Cross-Layer Flow and Congestion Control for Datacenter Networks

Andreea Simona Anghel, Robert Birke,
Daniel Crisan and Mitch Gusat

IBM Research GmbH, Zürich Research Laboratory

Outline

- Motivation
  - CEE impact on socket applications
- Evaluation methods
  - Simulation vs. Hardware
  - Focus inside rack and node
  - 3 workload classes: Hotspot, MapReduce, HPC
- Results
  - Highlights discussion
- Conclusions
  - Lessons learned

Motivation: 3 Overlapping Loops

- 802 DCB / CEE features on L2
  - Losslessness: PFC
  - Congestion management in h/w: QCN

- Most DC/Cloud apps are socket-based
  - Bulk of DC communication: **TCP**
  - Some UDP (FB, YouTube) + VN tunneling

Q1) How does TCP perform over CEE – tweaks … ?

Q2) Is PFC beneficial ?

Q3) Is QCN beneficial ?

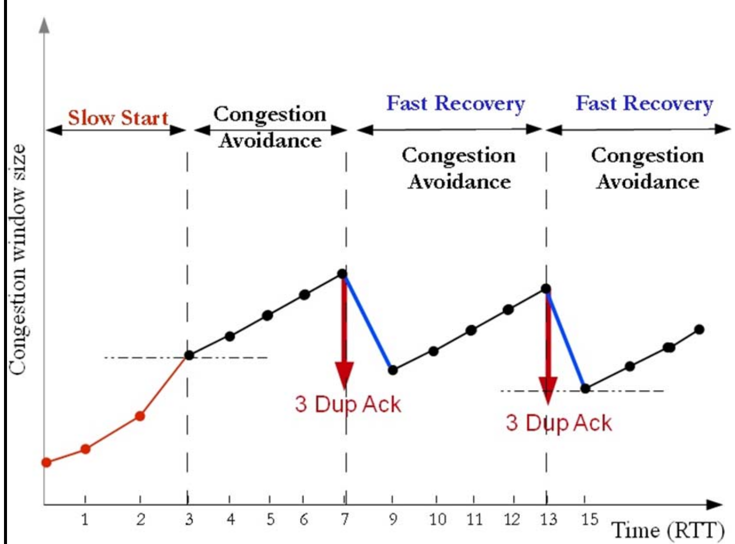
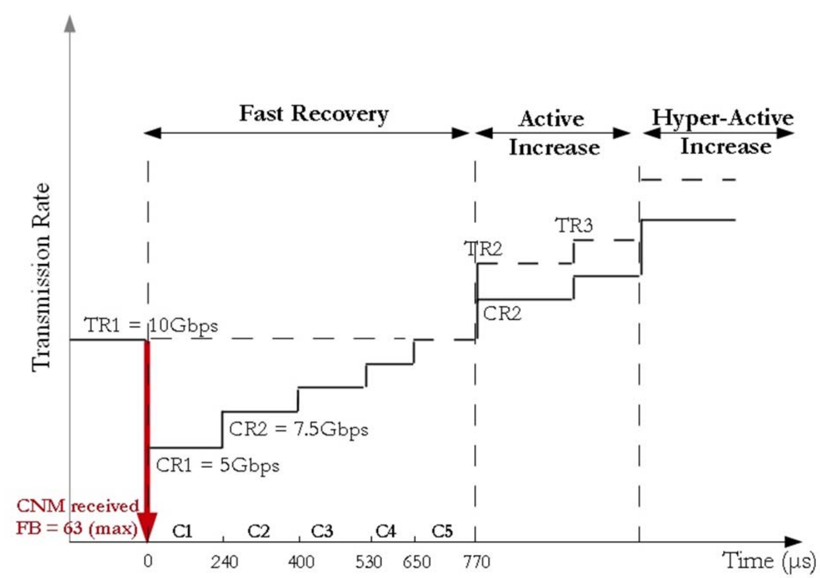
Stiff and Soft Controls: Exploration Space

- L4: TCP Congestion Control (x3)
  - NewReno
  - Vegas
  - Cubic
- L2 stiff: Link-level flow-control (x2)
  - PFC – i.e. lossless
  - Without PFC – i.e. lossy
- L2/3 h/w Congestion Mgnt. 'softies' (x4)
  - None, aka "Base"
  - QCN (L2) with Q_{eq} = 20K and 66K
  - RED - ECN (L3)
- Combinations: 3 x 2 x 4= 24 sim runs/result

Congestion Detection: L4 vs. L2

	L4 TCP (Reno)	L2 QCN
Detection Mechanism	1. @ destination (DupAck) 2. @ congestion point (AQM/ECN) 3. @ source (RTO)	@ congestion point (QCN sampler)
Feedback Type	1. Duplicate ACK (loss) 2. ECN/RED single-bit 3. Retransmission Timeout (latency)	Multibit: position, velocity
Burst Tolerance	Built-in	Low: instantaneous measure (depends on Q_{eq} setpoint)
Timescale	100s of *ms* (RTT dependent)	10s to 100s of μs

Congestion Control: L4 vs. L2

	L4 TCP (Reno)	L2 QCN
Principle of Operation	Window Controller @ SRC	Rate-based Controller @ SRC Finite State Machine : Cubic-like method
Increase & Decrease Control Law	Additive Increase Multiplicative Decrease (AIMD) 	Fb-proportional Decrease / Fast Recovery + Active Increase + Hyper Active Increase 

Evaluation Method 1: Simulation Environment (1)

- Workloads and Applications
  (1) Hotspot synthetic traffic: 802 DCB
    - Many sources to one destination, aka Input Generated (IG) congestion from 802 DCB
      - Collectives-like pathological hotspot

