Source Flow Control (SFC)

Jeongkeun “JK” Lee

Principal Engineer, Intel

jk.lee@intel.com

Paul Congdon

CTO, Tallac Networks

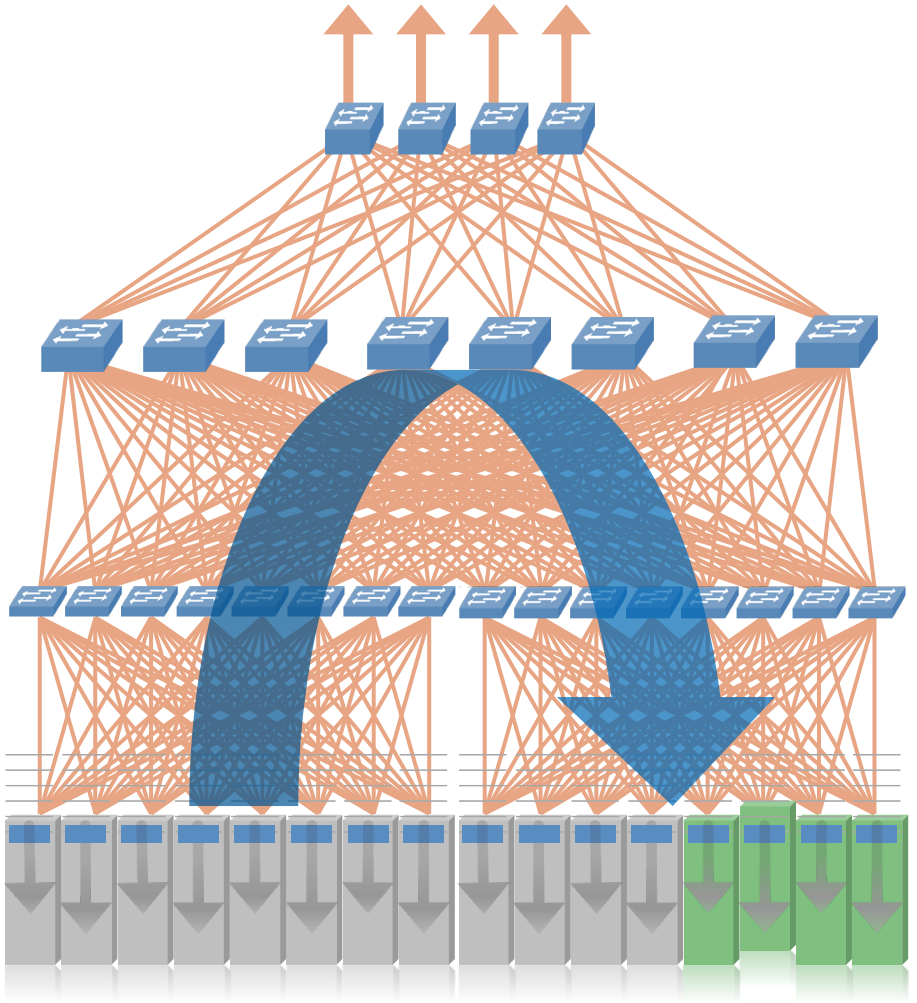
paul.congdon@tallac.com

Agenda

- Background
- Source Flow Control (SFC)
- Performance results
- Proposed changes to Qcz

