Smart-Pause: Using Precise Packet Delays to Pause at the Edge of the Network

June 7, 2019

Shiyu Liu, Balaji Prabhakar, Mendel Rosenblum

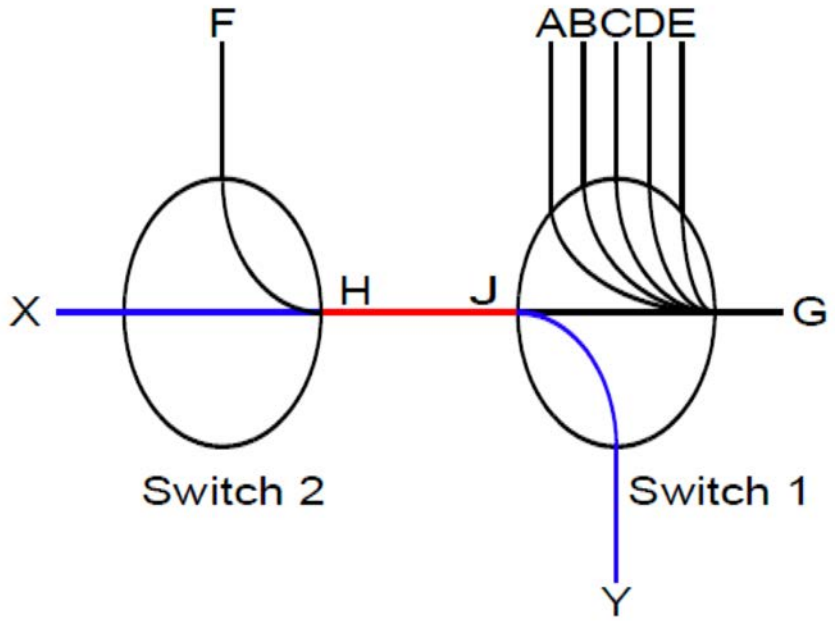
Overview

- What is Smart-Pause
- Evaluation of Smart-Pause
  - Baseline: DCQCN, TIMELY
  - Varying buffer sizes, load, incast fan-in

Pitfalls of Per-priority Pause in Switches

- The IEEE 802.1qbb standard for priority flow control (PFC)
  - Enables a switch to Pause transmissions on an Ethernet priority from upstream switches when the given switch's buffer occupancy in that priority exceeds a set threshold
  - When room frees up in the buffer, transmissions are resumed
  - The goal is to prevent packet drops and create "lossless Ethernet"

- In practice PFC can cause "congestion spreading" and severely reduce throughput
  - Congestion spreading affects "innocent flows" at upstream switches not passing through the bottleneck link
  - This, in turn, interferes with congestion control algorithms and severely degrades their performance



Source: https://community.mellanox.com/s/article/understanding-rocev2-congestion-management

- References

  - IEEE 802.1Qbb-2011 "Priority-based Flow Control"
  - C. Guo, et al. "RDMA over commodity ethernet at scale". SIGCOMM 2016
  - S. Hu, et al. "Deadlocks in Datacenter Networks: Why Do They Form, and How to Avoid Them". HotNets 2016
  - R. Mittal, et al. "Revisiting Network Support for RDMA". SIGCOMM 2018

Is Pause Useful?

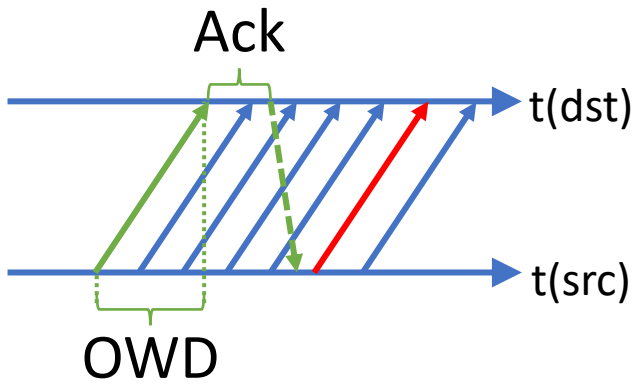
- Pause is the quickest way to react to congestion
  - It's mainly intended to prevent or minimize drops
  - It acts on each packet, as opposed to congestion control algorithms which act on an RTT's worth of packets
  - Pause acts essentially independently of congestion control and can work with any congestion control algorithm

