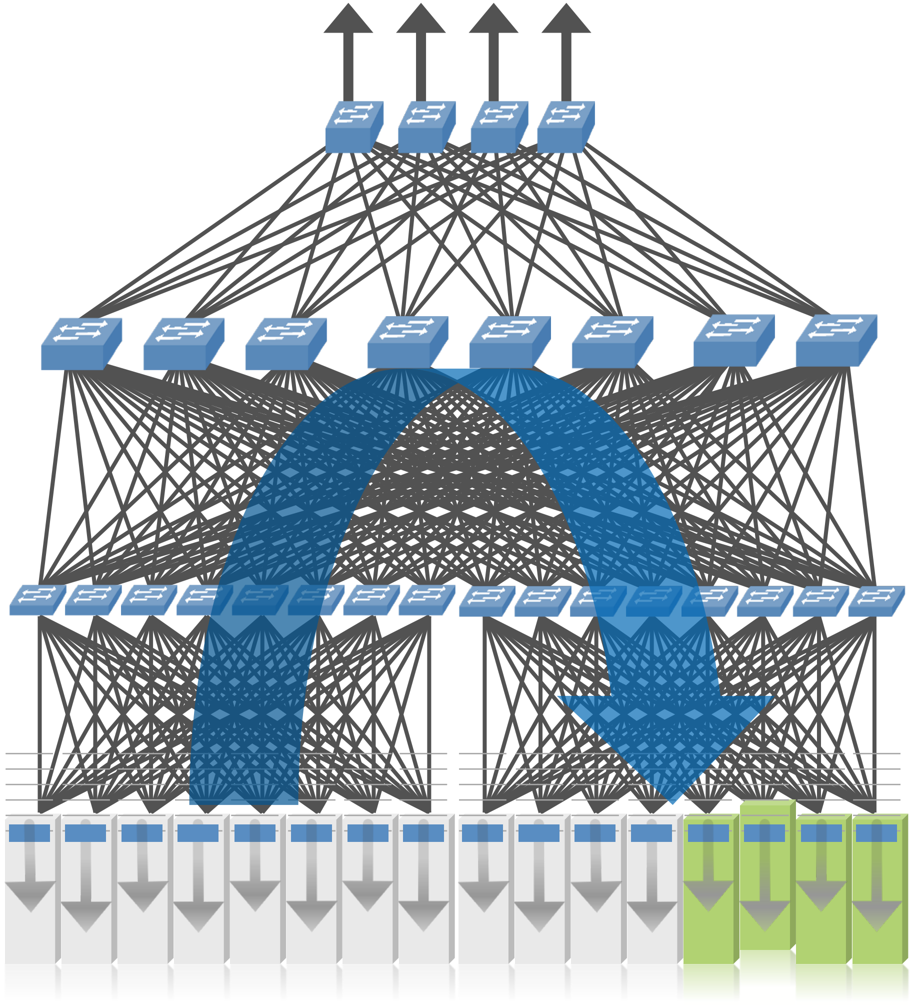
Contact: jk.lee@intel.com

Intel team: Jeongkeun Lee, Jeremias Blendin, Yanfang Le, Grzegorz Jereczek, Ashutosh Agrawal, Rong Pan

Collaboration at IEEE with Paul Congdon, Lily Lv (Huawei)

Incast Congestion in Data Centers

- Cause: many-to-one traffic pattern
- Mostly at the last 1 or 2 hop switches
- Governs max/tail latency
- Network tail latency can have a big performance impact on application workloads
- High incast of line-rate RDMA senders require very fast reaction at sub-RTT time, where RTT is congestion-free based RTT