Types of congestion in data centers

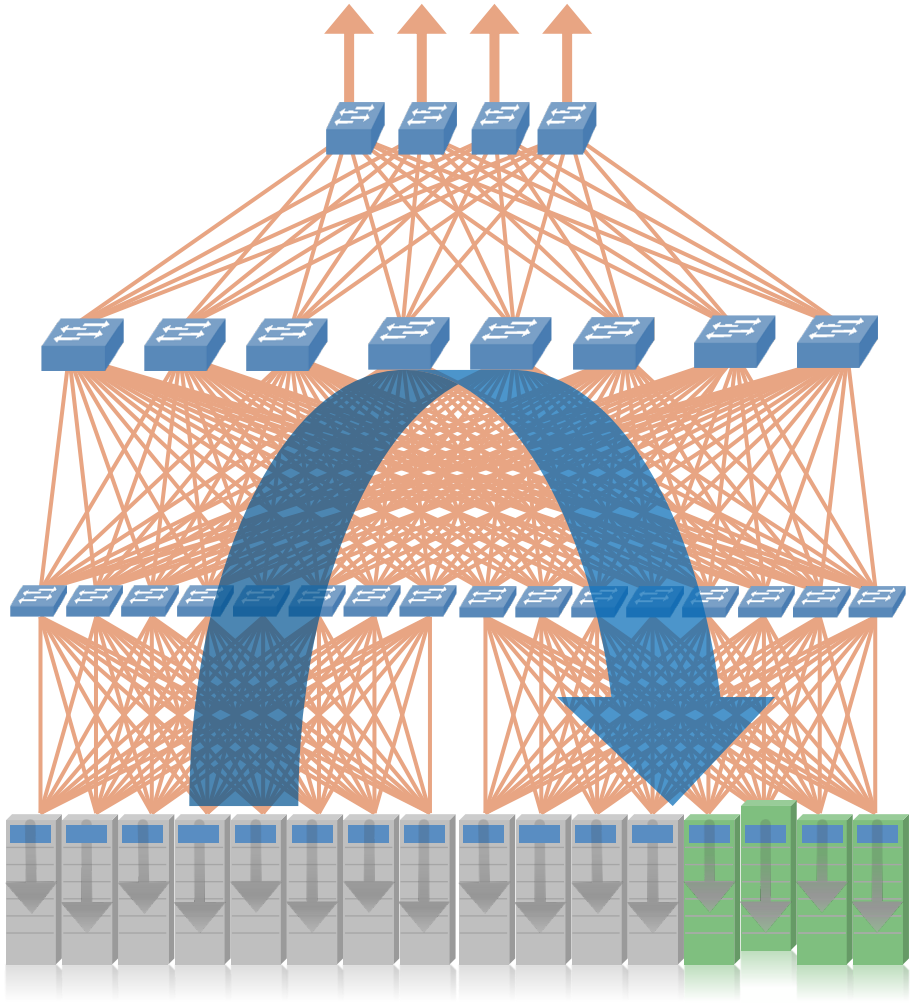
- Uplink/core
  - Cause: ECMP/LAG hash collision
  - Worse at oversubscribed networks
  - Governs median latency
- **Incast**
  - Cause: many-to-one traffic pattern
  - Mostly at the last-hop
  - Governs max/tail latency
- Receiver NIC
  - Cause: slow software/CPU, PCIe bottleneck



Solution space

- E2e congestion control
  - Principal
    1. Detect congestion anywhere in e2e path
    2. from forward-direction data pkts
    3. respond at senders by adjusting TX rates
  - Part of e2e transport such as TCP, QUIC, RoCEv2
- Hop-by-hop flow control
  - Signal previous hop queue xon/xoff
  - Example: PFC
    - Goal: lossless Ethernet (Layer 2)
    - Stops upstream queues when the buffer is nearly full
    - Incurs side effects, e.g., head-of-line blocking (HoL), PFC storm, deadlock

Need for a new edge-to-edge flow control



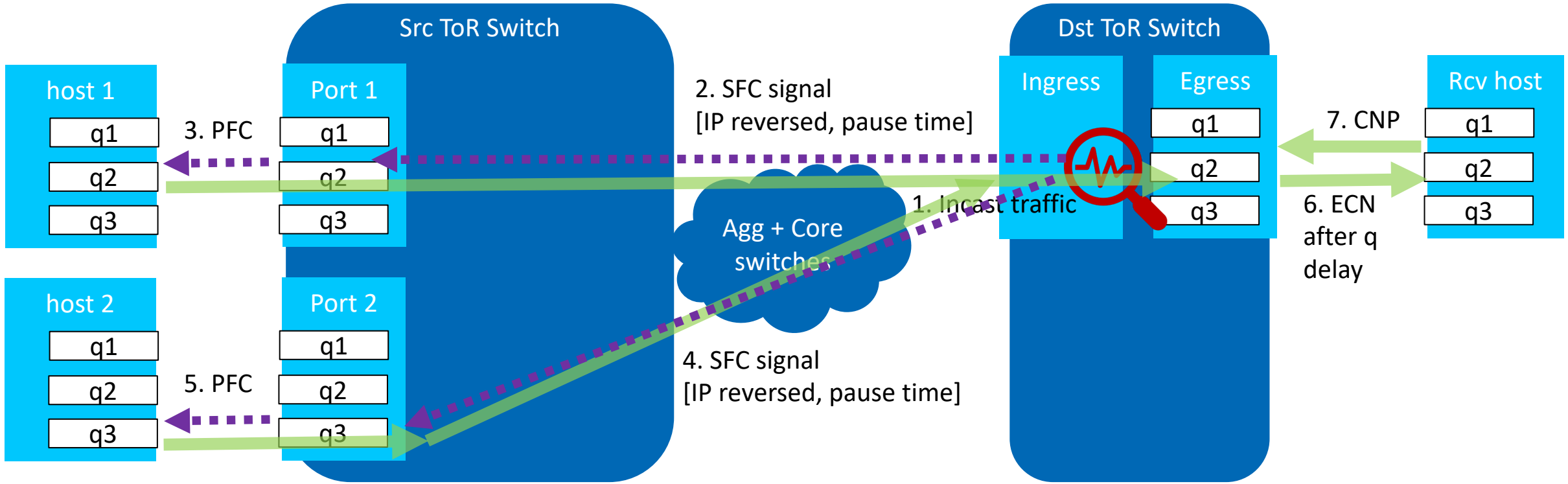
Source Flow Control (SFC)

