Hot Interconnects 20, Santa Clara CA, August 22 - 24, 2012

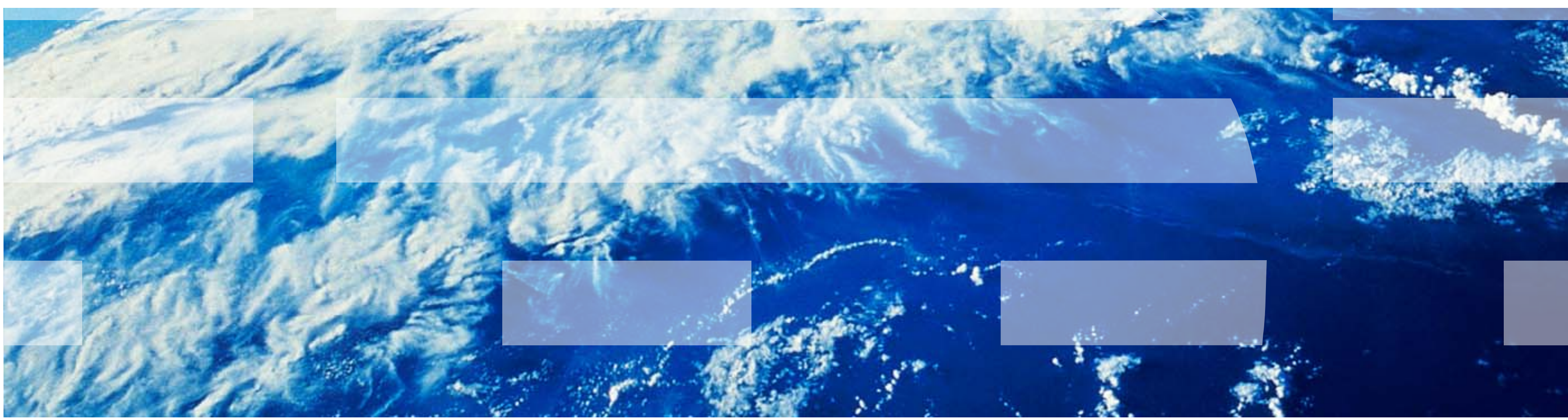
Occupancy Sampling for Terabit CEE Switches

Fredy D. Neeser, Nikolaos I. Chrysos, Rolf Clauberg,
Daniel Crisan, Mitch Gusat, Cyriel Minkenberg

IBM Research, Zurich, Switzerland

Kenneth M. Valk, Claude Basso

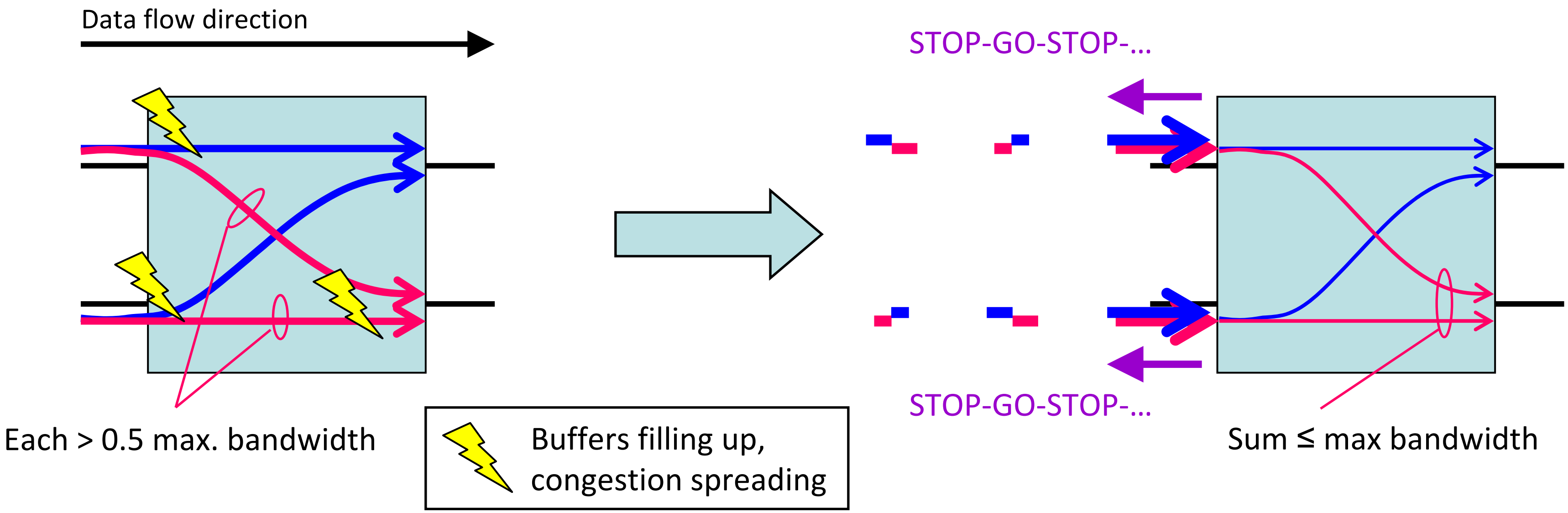
IBM Systems & Technology Group
Rochester, USA



Outline

- Losslessness in Converged Enhanced Ethernet (CEE)
  - HOL blocking in lossless traffic classes
  - IEEE congestion management framework (QCN, 802.1Qau)
  - Scalability of lossless CEE switches
  - Priority flow control (PFC, 802.1Qbb): Side effects in the absence of QCN
- QCN at the inputs of a VOQed high-radix switch
  - Arrival sampling
  - Occupancy sampling
- Simulation results and comparison of the sampling methods
  - Input-generated hotspot
  - Output-generated hotspot
- Conclusions

HOL Blocking in Lossless Traffic Classes



- **Lossless:** Dropping packets to reduce congestion is not an option (Ex. FCoE)
- **Link-level flow control:** Backpressure / stop the upstream senders (no credits, PFC-PAUSE)
- **No oracle ...** to give us a globally optimized bandwidth allocation
- **The problem: Tree saturation (*), multiple-bottleneck hotspot, high-order HOL blocking**
  - In a multistage network, congestion can and will propagate
  - Blocked (hot, culprit) flows prevent innocent (cold) flows from advancing

(*) G. Pfister and V. Kumar, "The onset of hotspot contention", in *Proc. Int. Conf. in Parallel Processing*, Aug. 1986.

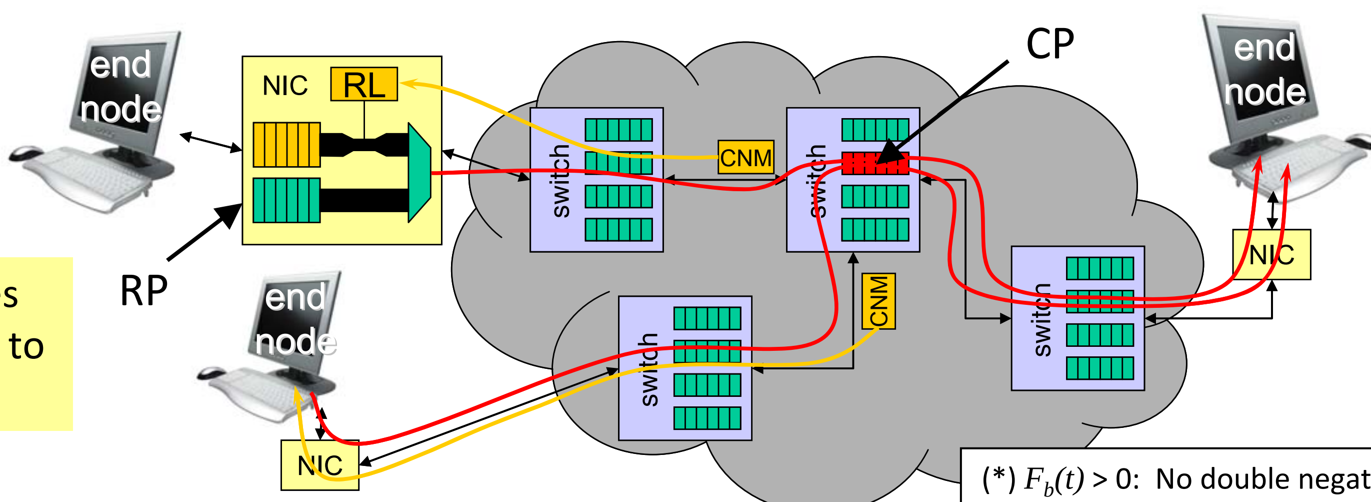
QCN Congestion Management Framework (1 of 2)