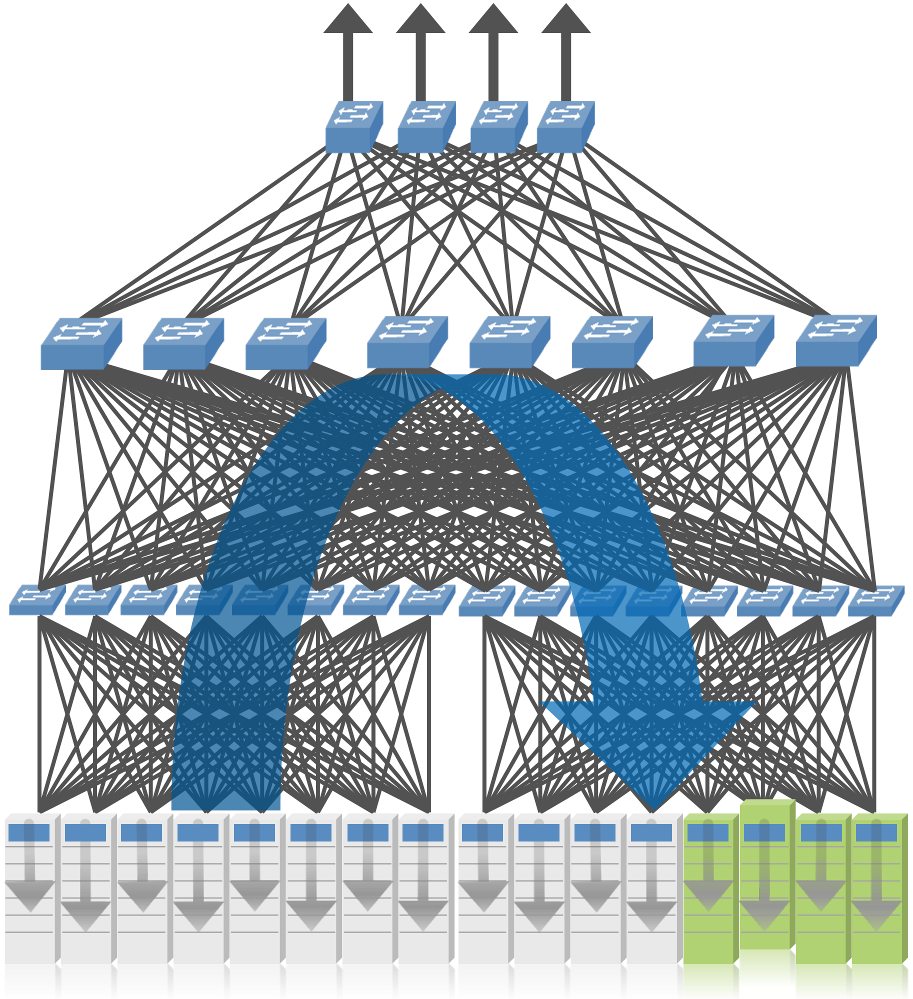


Solution space

- End-to-end (e2e) congestion control
  - Detect congestion in e2e path and adjust TX rates/cwnd
  - Congestion ‘signaling’ coupled w/ on-going congestion
  - Need many RTTs (100us to ms) to ‘flatten the curve’

    e.g., cut rate by half

    16:1 incast → 8:1 → 4:1 → 2:1 → 1:1 → … → 0
- Hop-by-hop L2 flow control, e.g., IEEE 802.1 Qbb PFC
  - Low-latency xon/xoff (less than 1us) to previous hop queue
  - Designed to prevent packet loss
  - Slows down the fabric at scale; operational side-effects
    - Head-of-line blocking (HoLB), backpressure
    - PFC storm, deadlocks



Need for a new, layer-3 flow control mechanism!

Proposed Approach to L3 Flow Control

At congested switch

- compute the minimal time needed to drain the incast queue, and
- L3 signal this info backwards towards the incast senders

Flow control reaction either by

1) Sender-side ToR switch converts it to standard PFC to sender NIC → ***“Source PFC (SPFC)”***
2) Sender NIC/host directly pauses the source flow → ***”Source Flow Control (SFC)”***