What is SFC?

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

SFC is/does not

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



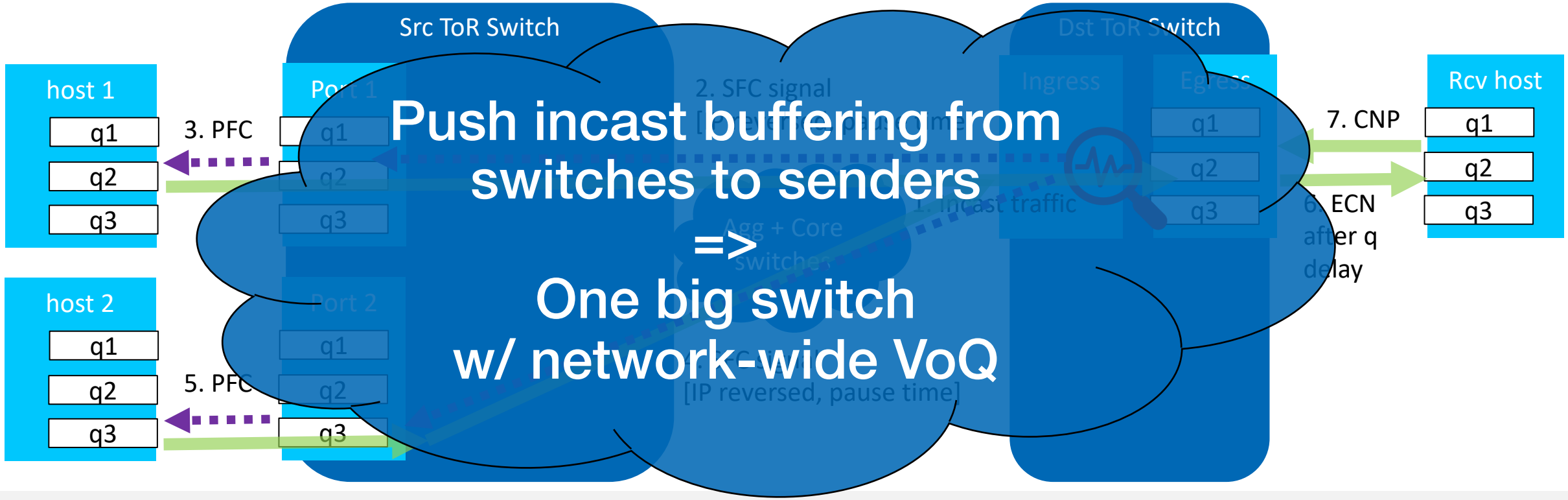
Source Flow Control (SFC)

What is SFC?

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

SFC is/does not

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



FAQs

Why *not* E2E congestion control?