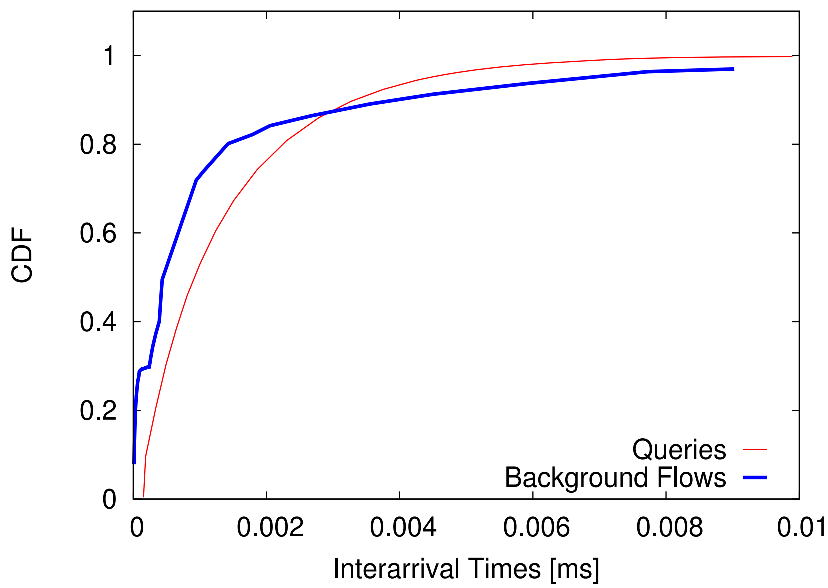
  (2) Commercial applications
    - Foreground: socket-based Partition/Aggregate
    - Background cross traffic: TCP or UDP flows

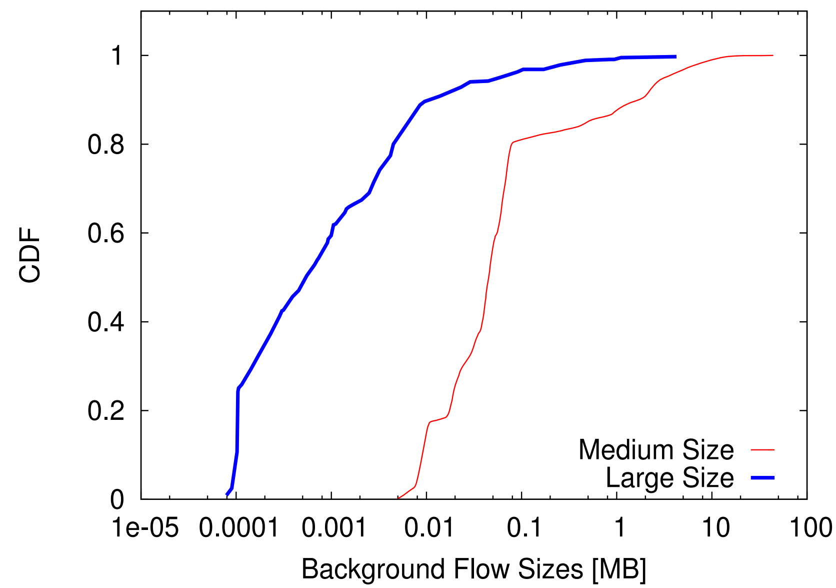
  (3) Scientific: 5 NAS + 4 other HPC benchmarks
    - Collected by BSC on Mare Nostrum

Evaluation Method 1: Simulation Environment (2)

(2) Commercial workload: MapReduce-like

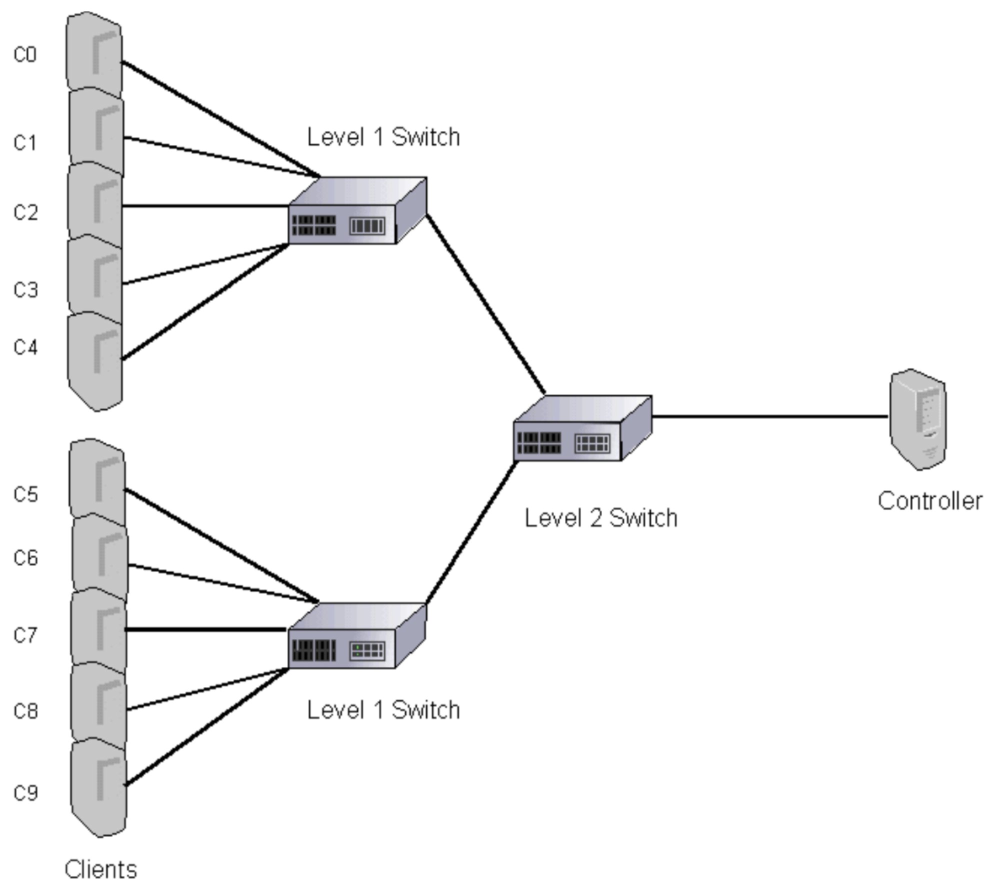
- Partition/Aggregate queries (see next)
- Background flows: Medium/Large size





Evaluation Method 2:Hardware Testbed (1)

- Hardware Topology
  - 10 hosts, 1 controller and 3 switches (802.3x PAUSE)
  - Fast Ethernet network



Evaluation Method 2:Hardware Testbed (2)

- L4: New Reno, Vegas and Cubic (x3)
- L2: 802.3x PAUSE (enabled/disabled) (x2)
- Without L2/3 CM

- Workloads and Applications
  - Commercial applications without background traffic
    - Socket-based Partition/Aggregate