<u>Congestion points (CPs) at switch output queues perform “sampling”</u>

1. Determine sampling instant: Every *n* bytes received (where *n* may be randomized)
2. Compute a feedback value $F_b(t)$ that estimates queue congestion
3. Generate a *backward* congestion notification (CNM):
   - Determine a “culprit” flow
   - If $F_b(t) > 0$, send a CNM with *multi-bit congestion info* to the source of the culprit (*)

<u>Reaction points (RPs) with rate limiters (RLs) at the traffic sources</u>

- Enqueue rate-limited flows separately from non-rate-limited ones
- Shape each congestive flow by multiplicatively decreasing its rate limit using $F_b(t)$
- Autonomously increase rate limit based on byte counting or timer (similar to BIC-TCP)



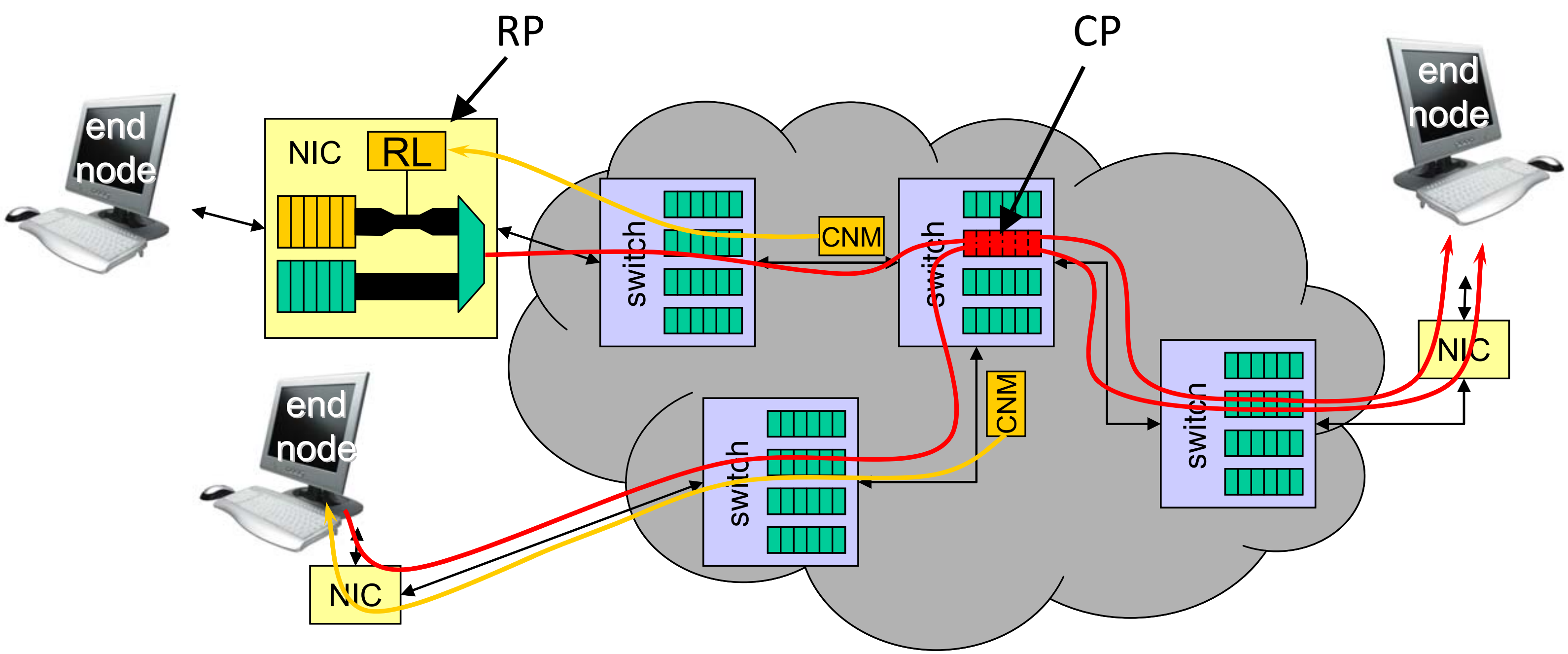
QCN pushes congestion to the edges

(*) $F_b(t) > 0$: No double negation in this paper

QCN Congestion Management Framework (2 of 2)

Congestion point (CP) calculations

- Position (queue length offset) $$q_{\text{off}}(t) = q(t) - Q_{\text{eq}}$$
- Velocity (queue length rate of change) $$q_{\delta}(t) = q(t) - q_{\text{old}}$$
- Feedback value $$F_b(t) = q_{\text{off}}(t) + w \cdot q_{\delta}(t) \qquad (*)$$