- Faster link speed ➔ shorter RTTs to finish a message ➔ need sub-RTT reaction
- E2E CC relies on forward signal, packets carrying the signals delayed by the congestion
- Cannot react to incast, sudden congestion

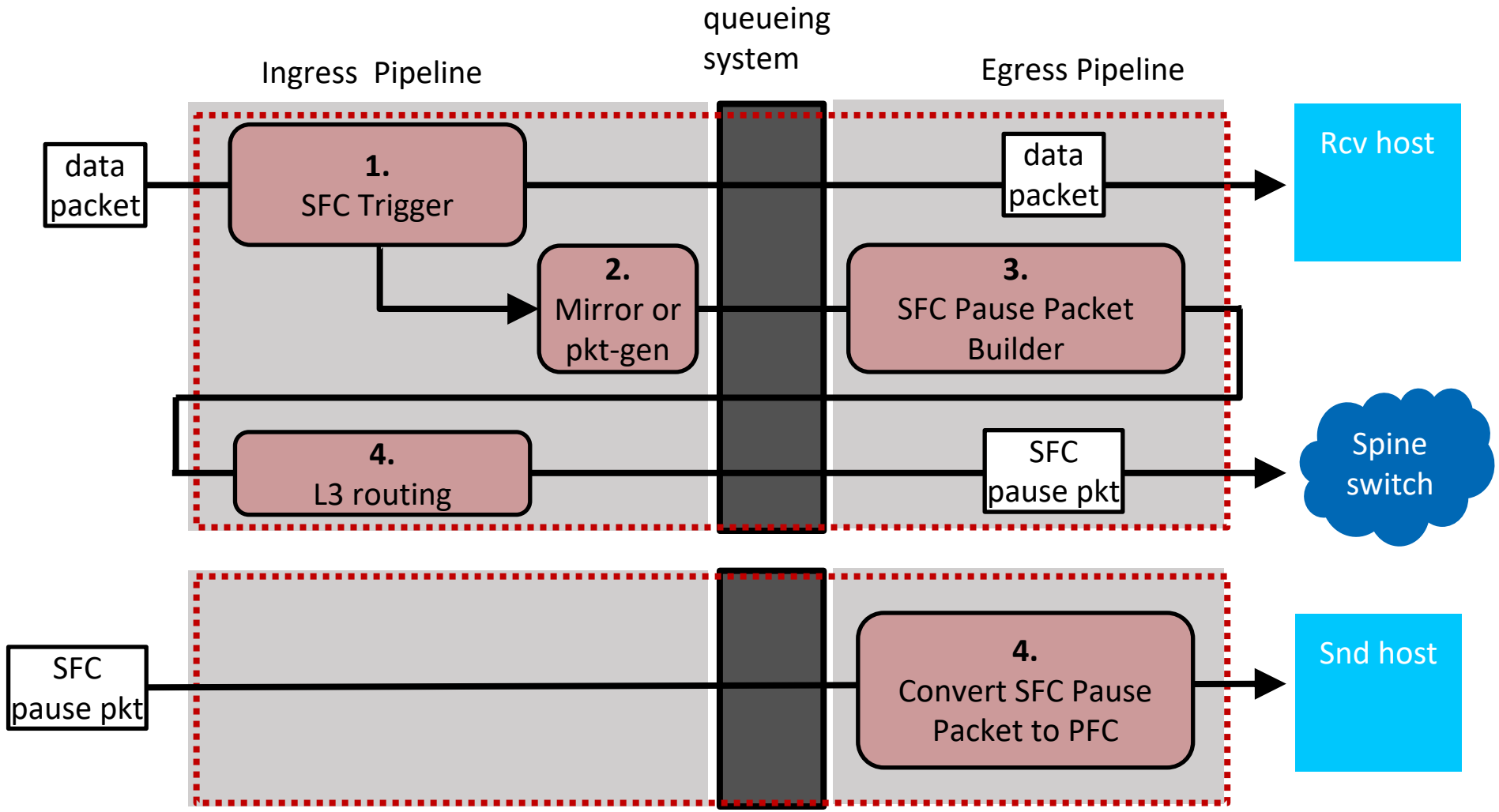
Why not just ‘backward’ CNP from switches?