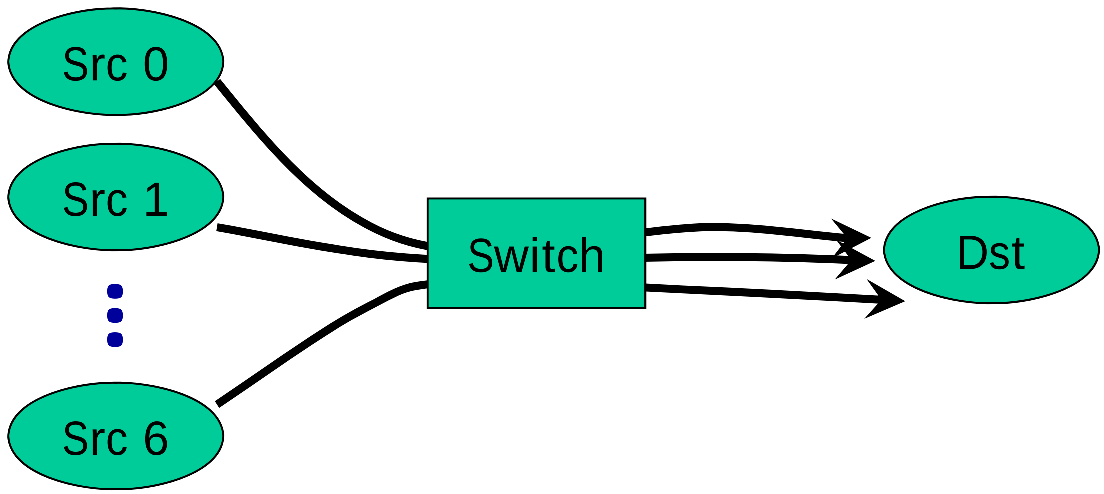
Outline

- Motivation
- Evaluation methods
- Results
- Conclusions

TCP Tweaks for DCN

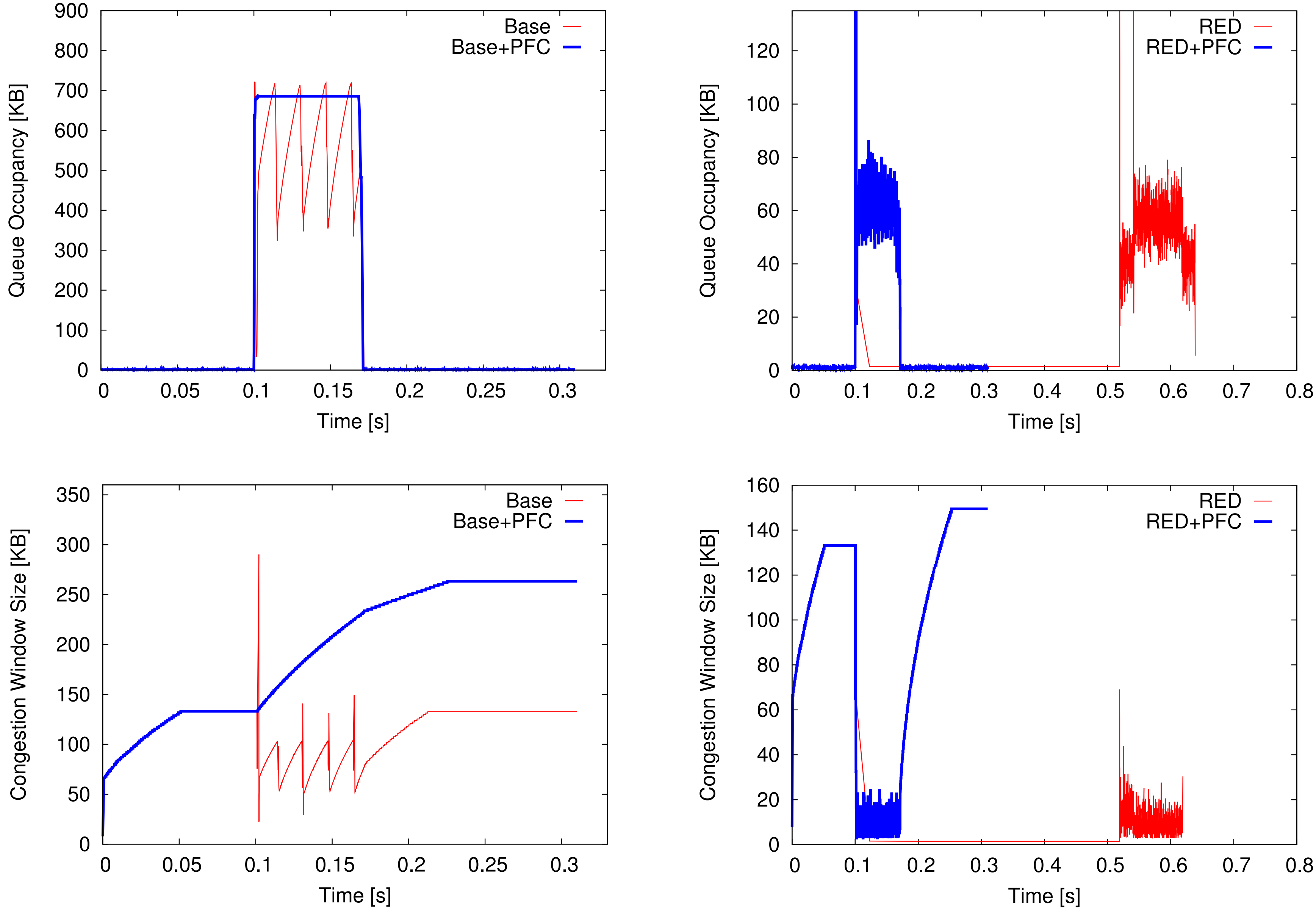
- Finer jiffy:
  - In datacenter networks empty RTT « kernel timer quanta
  - **Simulation**: Timer from 1ms to 1us
  - **Hardware**: Timer from 250HZ to 1000 HZ
- RTO = key to DC-TCP performance
  - Default to 3s → we set it: **Simulation** 10ms and **Hardware** 30ms
  - **Simulation**: we set RTOmin = 2ms
  - Variance of stack defaults to 200ms
    - **Simulation**: We set it to 20ms
- Jacobson's RTT estimator is critical(ly broken in DCN)
  - RTT variance ~(3-5) orders of magnitude
    - It's queuing, not flight, dominated
  - Processing time inside the kernel (10s of us) can be (MUCH) larger than DC network RTTs (0.5 – 10us empty)

Congestive Synthetic Traffic (1)