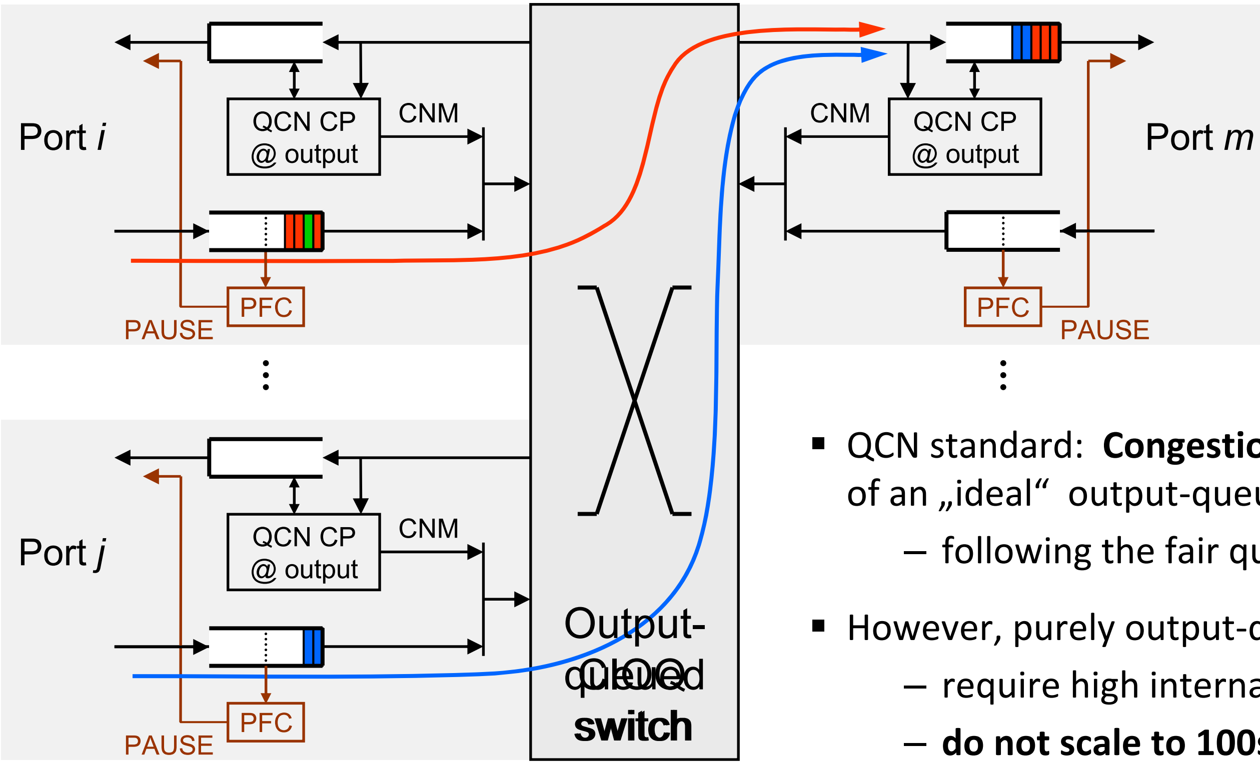


(*) $F_b(t) > 0$: No double negation in this paper

Datacenter operating conditions

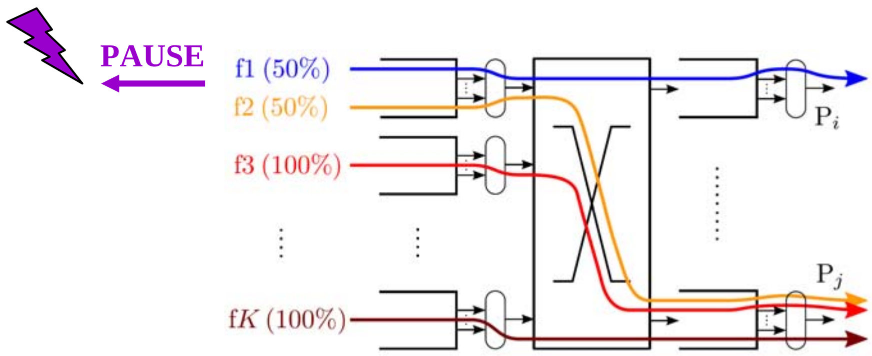
- Link-level flow control (lossless) → potential for saturation trees
- Short and fat:
  - Low latency: Few us, rather than milliseconds
  - High link speeds: 10 … 40 … 100 G
- Protocols w/o L4 TCP congestion mgmt
  - RoCE, UDP, overlay virtual networks
- Shallow buffers for scalability
  - 100s of KB

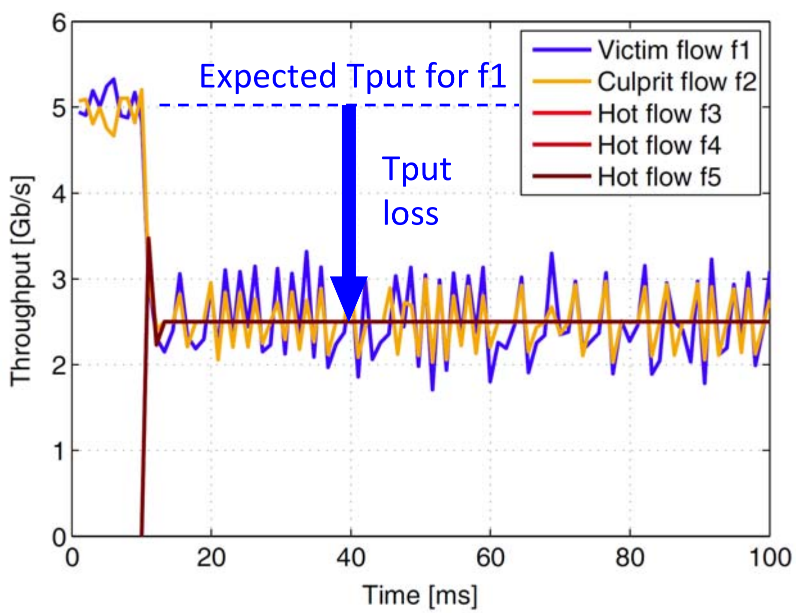
Scalability of lossless CEE switches?



- QCN standard: **Congestion points (CPs) installed @ outputs** of an „ideal“ output-queued switch
  - following the fair queuing tradition
- However, purely output-queued switches
  - require high internal speedup / shared memory
  - **do not scale to 100s of ports / 10+ G link speeds**
- In practice, **lossless operation** using PFC Pause @ 10+ Gb/s requires some **dedicated buffers per input port (headroom)**
- Need input buffers for scalability/PFC → CIOQ switch
  - → Consider VOQs to avoid HOL blocking
  - → Need QCN CPs at inputs? Interaction with VOQs?

Input-buffered lossless switch: Issues with PFC-only operation





- Input-generated hotspot
  - ***K*** **= 5 flows**, 10G links
  - f1 and f2 enabled from start; sharing an input buffer
  - f3 … f5 enabled at 10 ms → hotspot @ P_j
- Results with VOQs, PFC only
  - 0 … 10 ms: Fair shares
  - 10 … 100 ms:
    - f2 … f5 (**4 flows sharing** a 10G link) obtain their **fair 2.5 Gb/s shares**
    - Buffer hogging and PFC-induced HOL → f1 (victim) achieves only 2.5 G vs. the expected 5 Gb/s

Outline

- Losslessness in Converged Enhanced Ethernet (CEE)
  - HOL blocking in lossless traffic classes
  - IEEE congestion management framework (QCN)
  - Scalability of lossless CEE switches
  - Priority flow control (PFC): Side effects in the absence of QCN

- QCN at the inputs of a VOQed high-radix switch
  - Arrival sampling
  - Occupancy sampling

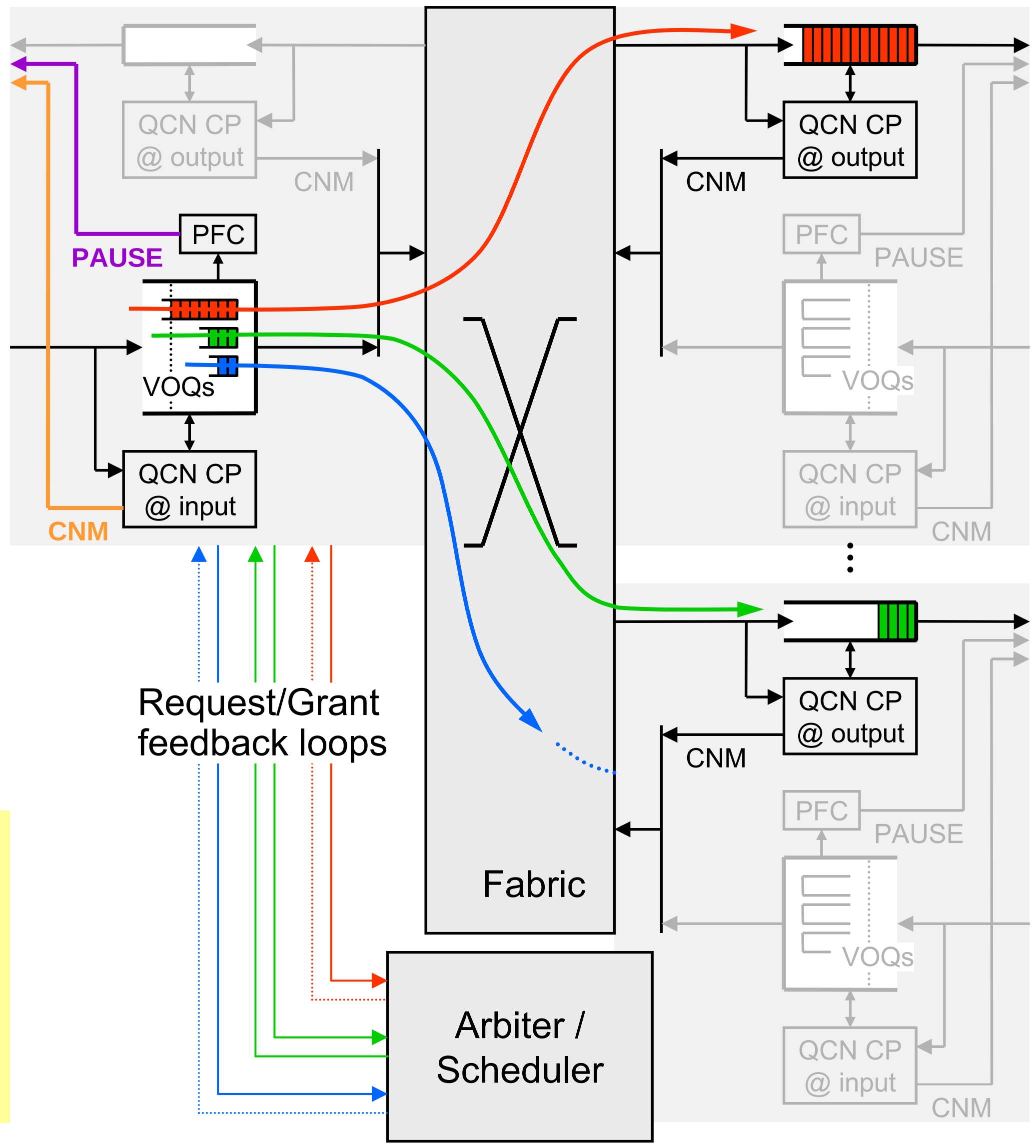
- Simulation results and comparison of the sampling methods
  - Input-generated hotspot
  - Output-generated hotspot