- CNP cuts rate by half ➔ need multiple RTTs to flatten down the curve
- CNP reaction by sender NIC on TX wire can be slow, up to 24us
- Note) PFC reaction time: sub-microsecond by IEEE 802.1Qbb

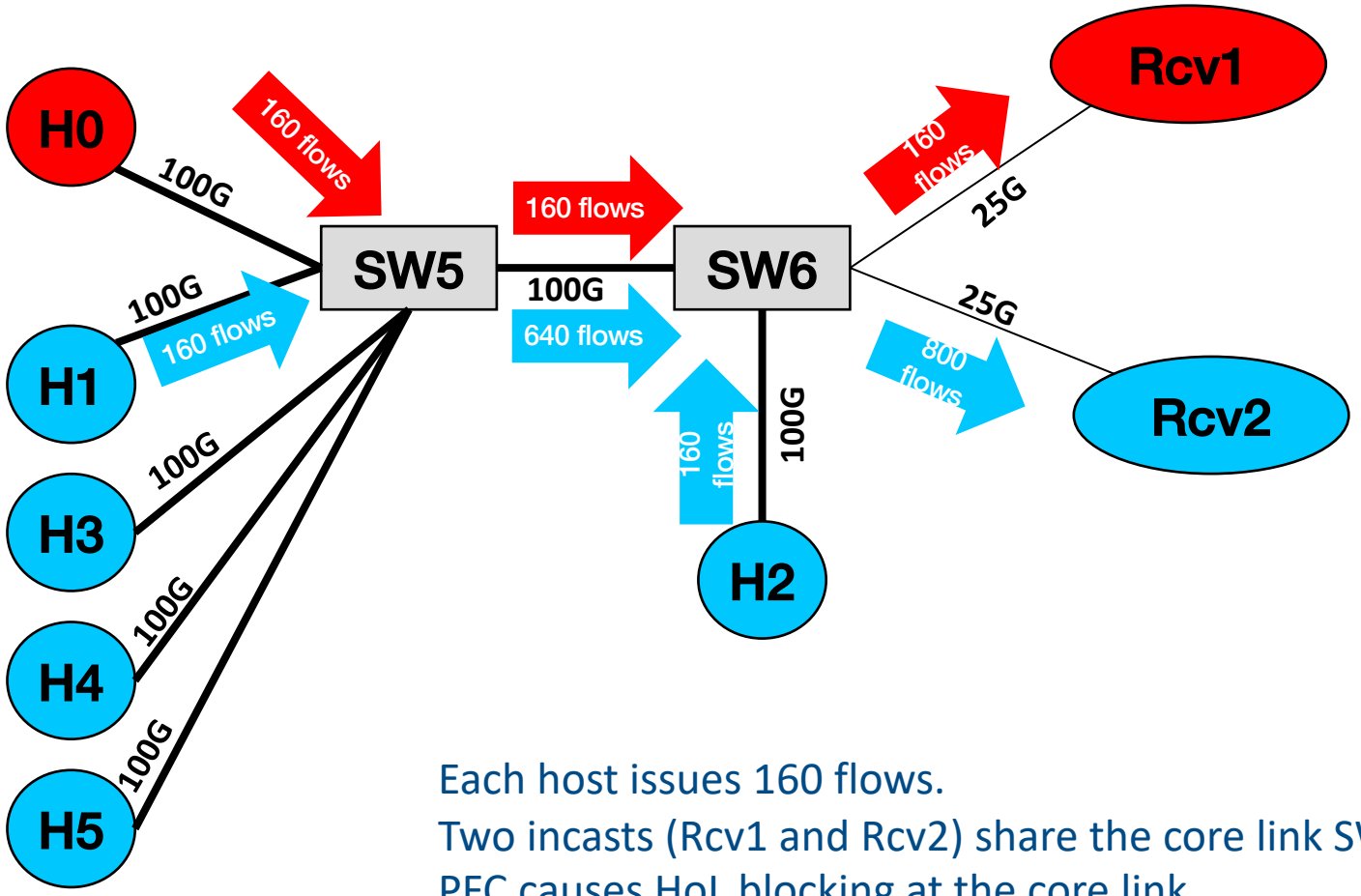
SFC data plane behavior (simplified)