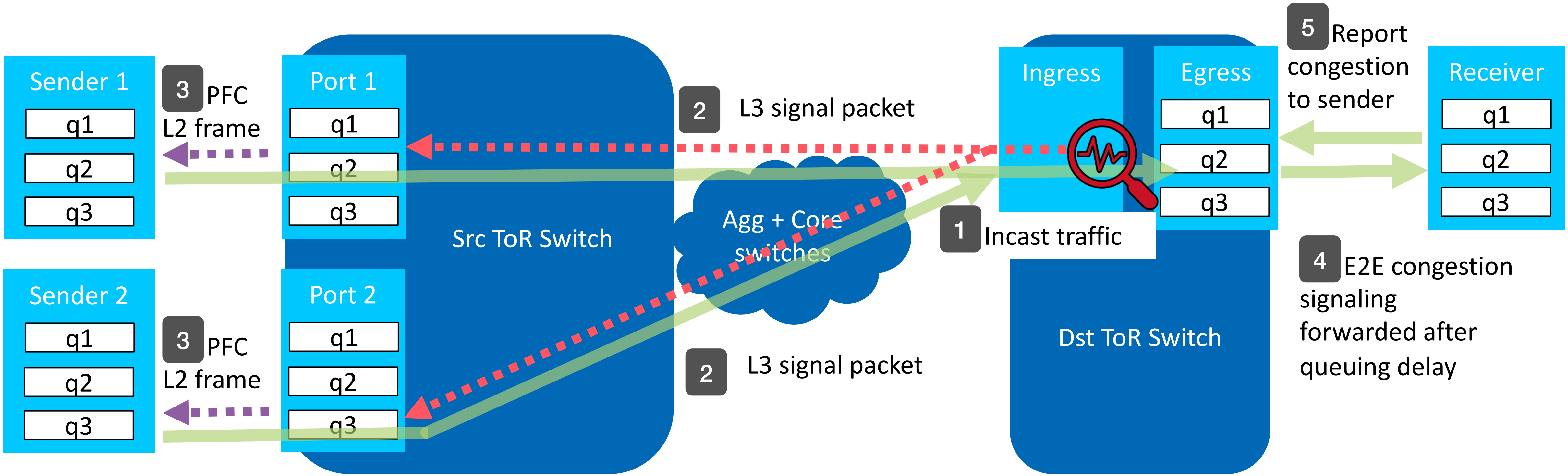
Source PFC: Edge-to-Edge View

What is SPFC?

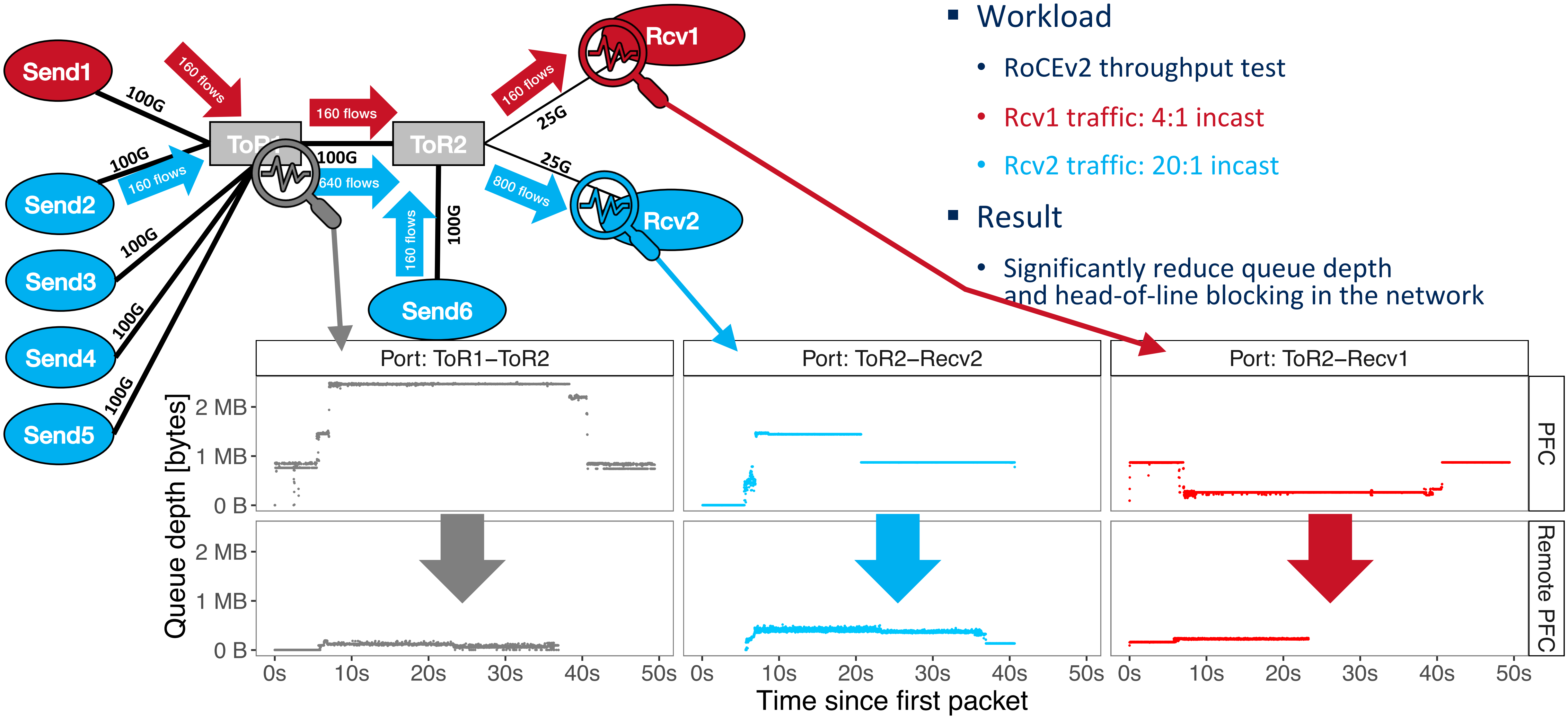
- Edge-to-Edge signaling of congestion
- Flow control that instantly ‘flattens the curve’
- Signaling + source flow ctrl all in sub-RTT
  - RTT = congestion-free base RTT