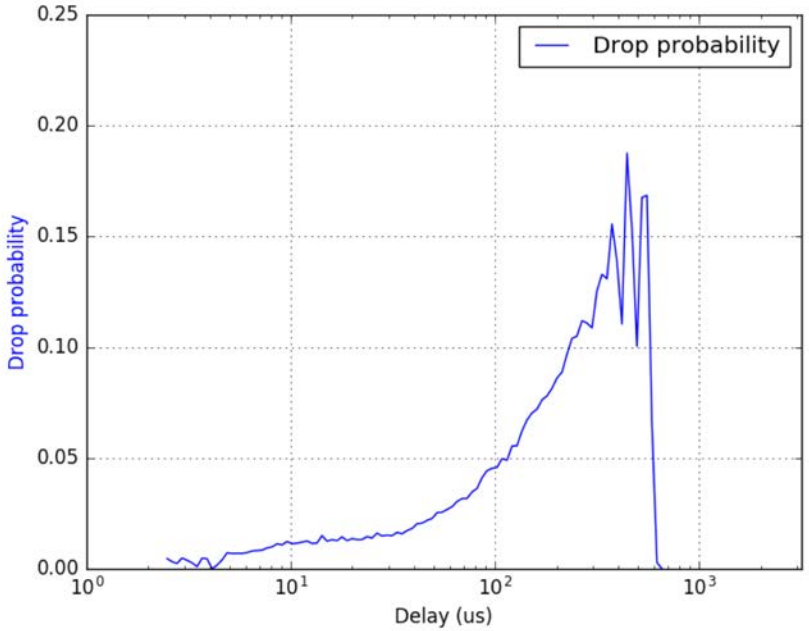
- Ultimately, whether and where to Pause is determined by
  - The benefits of Pause (and the absence of any disbenefits)
  - What information is available to exert Pause

  → This talk is a preliminary exploration of these questions

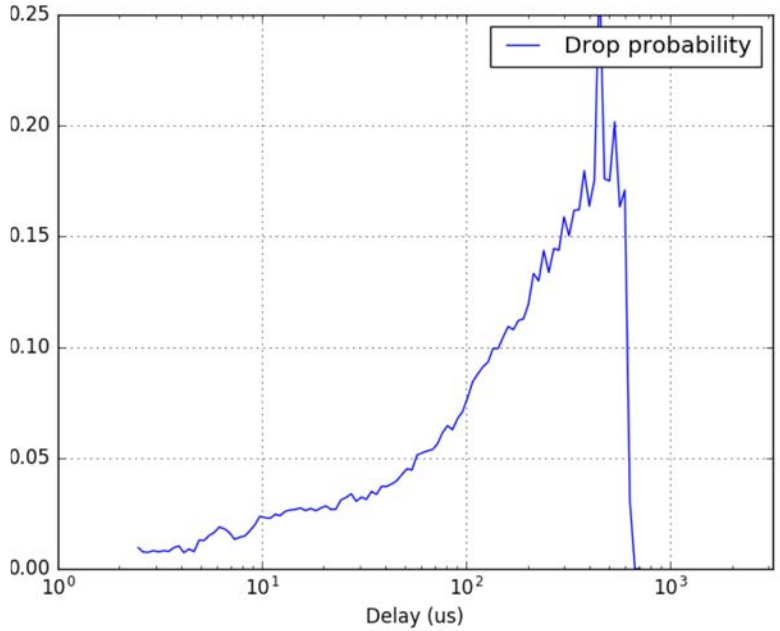
Our approach

- Move Pause to the edge of the network; implement it in Smart NICs
  - On a per-packet basis using on a novel measure of “path- and load-dependent” congestion
- The One-way Delay (OWD) vs drop probability curve
  - X-axis: the OWD of the last packet for which sender has received an ack
  - Y-axis: the probability that the next packet to be transmitted from that flow will be dropped