Receiver-side ToR

Sender-side ToR

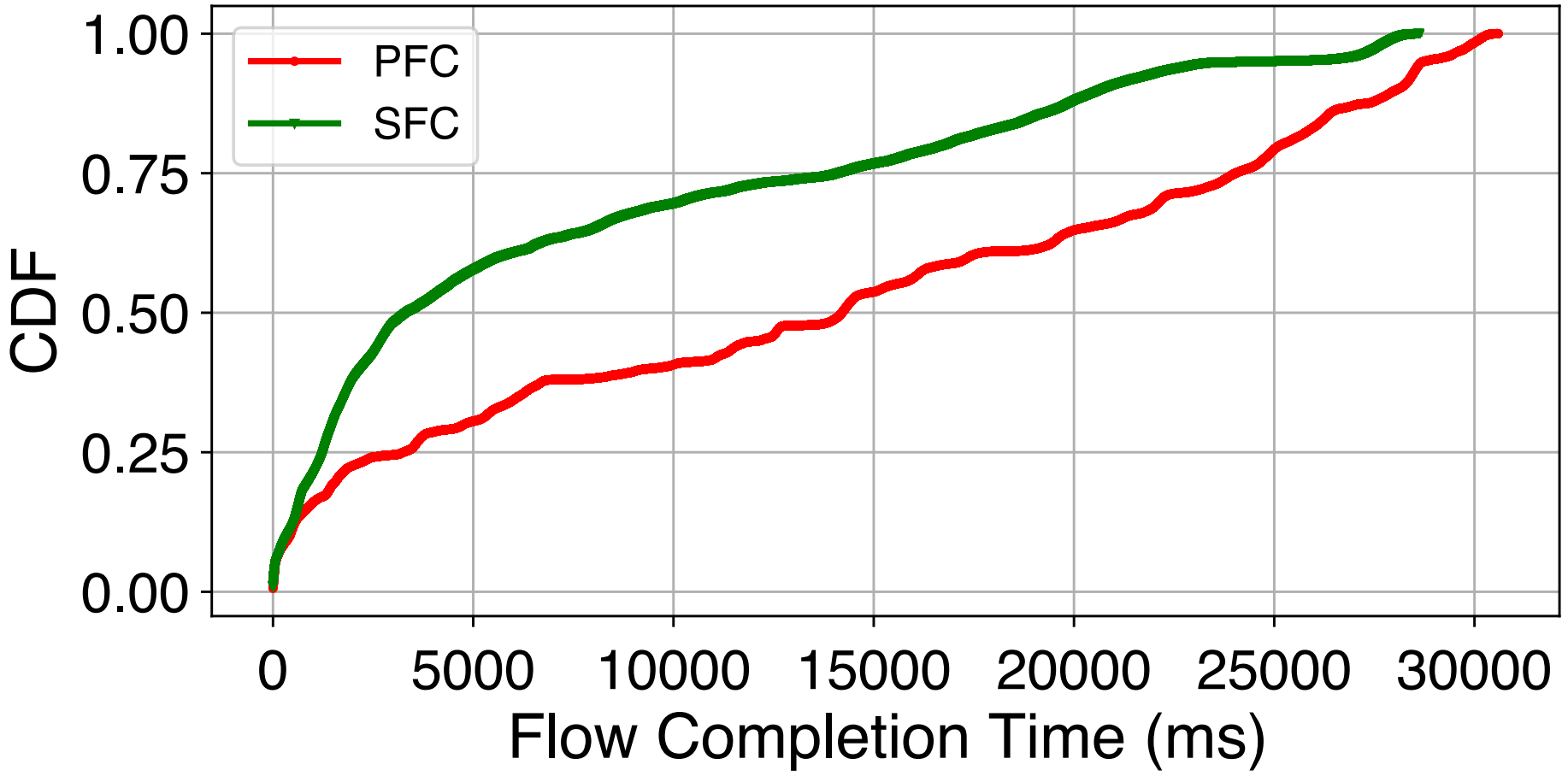


Testbed topology w/ Tofino switches

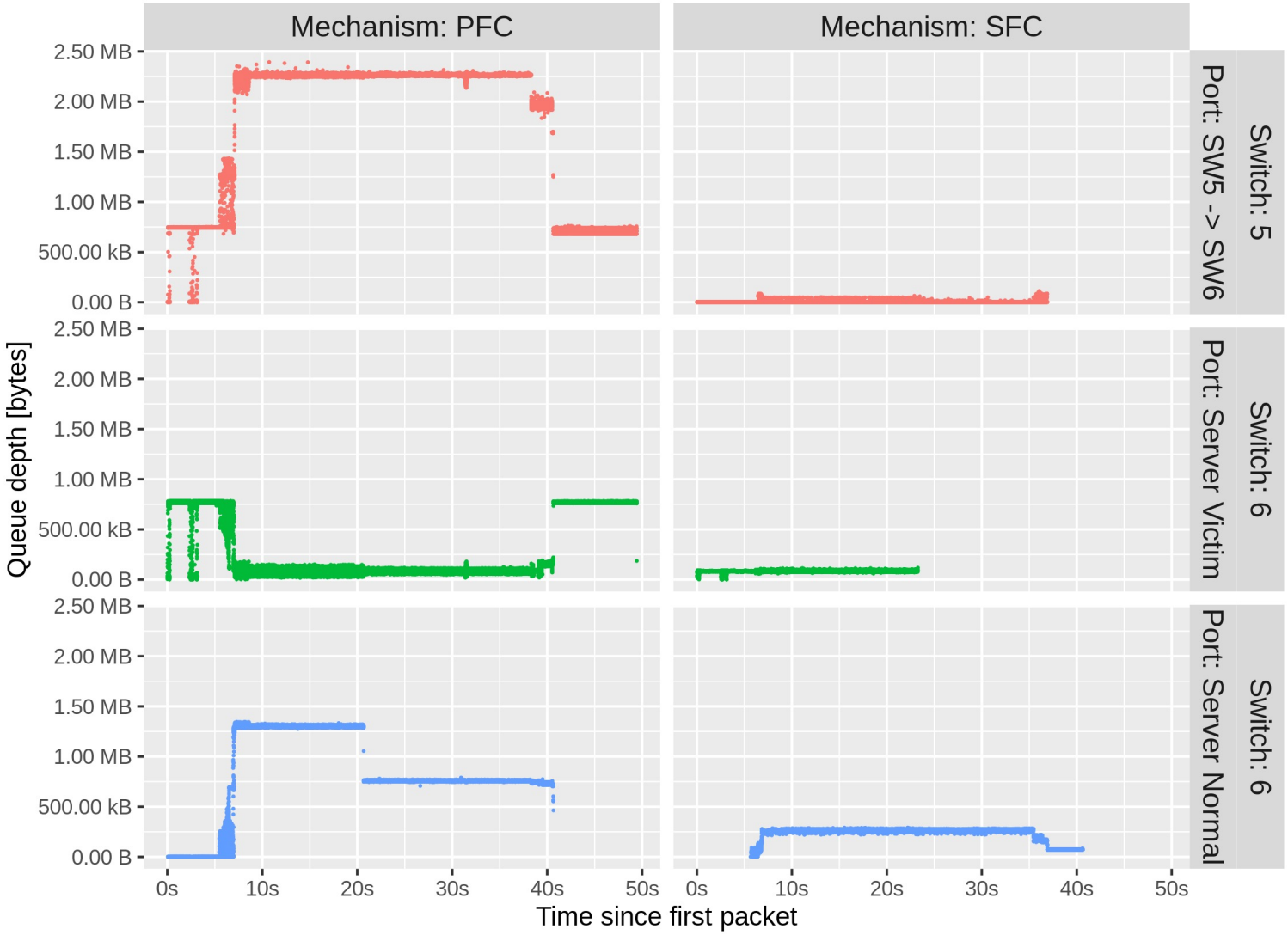