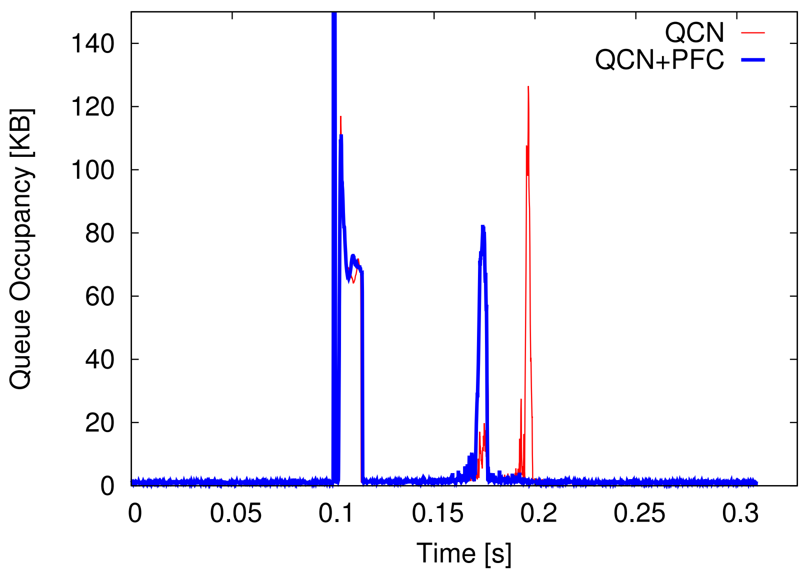


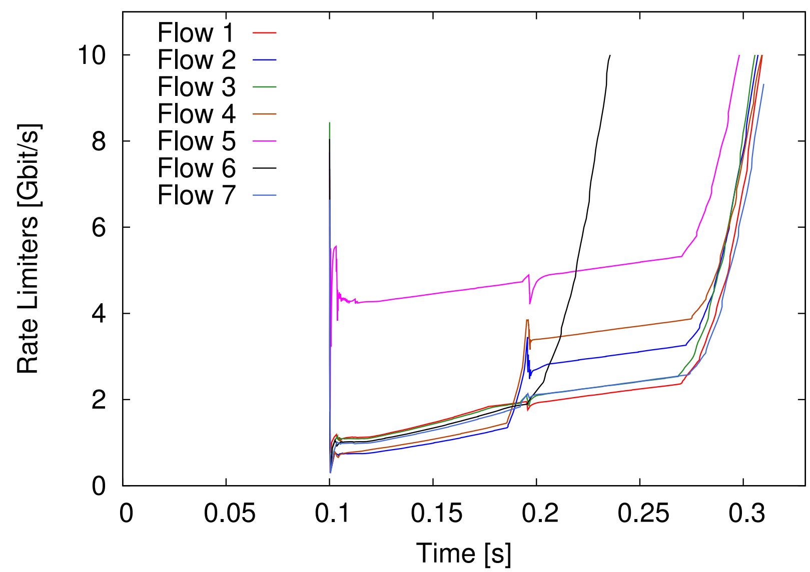
- TCP incast
  - 7 sources -> 1 destination
  - From $\mathbf{t_0}$**=0ms** to $\mathbf{t_1}$**=100ms** admissible traffic
  - Followed by a **10ms** 4x overload of the destination
- Tested in the simulation environment only

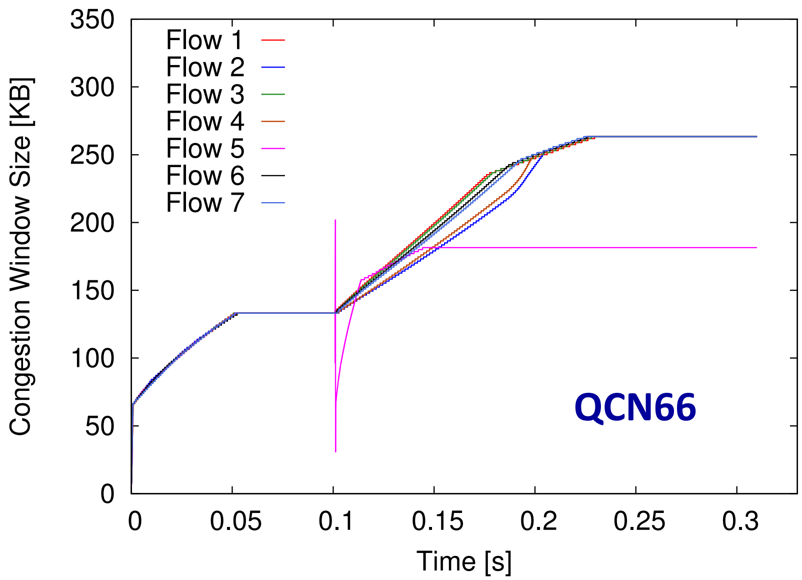
Congestive Synthetic Traffic (2)