- Conclusions

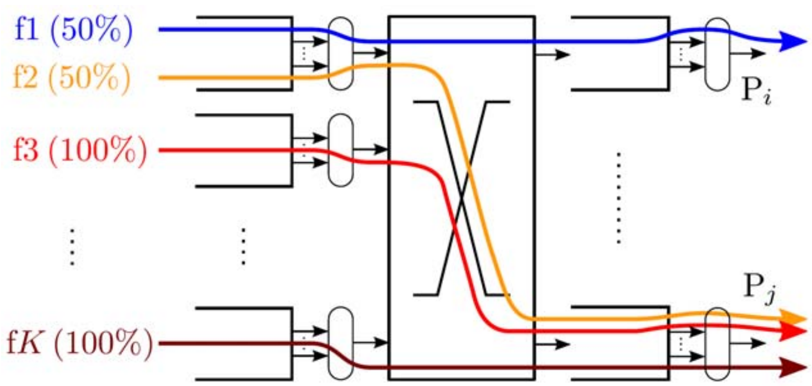
QCN for a lossless CIOQ switch with VOQs

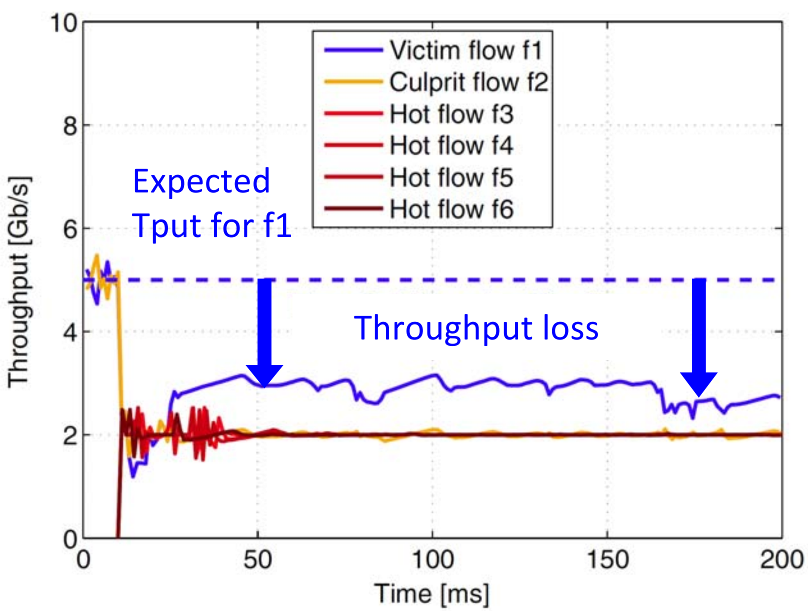
- Scalable CIOQ switch architecture
  - VOQs eliminate HOL blocking in IQs
  - Switch arbitration/scheduling based on request/grant buffer reservation
- VOQ-based flow isolation also requires
  1. private per-VOQ buffers (may not scale)
  2. per-VOQ discriminate flow control b/w switch and adapter (not in Ethernet)
- VOQs share an input buffer
  - Sharing can lead to buffer hogging
  - Eventually, input asserts PFC PAUSE
  - Indiscriminate PFC cannot prevent HOL blocking *within a priority*

- **Questions**
  1. Do QCN CPs @ inputs prevent hogging?
  2. Non-FIFO -- Input buffer has multiple servers → Does QCN find the culprits?
  3. Install QCN CPs at inputs, outputs, both?



Experiment: IG hotspot, CIOQ/VOQ, PFC + QCN-AS @ inputs



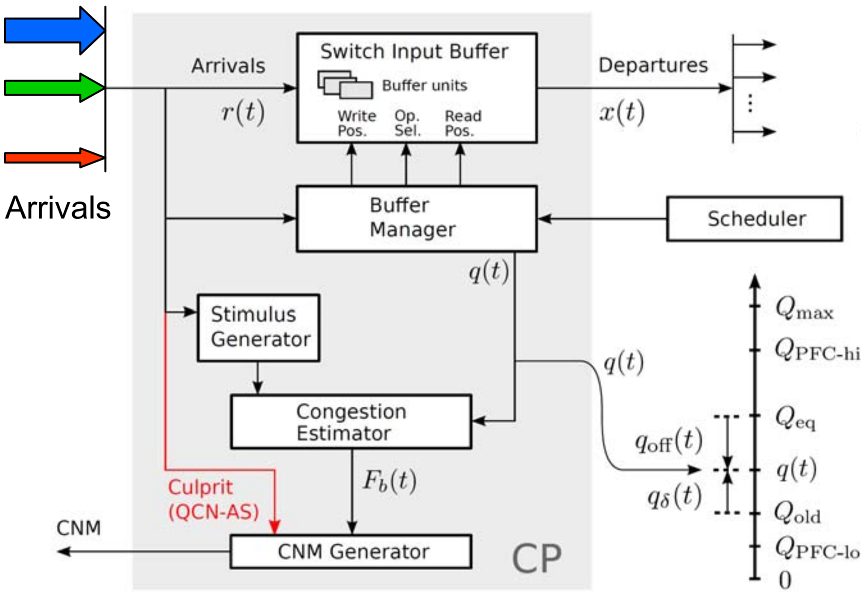


- Input-generated hotspot
  - ***K* = 6 flows**, 10G links
  - f1, f2 enabled from start; sharing input buffer
  - f3 … f6 enabled at 10 ms → hotspot @ P_j
  - f2 … f6 (5 flows) share a 10G output **(fair = 2G)**

- Standard **arrival sampling** (QCN-AS) @ inputs
  - 0 … 10 ms: Fair shares
  - 10 … 100 ms:
    - f2 … f6 obtain their fair 2G shares
    - f1 (victim) throttled to ≈ 2G vs. expected 5G