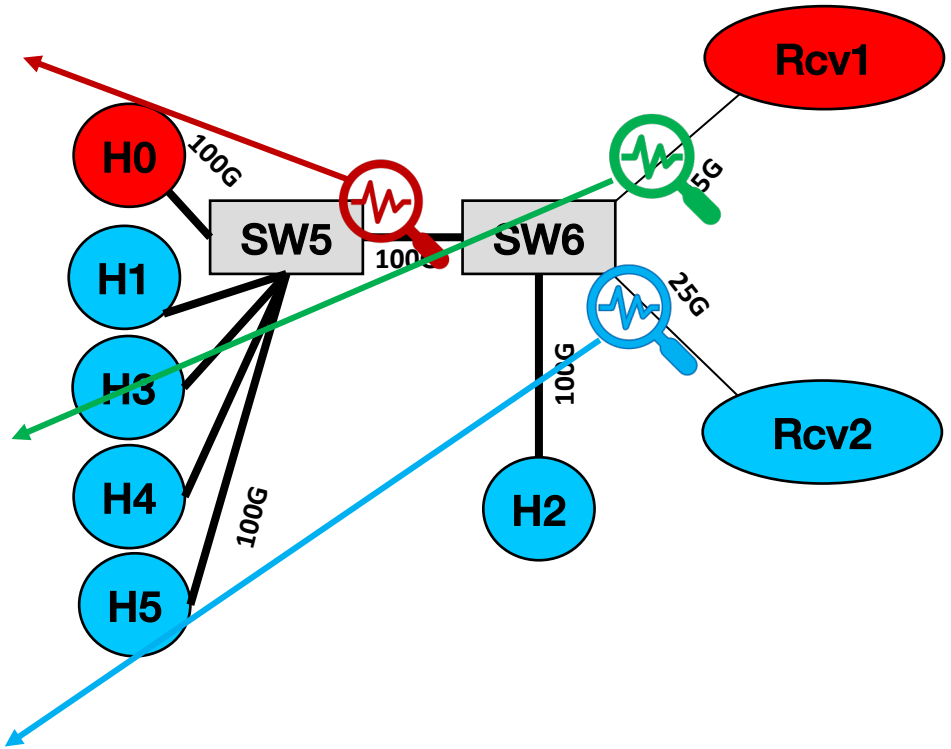
Each host issues 160 flows.
Two incasts (Rcv1 and Rcv2) share the core link SW5→SW6.
PFC causes HoL blocking at the core link.