SPFC does not target/does target

- ~~aim 100% lossless~~ vs min switch buffering
- ~~e2e congestion ctrl~~ vs NIC flow ctrl
- ~~Pause Agg/Core switches~~ → no PFC side effects
- ~~Must greenfield deployment~~ → ToR-only upgrade



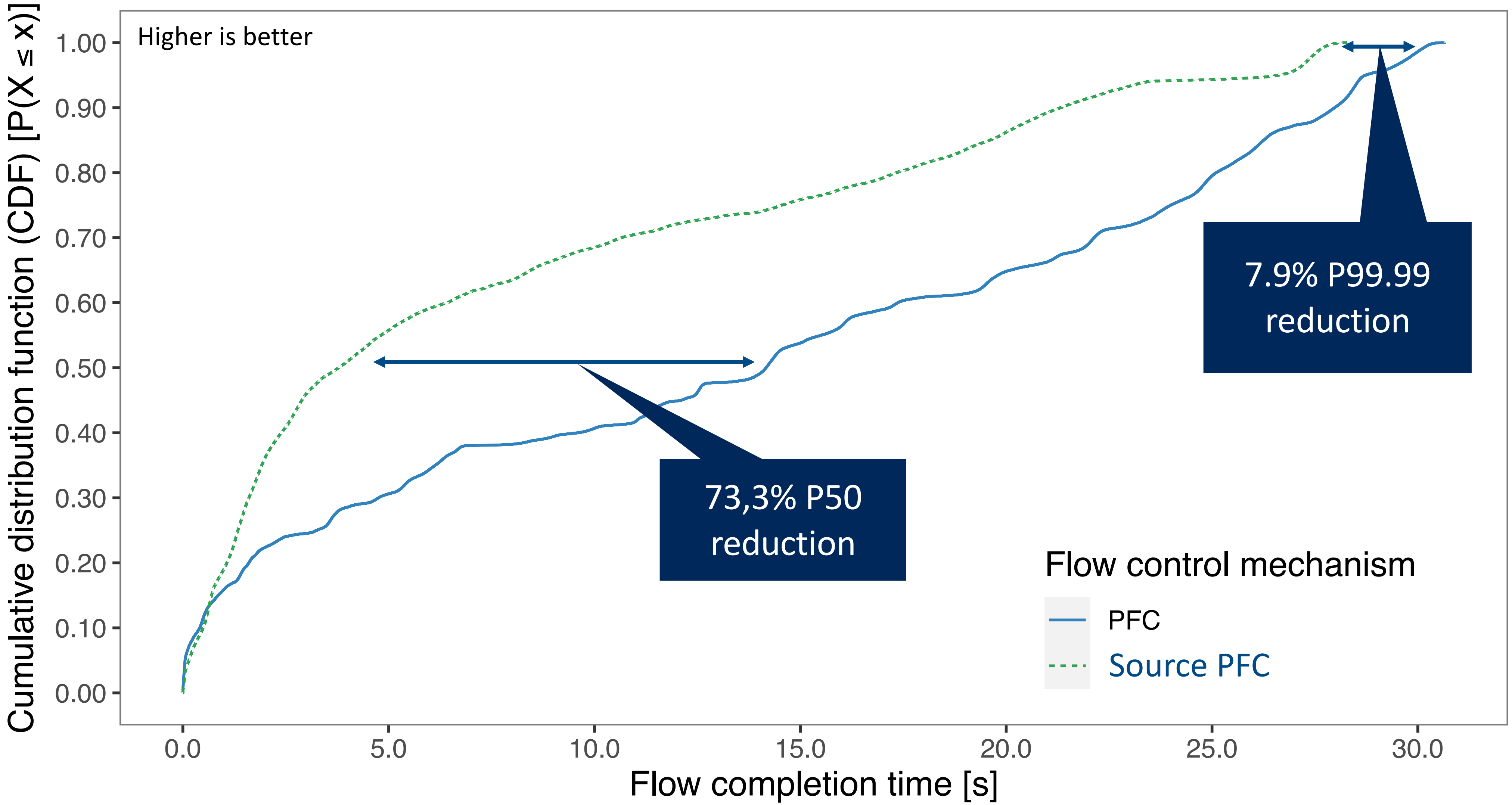
SPFC’s Effect on Queue Depth



- Workload
  - RoCEv2 throughput test
  - Rcv1 traffic: 4:1 incast
  - Rcv2 traffic: 20:1 incast
- Result
  - Significantly reduce queue depth and head-of-line blocking in the network

See backup for workloads and configurations. Results may vary.