- Tput problem with QCN-AS @ inputs
  - **Not caused by buffer hogging:** $q(t) \approx Q_{eq}$
  - Arrival sampling selects culprit flows in proportion to the flow arrival rates

A close look at QCN arrival sampling (QCN-AS) @ switch input



- Set of N flows with average frame sizes $\overline{S}_n$ arriving at the CP with rates (in bytes/s)

$$R(t) = \sum_{i=1}^{N} R_i(t) = \sum_{i=1}^{N} \overline{S}_i \lambda_i(t)$$

QCN arrival sampling
- samples flows in proportion to their average arrival byte rates
- equalizes the flow injection rates whenever there is congestion

- What exactly does „arrival sampling“ do?
  - (a) Determine next sampling instant:
    Count total # bytes received since previous sampling instant and compare to current sampling interval 'K' (base: I_S = 150 KB).
  - (b) Triggered by (a), compute $F_b(t)$
  - (c) If $F_b(t) > 0$, **culprit := most recent arrival**

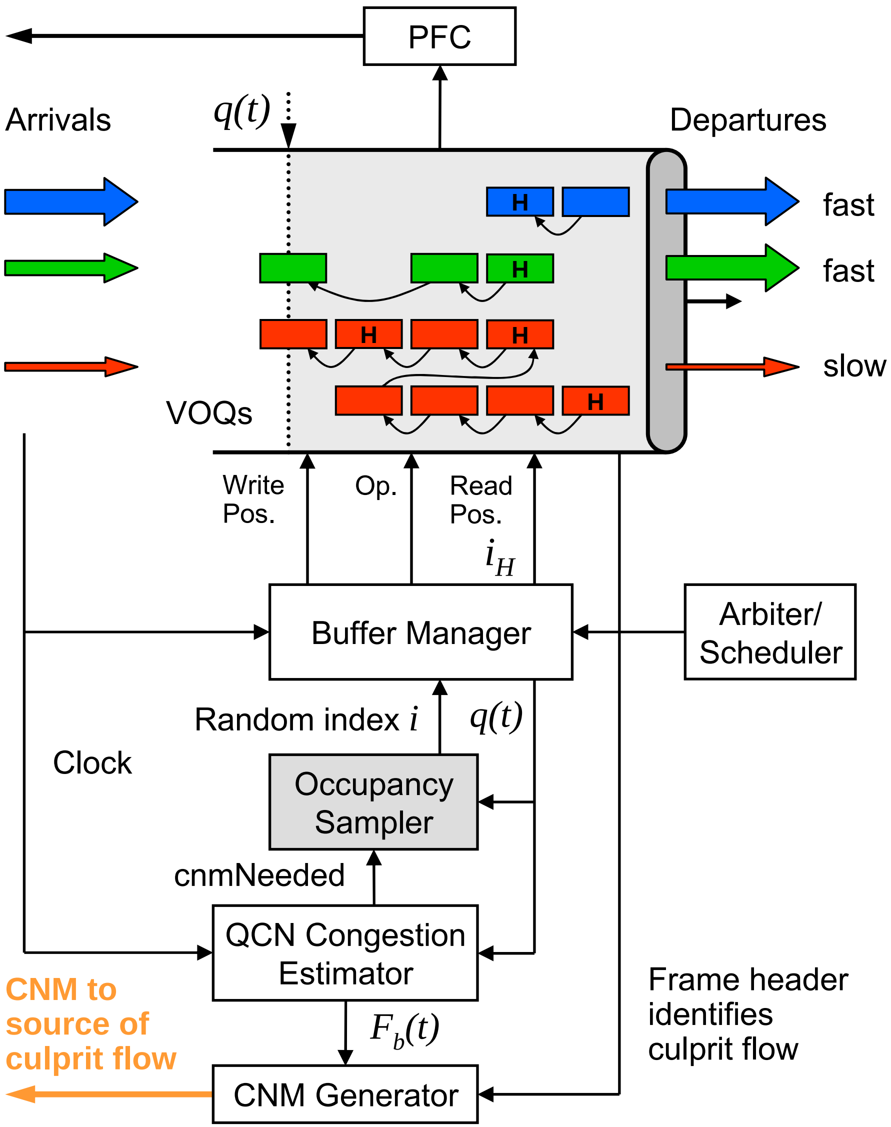
- **Flow sampling probability:**
  Conditioned on the event that the CP has counted K bytes (assuming randomization of K from interval to interval), the probability that the most recent frame belongs to the n-th flow is:

$$P_n^{(s)} = R_n / R \qquad (1)$$

- **Flow reflection (marking) probability,**
  conditioned on {CP has counted K bytes at time t }:

$$P_n^{(r)}(t) = P_n^{(s)}(t) \cdot \Pr\{F_b(t) > 0\} = \frac{R_n(t)}{R(t)} \cdot \Pr\{F_b(t) > 0\} \qquad (2)$$

QCN-OS @ input: Fairness by finding the real culprits



- Input buffer with multiple servers → $q_n(t)$ generally **not** proportional to $r_n(t)$
- **Idea:** Mark flows not in proportion to their arrival rates, but in proportion to their contribution to overall input buffer occupancy
- Random occupancy sampling:
  1. Randomly pick an occupied buffer unit i from a pool of fixed-size buffer units
  2. Identify corresponding frame header index i_H
- **Flow sampling probability:** Conditioned on the event that the CP has counted K bytes, the probability that the most recent frame belongs to the n-th flow is

$$P_n^{(s)} = q_n(t)/q(t) \qquad (3)$$

- **Flow reflection (marking) probability**, conditioned on {CP has counted K bytes at t }:

$$P_n^{(r)}(t) = \frac{q_n(t)}{q(t)} \cdot \Pr\{F_b(t) > 0\} \qquad (4)$$

Outline

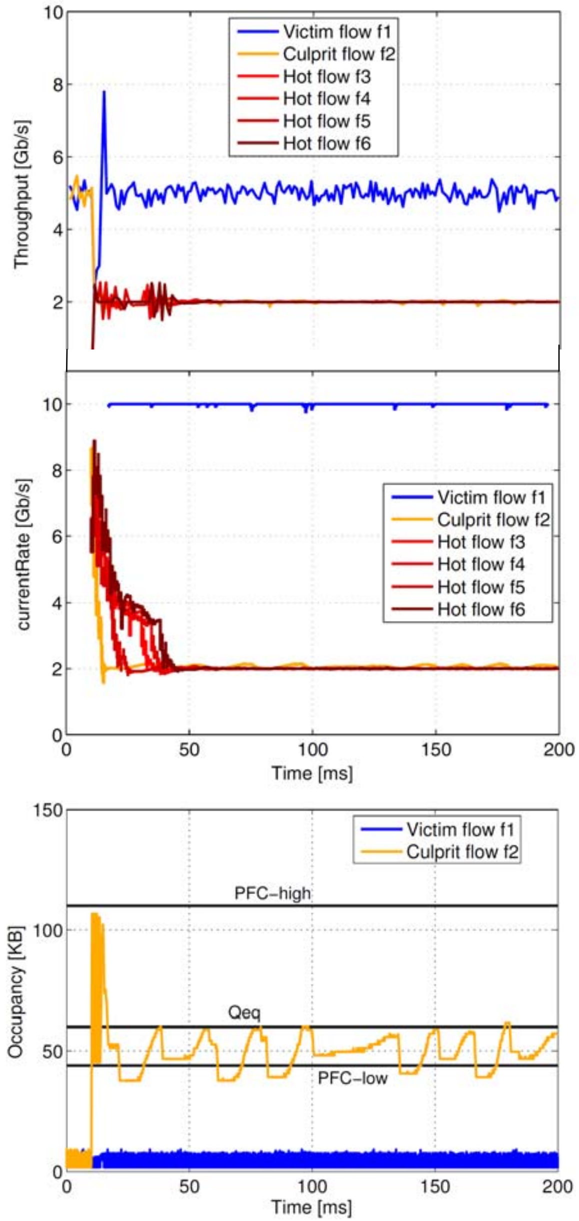
- Losslessness in Converged Enhanced Ethernet (CEE)
  - HOL blocking in lossless traffic classes
  - IEEE congestion management framework (QCN)
  - Scalability of lossless CEE switches
  - Priority flow control (PFC): Side effects in the absence of QCN