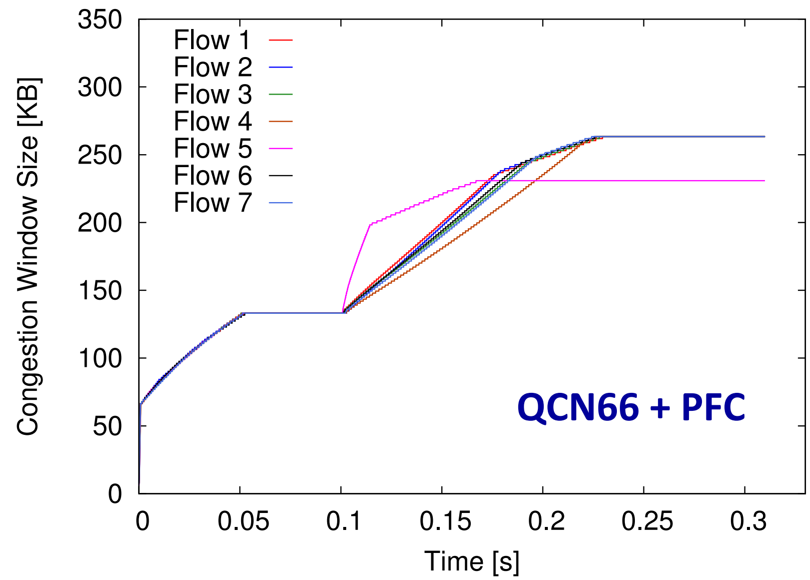


Congestive Synthetic Traffic (3): QCN

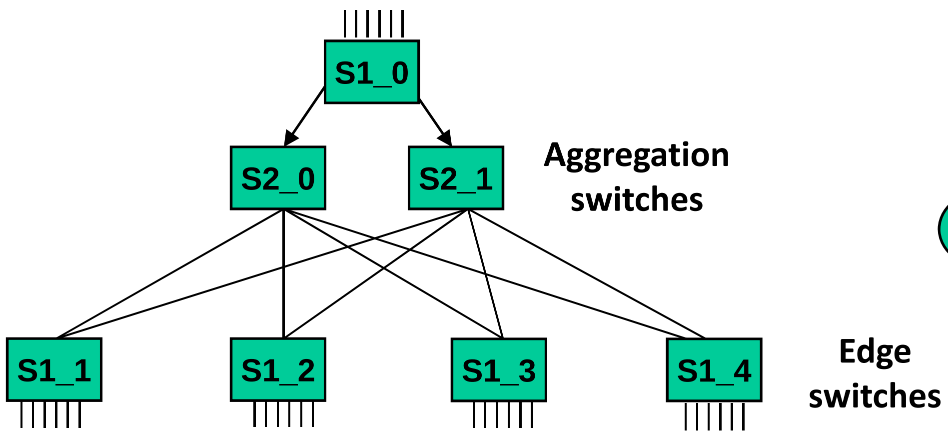


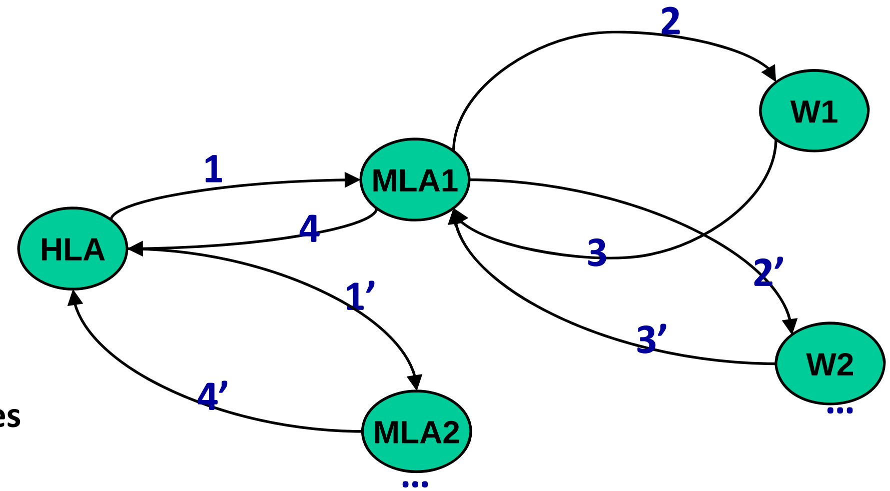






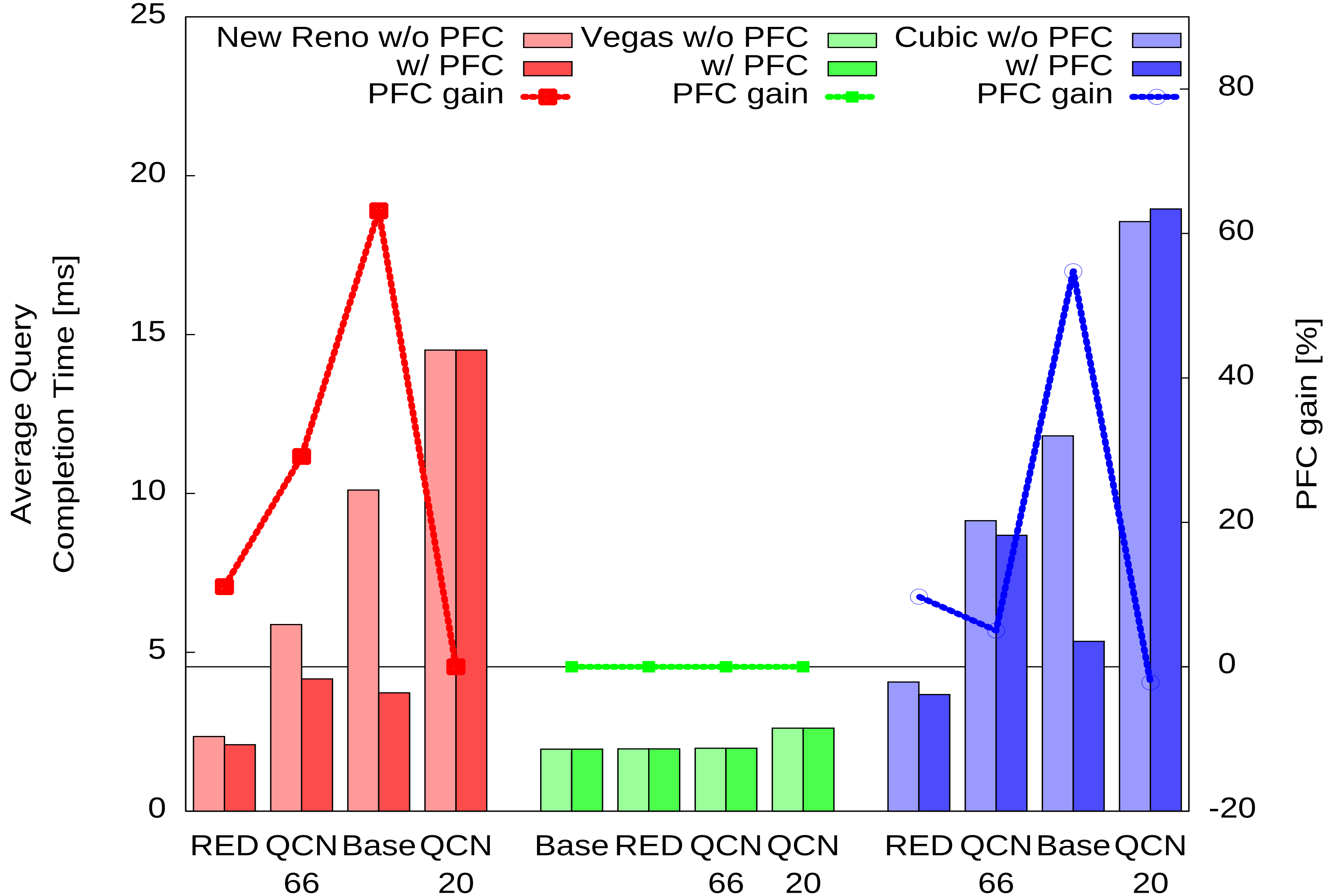
Commercial Workloads: Incast Culprits



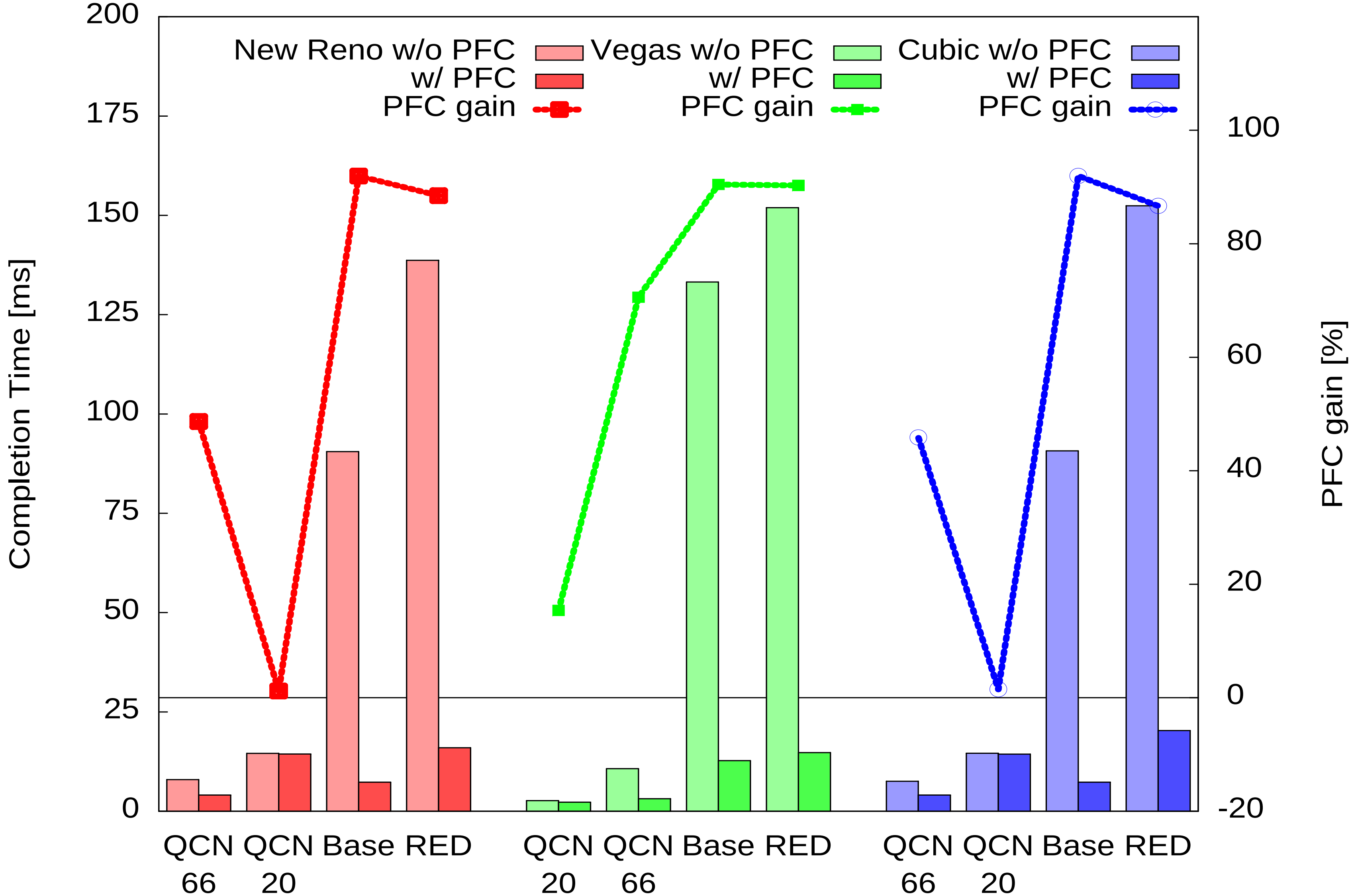


- **Queries** – Partition /Aggregate, Scatter/Gather, MapReduce
  - Nodes in chassis 0 are High Level Aggregators (HLA)
  - Each HLA chooses a random Mid Level Aggregator (MLA) in chassis 1 to 4 and distributes the query to them