SPFC’s Effect on Flow Completion Time



73,3% P50 reduction

7.9% P99.99 reduction

See backup for workloads and configurations. Results may vary.

Information to carry in the L3 signaling pkt
(see backup for IEEE 802.1 Qcz CIM hdr format)

- For SPFC mode
  - Source and destination IPs of the data pkt
    - SRC IP for reverse forwarding
    - Optional) DST IP for caching pause time per dst IP at sender ToR
    - simply swap src IP <-> dst IP from the data pkt into the signal packet; or need to ‘learn’ sender-ToR
  - DSCP, as needed to identify the PFC priority @ sender NIC
  - Pause time duration **<=** minimal drain time to reach the target queue level
  - Optional) congestion locator such as congested switch/port/queue IDs

- Additional info for true ‘source’ flow control (SFC)
  - More tuples of the data pkt, e.g., L4 ports, to identify the sender flow/connection
  - Note) L4 congestion control becoming part of NIC HW

‘Source’ Flow Control (SFC) = pause at flow level

- Either at SW stack or modified RDMA HW stack
  - E.g., On-Ramp @ NSDI’21 implemented at qdisc

- How does differ from ICMP Source Quench (SQ, deprecated RFC)?
  - SQ didn’t specify which info to signal or how to react
    - SFC carries pause time duration, and immediately pause the source flow
  - SQ was for WAN Internet
    - SFC is for data center with single administrative domain
- How does differ from IEEE QCN?
  - QCN is *Layer-2* congestion control btw switches and senders, needing multiple RTTs to ‘flatten the curve’
  - Note) RoCEv2 DCQCN is a L3 adoption of QCN, using ECN for e2e congestion control signal

Q/A

- How is the protocol secured? concerns of spoofing the control messages
  - For a single-domain data center of trusted switching devices
  - Signaling between switches (for SPFC) ~= LLDP or BGP
    - Note) BGP encryption may stop a man-in-the-middle attack; but doesn’t solve the problem of a malicious or poorly implemented router
  - SFC signaling to sender transport ~= ECN marking
  - ACL at domain boundaries can block signal pkts coming from NIC/host/outside
- Is there another use case for this besides RoCEv2?
  - RDMA is the primary use case of SPFC, making RDMA (regardless of transport) to scale on standard Ethernet fabric
    - See backup for the case with ML training
  - SFC can be applied to non-RDMA use cases; evaluation WiP
- Edge-to-Edge signaling delay will be proportional to RTT, solution for large DC?
  - Cache per-dstIP pause time at sender-ToR or NIC; instant flow control new senders towards the incast dst IP

History & Status

- Public presentations of the concept and data at P4 Workshops (Apr’20, May’21) and Open Fabrics Alliance (Mar’21)
  - https://opennetworking.org/wp-content/uploads/2020/04/JK-Lee-Slide-Deck.pdf (slide 12)
  - https://www.openfabrics.org/wp-content/uploads/2021-workshop-presentations/503_Lee_flatten.pdf
  - https://opennetworking.org/wp-content/uploads/2021/05/2021-P4-WS-JK-Lee-Slides.pdf (slide 14)
- Previous contributions “Source Flow Control (SFC)” have been presented in IEEE 802.1 Nendica
  - https://mentor.ieee.org/802.1/dcn/21/1-21-0055-00-ICne-source-flow-control.pdf - 9/16/2021
  - https://mentor.ieee.org/802.1/dcn/21/1-21-0061-00-ICne-source-remote-pfc-test.pdf – 10/14/2021
- Suggested that IETF should be aware of the activity as the proposed signaling is Layer-3
  - Announced in IEEE802-IETF Coordination Meeting - October 25, 2021
  - IETF 112, ICCRG session ← today
- Plan at IEEE
  - Consider a modified P802.1Qcz CIM Layer-3 message
  - Propose ‘Changes to 802.1Q and/or Qcz’ presentation in Nendica – Nov 2021 Plenary
  - Consider a motion to develop PAR & CSD at the March 2022 Plenary

Summary & next steps

- Source PFC allows sender-side switch to react to remote switch congestion
  - w/o fabric HoL blocking or backpressure
  - ‘remote switch’ = receiver-side switch, spine, … , gateway switch in inter-DC RDMA
  - Q: Inter-DC operation: secure tunneling?