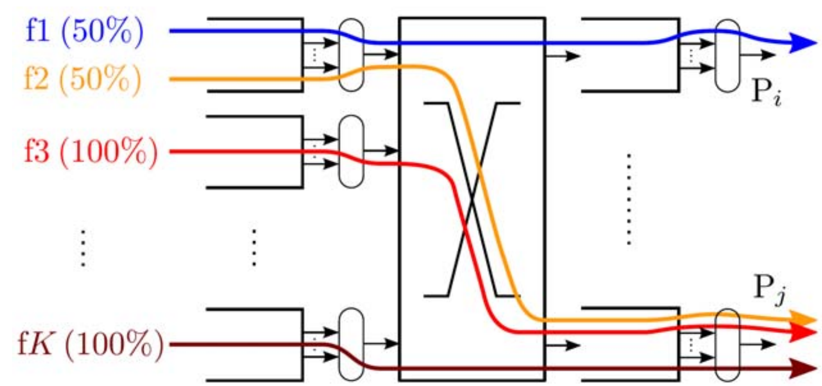
- QCN at the inputs of a VOQed high-radix switch
  - Arrival sampling
  - Occupancy sampling

- Simulation results and comparison of the sampling methods
  - Input-generated hotspot
  - Output-generated hotspot

- Conclusions

IG hotspot, CIOQ/VOQ, PFC + QCN Occupancy Sampling @ inputs

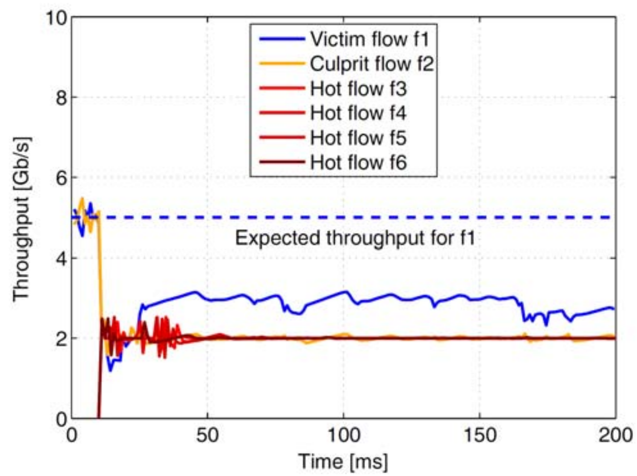


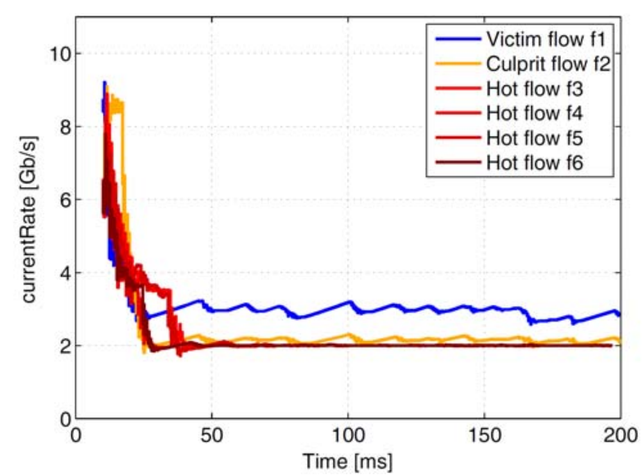


- Input-generated hotspot
  - ***K* = 6 flows**, 10G links
  - f1, f2 enabled from start; sharing an input buffer
  - f3 … f6 enabled at 10 ms → hotspot @ P_j
  - f2 … f6 (5 flows) share a 10G output; fair share = 2G
- **Occupancy sampling** (QCN-OS) @ inputs
  - 0 … 10 ms: Fair shares
  - 10 … 100 ms:
    - f2 … f6 (5 flows) obtain their fair 2G shares
    - **f1 virtually unaffected by the hotspot @ P_j**

IG hotspot, CIOQ/VOQ: QCN-AS vs. QCN-OS, with CPs @ inputs

Arrival sampling

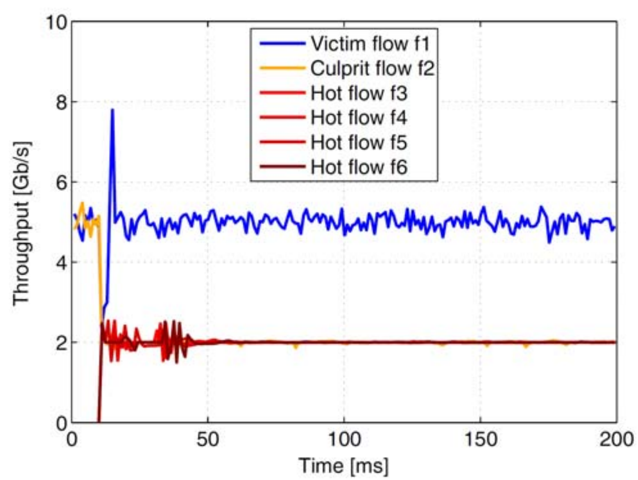


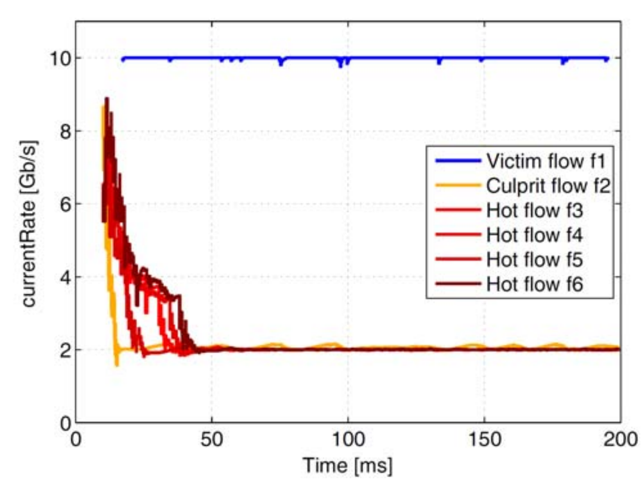


Input-generated hotspot (as on p. 15)

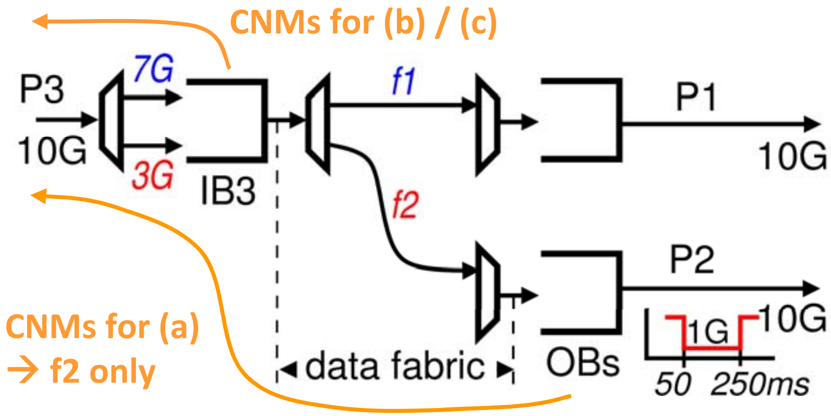
- ***K* = 6 flows**, 10G links
- f2 … f6 (5 flows) share a 10G output **(fair = 2G)**