  - Each MLA distributes the query to all the other servers in it's own chassis that act as Workers (W)
  - Edges 1,2 are Requests
  - Edges 3,4 are Replies (answers)
  - Replies and requests are sent and received in parallel
- **Background traffic** – each server in chassis 1 to 4 chooses a random destination and sends it a single flow

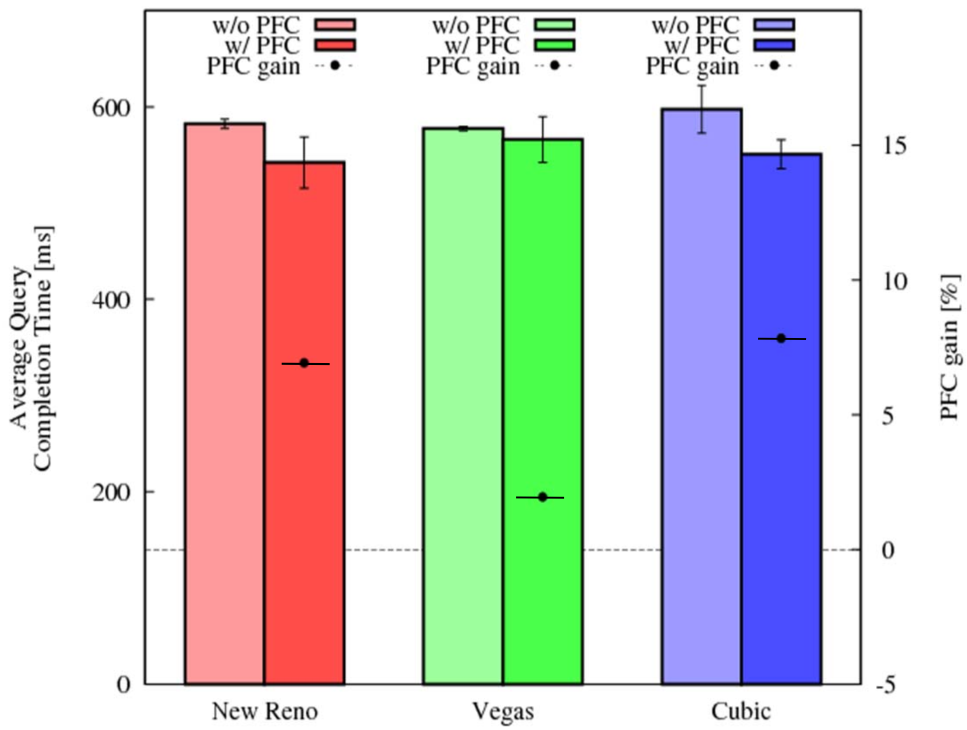
Simulation: P/A Applications + TCP background