- SFC can be consumed by sender transport for flow-level reaction
  - Q: transport agnostic signaling (in IEEE/IETF) or signaling within transport hdr?
- Need a decision at IEEE802.1-IETF coordination

Notices and Disclaimers

- Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
- Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.
- Your costs and results may vary.
- Intel technologies may require enabled hardware, software or service activation.
-

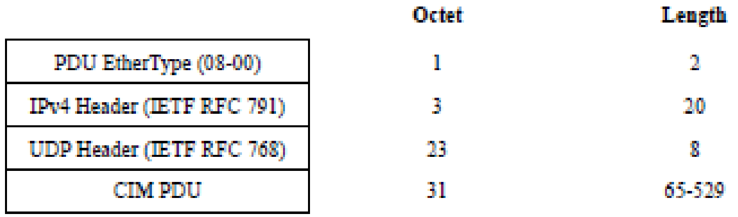
intel®

From Paul Congdon

Msg format: can leverage 802.1 Qcz Congestion Isolation Message (CIM)

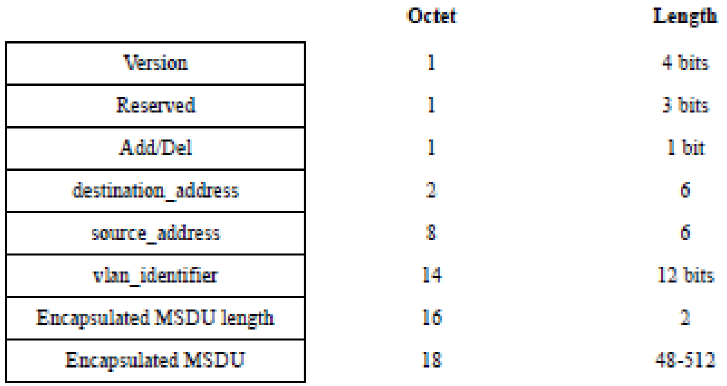
- Qcz CIM has Layer-2 and Layer-3 formats
- The CIM PDU contains enough of the payload to identify the offending flow
- Carrying the needed information:
  - Src / Dest IP addresses
  - DSCP
  - Additional tuples of the data pkt
- What’s missing?
  - Pause time
  - Simplified format of above information (i.e not MSDU)
  - Selection of CIM Destination IP (NOT previous hop)

Table 47-2—IPv4 layer-3 CIM Encapsulation



	Octet	Length
PDU EtherType (08-00)	1	2
IPv4 Header (IETF RFC 791)	3	20
UDP Header (IETF RFC 768)	23	8
CIM PDU	31	65-529

Table 47-4—CIM PDU



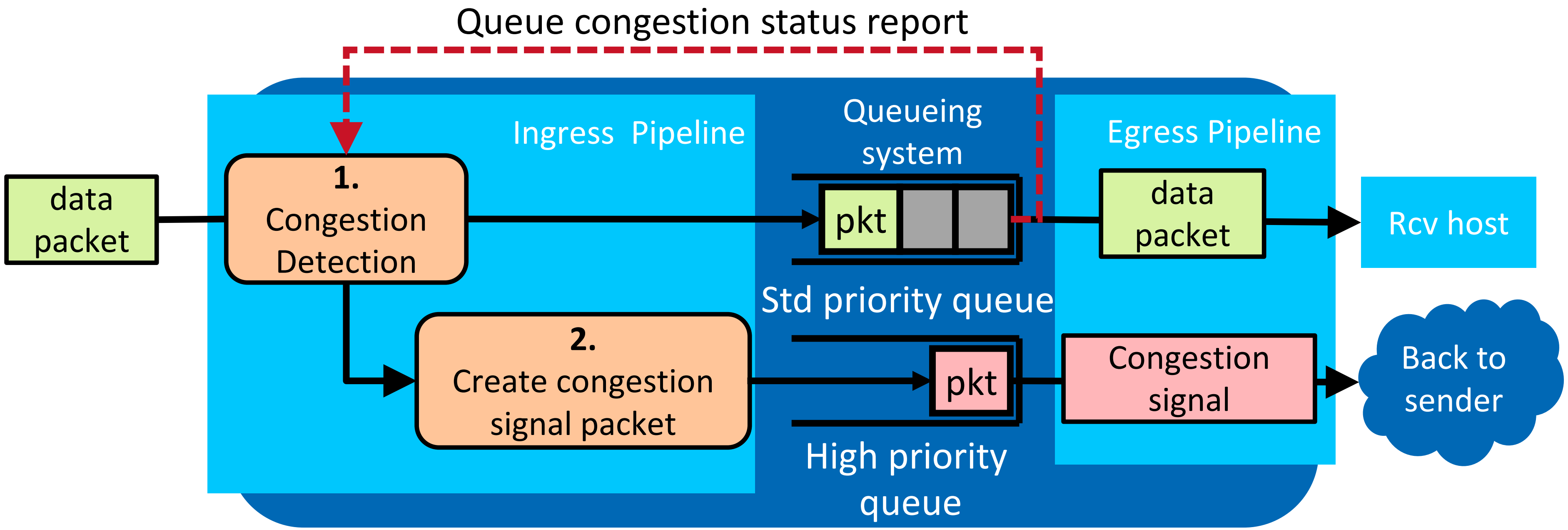
	Octet	Length
Version	1	4 bits
Reserved	1	3 bits
Add/Del	1	1 bit
destination_address	2	6
source_address	8	6
vlan_identifier	14	12 bits
Encapsulated MSDU length	16	2
Encapsulated MSDU	18	48-512