Flow Completion Time



Queue Depth Reports





Information to carry in 802.1 Qcz header

- Critical for the minimal ‘remote PFC’ mode (SFC converting to PFC)
  - Source and destination IPs of the data pkt
    - SRC IP for reverse forwarding
    - DST IP for caching
    - We can simply swap the IPs in the SFC pause packet
  - DSCP, as needed to identify the PFC priority @ sender NIC
  - Pause time duration = drain time to reach the target queue level
- Additional info for true ‘source’ flow control
  - 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

Leveraging the Qcz Congestion Isolation Message

- 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


intel

Config parameters

- **Ports and queues** to monitor by SFC
- **DSCP code points** to trigger SFC for

- SFC trigger condition
  - **Threshold against queue length**, e.g., max ECN threshold
- Parameter for pause time
  - **Target qdepth to drain**, e.g., min ECN threshold
- SFC suppression, to avoid SFC for every data pkt
  - **Suppression timer**

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

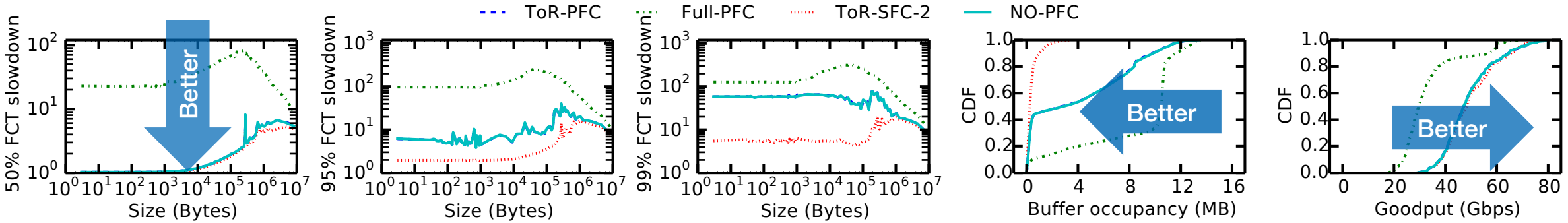
Workload: RDMA writes

- 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

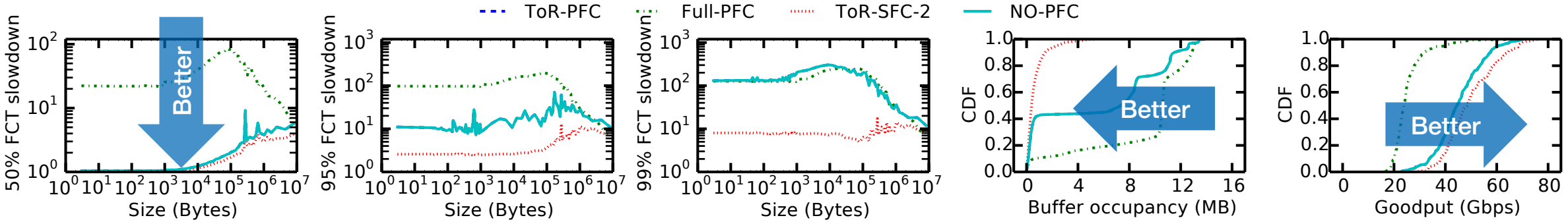
Metrics

- 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)