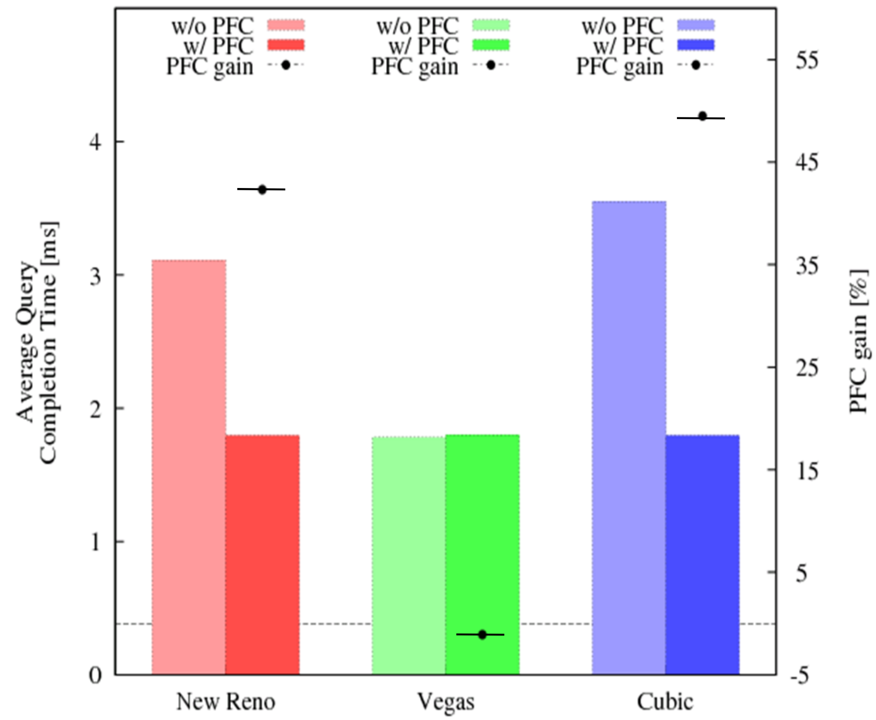
Simulation:
P/A Applications + UDP background



Hardware Platform Validation (1)
P/A workload w/o background



Hardware Results



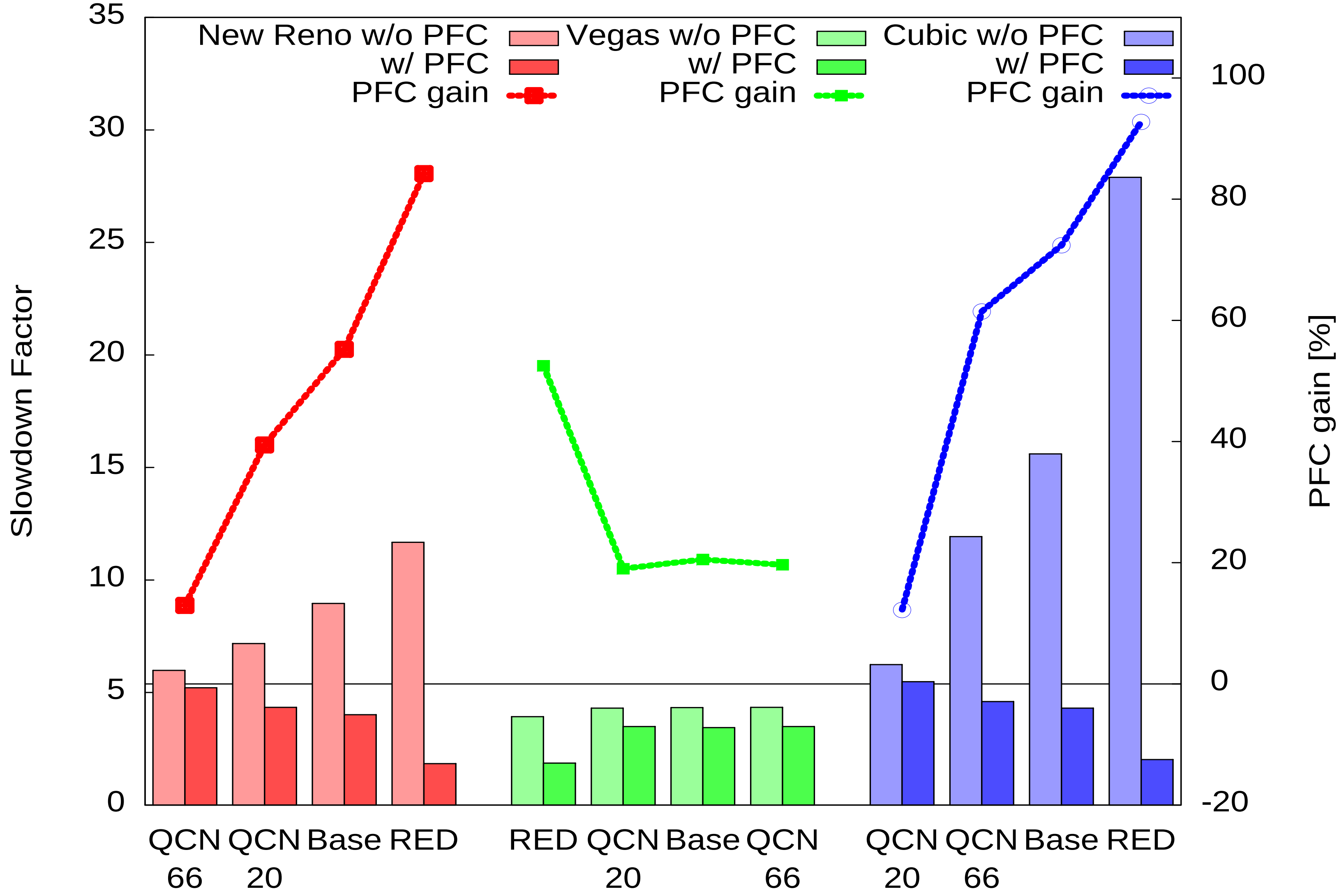
Simulation Results

Hardware Platform Validation (2)

- PFC is always beneficial...