SPFC is a good fit for ML training workload

- ML training: mostly elephant RDMA flows
- Data parallel: allreduce collective or its variants → source of incast
  - Many-to-one pattern: at a given time, a NIC sends only to one receiver
  - Reduced or no concern of HoL blockings at NIC queue
- Model parallel: all-to-all collective
  - Note) trend is converting all-to-all to a set of allreduce
- Recommend: isolate all-to-all and allreduce traffic by DSCP
  - Isolate two in different NIC PFC queues
  - Switch: SPFC signal only to allreduce senders
- Cluster experiment WiP

Intelligent Congestion Detection

1. The programmable logic checks the congestion status of an outgoing queue before enqueuing a packet
2. If congestion is detected, a signaling packet is created that skips the congestion and is sent directly back to the sender
    1. Redundant signaling back to the same sender/flow is suppressed temporally



Simulation setup

Custer: 3-tier, 320 servers, full bisection, 12us base RTT

Switch buffer: 16MB, Dynamic Threshold

Congestion control: DCQCN+window, HPCC

SFC Parameters

- SFC trigger threshold = ECN threshold = 100KB, SFC drain target = 10KB

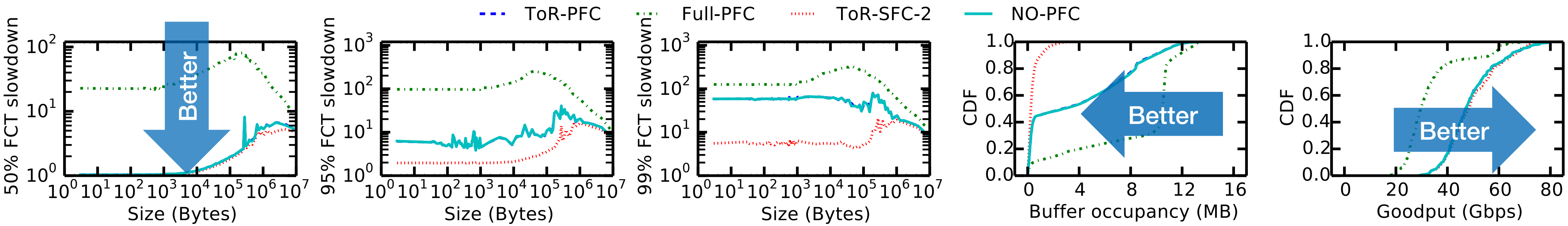
Workload: RDMA writes

- 50% Background load: shuffle, msg size follows public traces from RPC, Hadoop, DCTCP
- 8% incast bursts: 120-to-1, msg size 250KB, synchronized start within 145us