Occupancy sampling





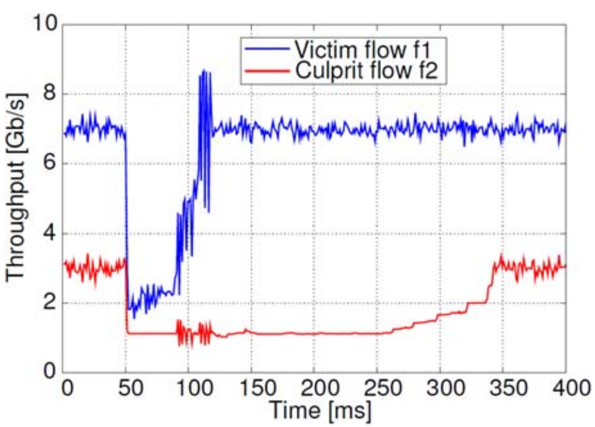
Output-generated (OG) hotspot



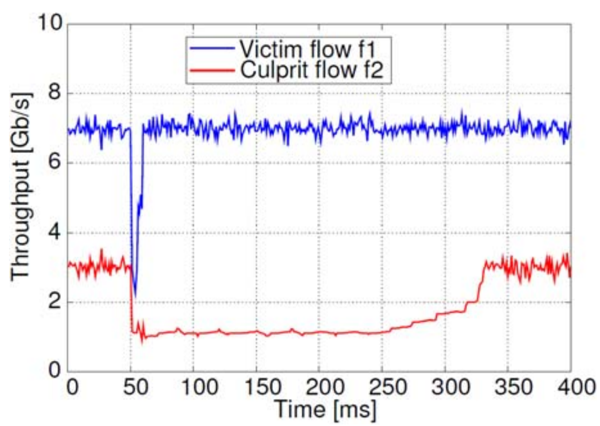
OG hotspot scenario

- From 50 to 250 ms, the available capacity of output P2 drops from 10G to 1G
- May occur due to higher-priority traffic or because of server/CPU overload at destination

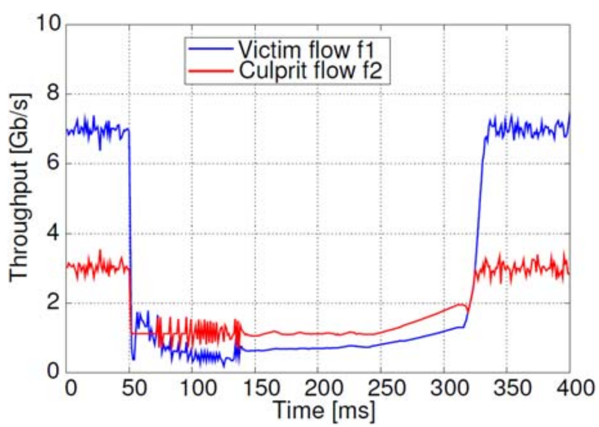
(a) QCN-AS, CPs at outputs



(b) QCN-OS, CPs at inputs



(c) QCN-AS, CPs at inputs



- No CNMs during PAUSE
- f1 suffers from HOL for ≈ 70 ms (CIOQ switch!)

Resolves congestion faster than (a)

Wrongly identifies f1 as culprit

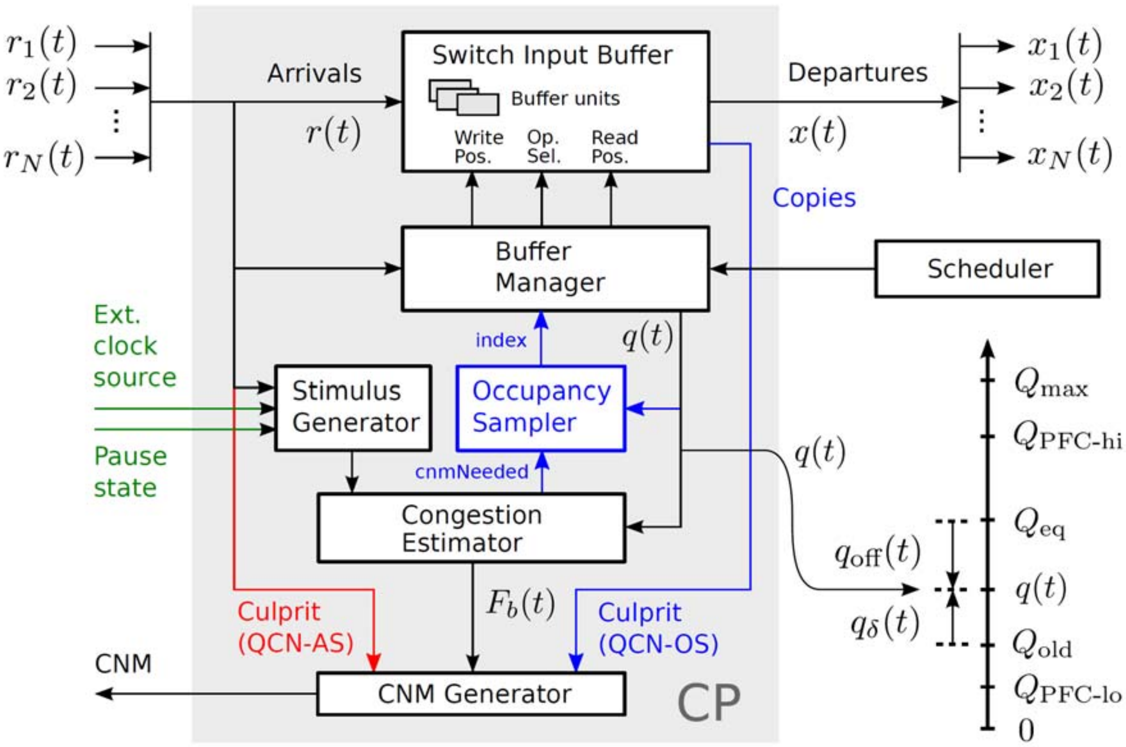
Conclusions

- QCN counteracts the detrimental side effects of link-level flow control, preventing Tput collapse in lossless CEE traffic classes
  - Supporting protocols without L4-based CM (FCoE, RoCE, UDP, overlay networks ...)
- **QCN occupancy sampling**
  - Novel QCN-compatible marking scheme for *lossless* aggregate queues,
    - suitable for large/high-speed CIOQ switches using VOQs
    - suitable for installation at switch input buffers
  - Provides VOQ-like flow granularity, but extending to the adapter via multiple hops
  - Correctly identifies congestive culprits in aggregate queues w/ multiple servers
  - Faster resolution of OG hotspots than with arrival sampling
  - Fair rate allocation under IG and OG hotspots
  - Eliminates buffer hogging + reduces need for the big hammer (PFC PAUSE)

THANK YOU !!!