- Not as spectacular as in the 10G DCB
- Why only 7-8% improvement?
  - 100x slower network (Fast Ethernet vs. 10Gbps)
  - No CEE support
  - Only 10 end nodes (10 vs. 80)
  - Simple network topology (3 switches)
  - No access to the 802.3x PAUSE thresholds
  - Consistent w/ most recent other h/w publications

Simulation: Scientific Applications – HPC Workload

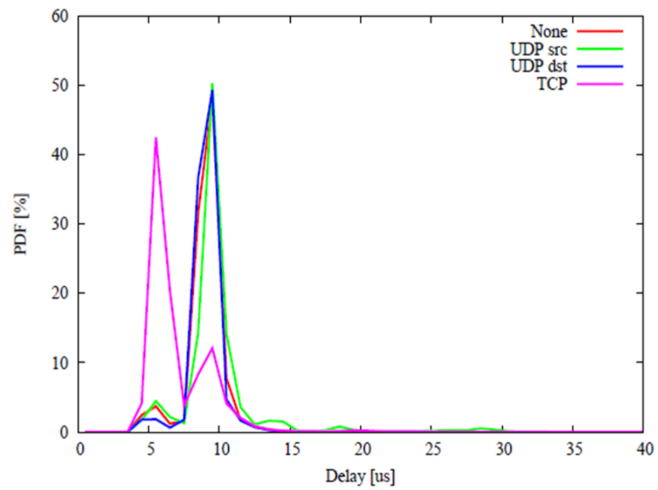


Conclusions

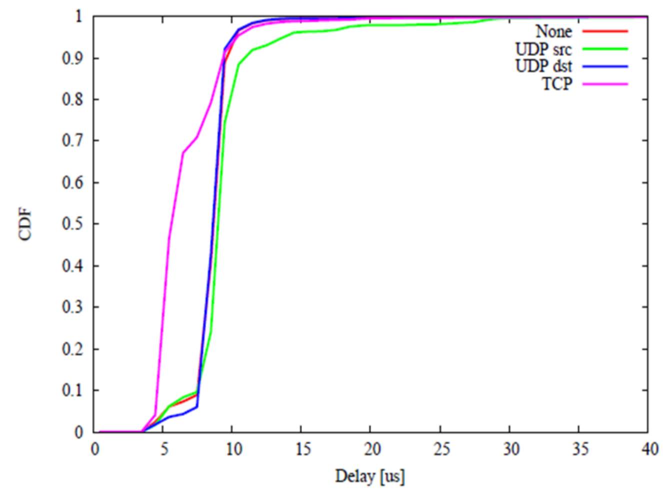
- How does TCP perform over CEE ?
  - TCP Vegas is the best
  - Cubic not well suited
- Is PFC beneficial ?
  - **YES**: Loss is a latency singularity!
- Is QCN beneficial ?
  - Depends
    - if TCP competes against UDP => YES
    - on the proper tuning per application => Not practical
    - This actually may mean **NO**
  - Ditto for RED
- To fix: TCP's RTO calculations are broken for DCN

Backup slides

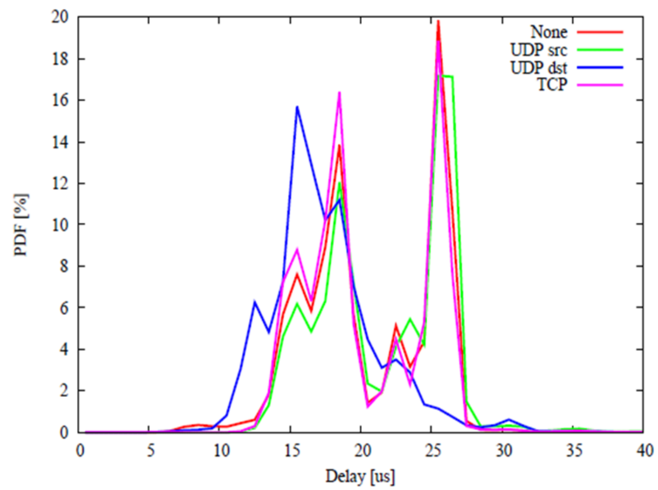
OS Stack delays



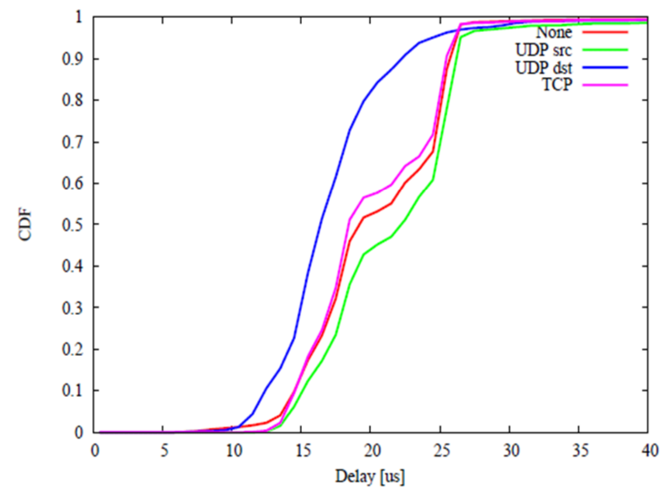
(a) TX stack delay PDF.



(b) TX stack delay CDF.



(c) RX stack delay PDF.



(d) RX stack delay CDF.

Simulation Parameters

Parameter	Value	Unit	Parameter	Value	Unit
TCP					
buffer size	128	KB	TX delay	9.5	μs
max buffer size	256	KB	RX delay	24	μs
default RTO	10	ms	timer quanta	1	μs
min RTO	2	ms	reassembly queue	200	seg.
RTO variance	20	ms			
ECN-RED					
min thresh.	25.6	KB	W_q	0.002	
max thresh.	76.8	KB	P_{max}	0.02	
QCN					
Q_{eq}	20 or 66	KB	fast recovery thresh.	5	
W_d	2		min. rate	100	Kb/s
G_d	0.5		active incr.	5	Mb/s
CM timer	15	ms	hyperactive incr.	50	Mb/s
sample interval	150	KB	min decr. factor	0.5	
byte count limit	150	KB	extra fast recovery	enabled	
PFC					
min thresh.	80	KB	max thresh.	97	KB
Network hardware					
link speed	10	Gb/s	adapter delay	500	ns
frame size	1500	B	switch buffer size/port	100	KB
adapter buffer size	512	KB	switch delay	100	ns