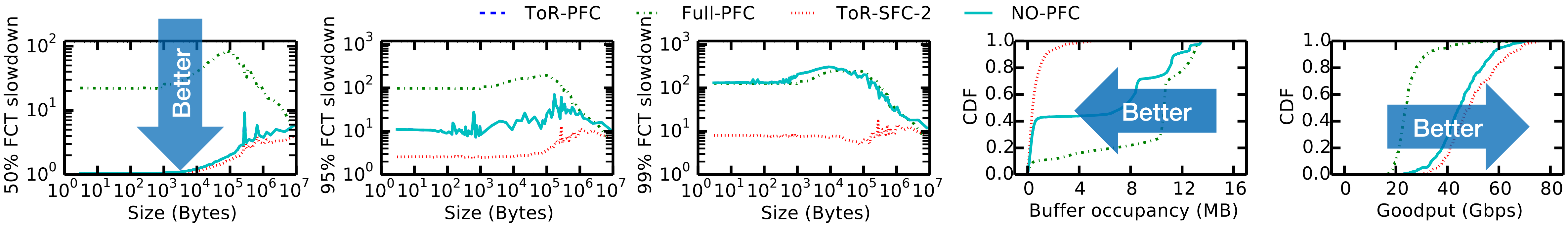
Metrics

- FCT slowdown: FCT normalize to the FCT of same-size flow at line rate
- Goodput, switch buffer occupancy

Large Scale Simulation with RPC Workload



DCQCN+Window



HPCC (INT-based High-Precision Congestion Control)

Switch Config

	Switch Config1 (Remote PFC “off”, PFC “on”)	**Switch Config2 (Remote PFC “on”, PFC “off”)**
Test by	Intel	
Test date	04/08/2021	
SUT Setup		
Platform	Accton AS9516 32d-r0	
# Switches	2 (ToR1, ToR2)	
HWSKU	Newport	
Ethernet switch ASIC	Intel® Tofino™ 2 Programmable Ethernet Switch ASIC	
SDE version	9.5.0-9388-pr	
OS	SONiC.master.111-dirty-20210201.022355	
Buffer Pool allocation	Ingress Lossless pool size is 7.6MB and lossy pool size is 7.6MB. Egress lossless pool size is 16.7MB, and lossy pool size is 6.4MB.	
Remote PFC threshold	N/A	100KB
PFC threshold	Headroom size is 184KB, dynamic threshold is 4.	N/A

Server Config

	Two server models (A and B) are used at the same time in the testbed	
Server model	Model A	Model B
Test by	Intel	Intel
Test date	04/08/2021	04/08/2021
Server Setup		
Platform	Intel S2600WFT	Supermicro X10DRW-i
# Nodes	3 (Send 6, Recv 1, 2)	5 (Send 1, 2, 3, 4, 5)
# Sockets	2	2
CPU	Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
Cores/socket, Threads/socket	18/36	8/16
Microcode	0x5003003	0xb000038
HT	On	On
Turbo	On	On
Power management (disabled/enabled)	enabled	enabled
# NUMA nodes per socket (1, 2, 4...)	2	2
Prefetcher'e enabled (svr_info)	Yes	Yes
BIOS version	SE5C620.86B.02.01.0008.031920191559	3.0a
System DDR Mem Config: slots / cap / speed	6 slots / 16GB / 2934 (*)	8 slots / 32 GB / 2133
Total Memory/Node (DDR, DCPMM)	96, 0	256, 0
NIC	1x 2x100GbE Mellanox ConnectX-6 NIC	1x 2x100GbE Mellanox ConnectX-6 NIC
PCH	Intel C620	Intel C610/X99
Other HW (Accelerator)	RoCEv2 protocol engine in Mellanox ConnectX-6 NIC	RoCEv2 protocol engine in Mellanox ConnectX-6 NIC
OS	Ubuntu 20.04.2 LTS	Ubuntu 20.04.2 LTS
Kernel	5.4.0-66-generic	5.4.0-66-generic
Workload	Custom trace based on Homa (Sigcomm 2018) "Facebook Hadoop" dataset	Custom trace based on Homa (Sigcomm 2018) "Facebook Hadoop" dataset
Compiler	gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0	gcc (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Libraries	MLNX_OFED_LINUX-5.1-2.5.8.0 (OFED-5.1-2.5.8)	MLNX_OFED_LINUX-5.1-2.5.8.0 (OFED-5.1-2.5.8)
NIC driver	mlx5_core	mlx5_core
NIC driver version	5.1-2.5.8	5.1-2.5.8
NIC Firmware version	20.28.2006 (MT_0000000224)	20.28.2006 (MT_0000000224)

*The memory population is per system. For server Model A only half of the memory channels are used per socket. This is a sub-optimal memory configuration compared to the best-known configuration where all memory channels are populated but is not a performance-critical issue. The performance-critical path for the workload runs in the RoCEv2 hardware engine of the RDMA NIC and accesses the memory controllers of the CPUs directly. The maximum network throughput on the NIC is limited to the port speed of 100Gbps. The maximum load on the memory controller is limited to 12.5GB/s and hence the memory controller is not a performance limiter.