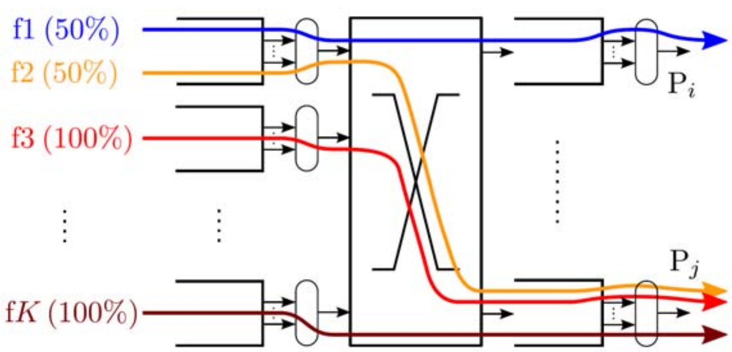
BACKUP

Simulation parameters



Parameter	Value
Input buffer per port	150 KiB
Output buffer per port	150 KiB
PFC-hi (STOP) threshold	110 KiB
PFC-lo (GO) threshold	44 KiB
Q_{eq}	60 KiB
w (weight for velocity)	2
I_s (base sampling interval)	150 KiB

IG hotspot, CIOQ/VOQ: QCN-AS @ outputs vs. QCN-OS @ inputs

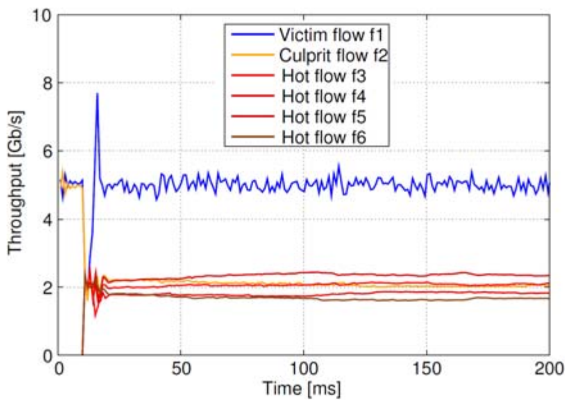


Parameter	Value
Input buffer per port	150 KiB
Output buffer per port	150 KiB
PFC-hi (STOP) threshold	110 KiB
PFC-lo (GO) threshold	44 KiB
Q_{eq}	60 KiB
w (weight for velocity)	2
I_s (base sampling interval)	150 KiB

Input generated hotspot (as before)

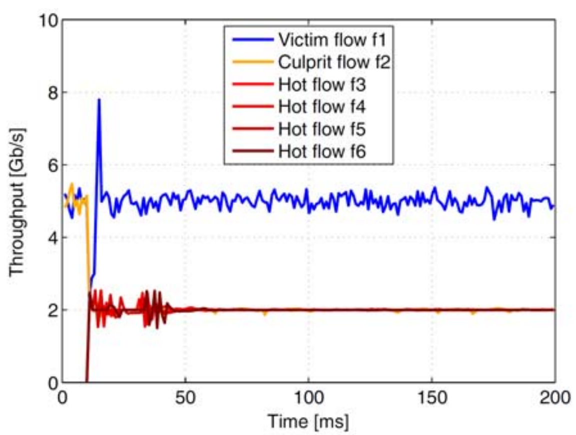
- ***K* = 6 flows**, 10G links
- f2 … f6 (5 flows) share a 10G output **(fair share = 2G)**

Arrival sampling (QCN-AS @ outputs)



± 0.4G rate unfairness at congested output port

Occupancy sampling (QCN-OS @ inputs)



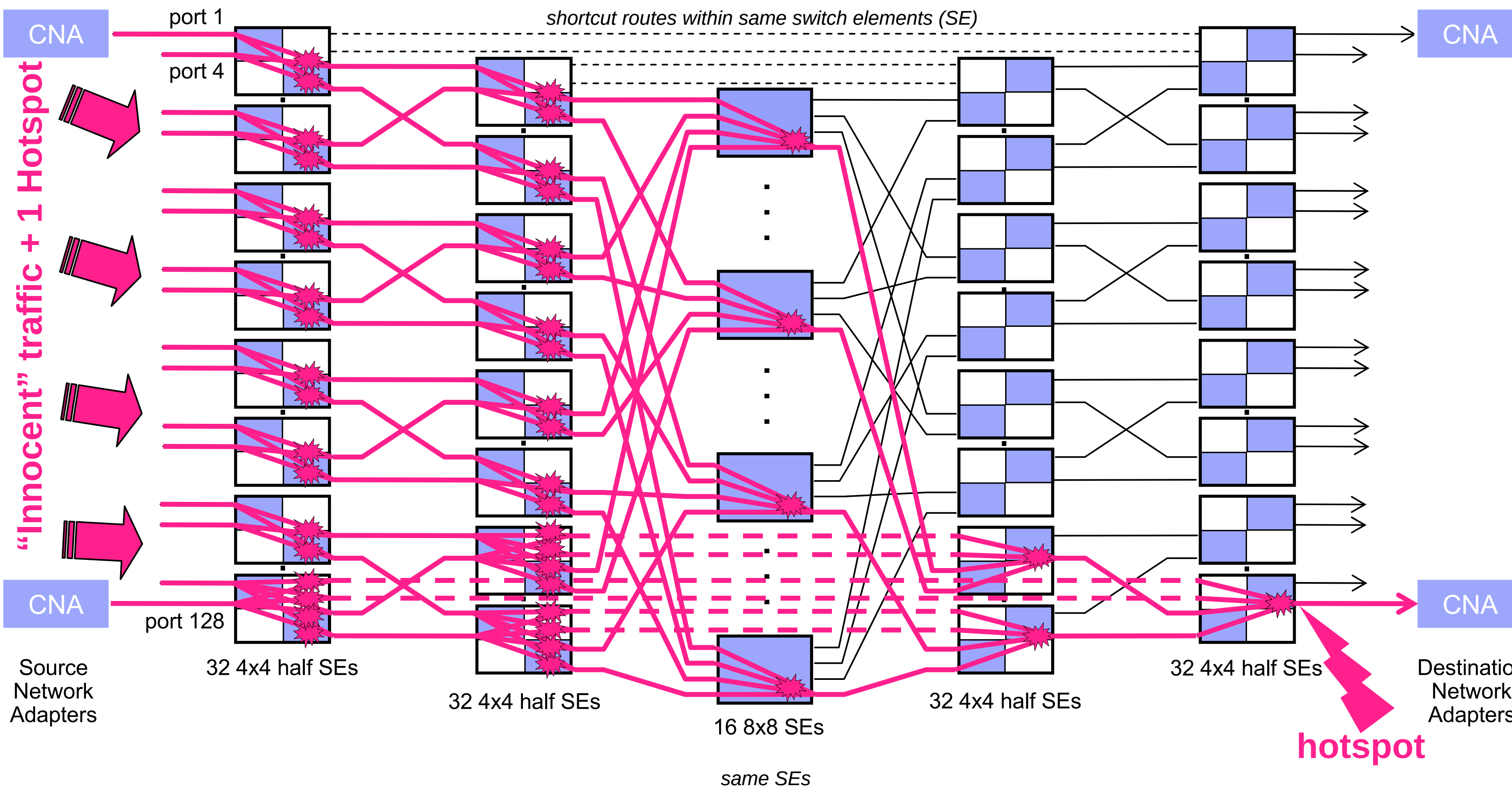
Strictly fair rate allocation owing to switch arbitration

Occupancy sampling applications

- Per-priority RQ in a virtualized converged network adapter (CNA)
  - Keep queuing delay low → limit receive buffers per VM →
  - If a VM is slow (e.g., due to CPU quota) → CNA L2 receive queue grows
  - 100s of VMs (some fast and some slow)
    - Having 8 dedicated priority queues per VM might not scale
  - CP with QCN-OS keeps aggregate CNA RQ shallow
    - Maintain low queuing delay
    - Avoid buffer hogging due to misbehaving VMs / flows

- Input queues for gateways linking disjoint lossless (QCN CN) L2 domains
  - Tunnel through lossy L3 network
  - Build a large lossless CEE datacenter by connecting smaller L2 domains … while enforcing QCN-compliant congestion mgmt in the gateways

Hotspot Congestion in Lossless Networks → Saturation Trees



CNA: Converged Network Adapter
SE: Switch Element

Courtesy Mitch Gusat, IBM Zurich Research