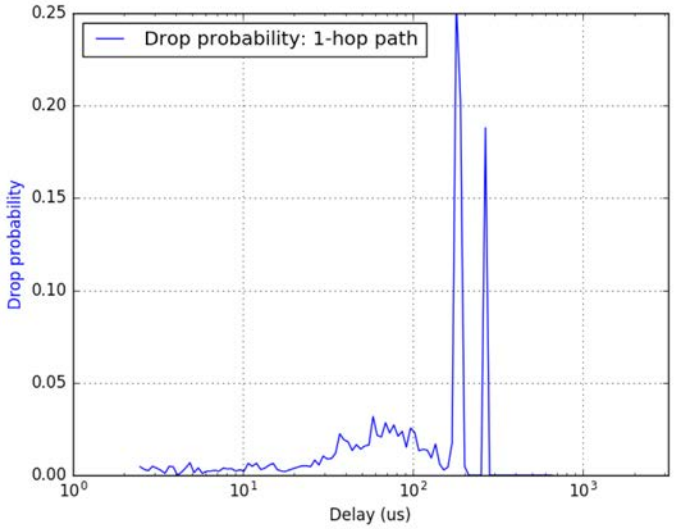
60% load, fanout = 1, 2MB buffer



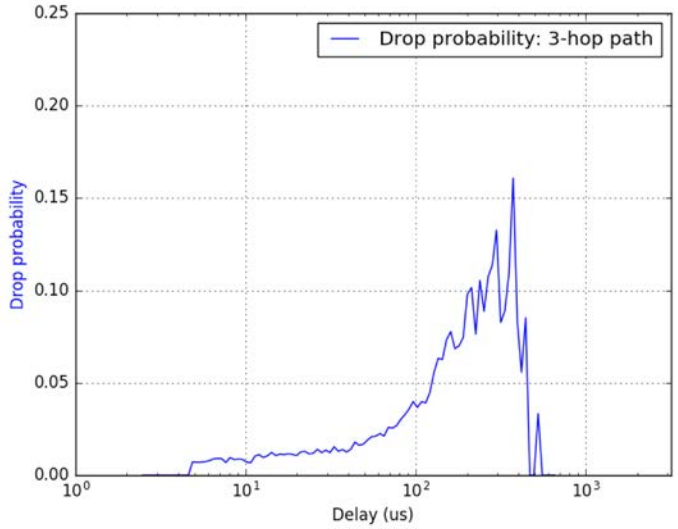
80% load, fanout = 1, 2MB buffer

Our approach

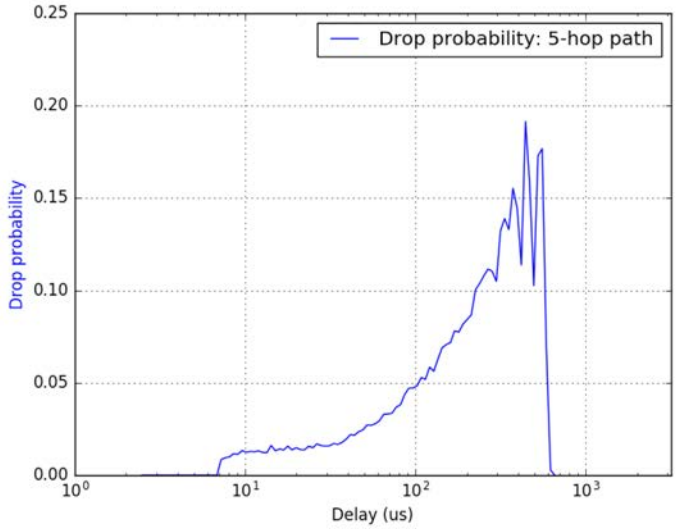
- Move Pause to the edge of the network; implement it in Smart NICs
  - On a per-packet basis using on a novel measure of “path- and load-dependent” congestion
- The One-way Delay (OWD) vs drop probability curve
  - X-axis: the OWD of the last packet for which sender has received an ack
  - Y-axis: the probability that the next packet to be transmitted from that flow will be dropped



1 hop



3 hops

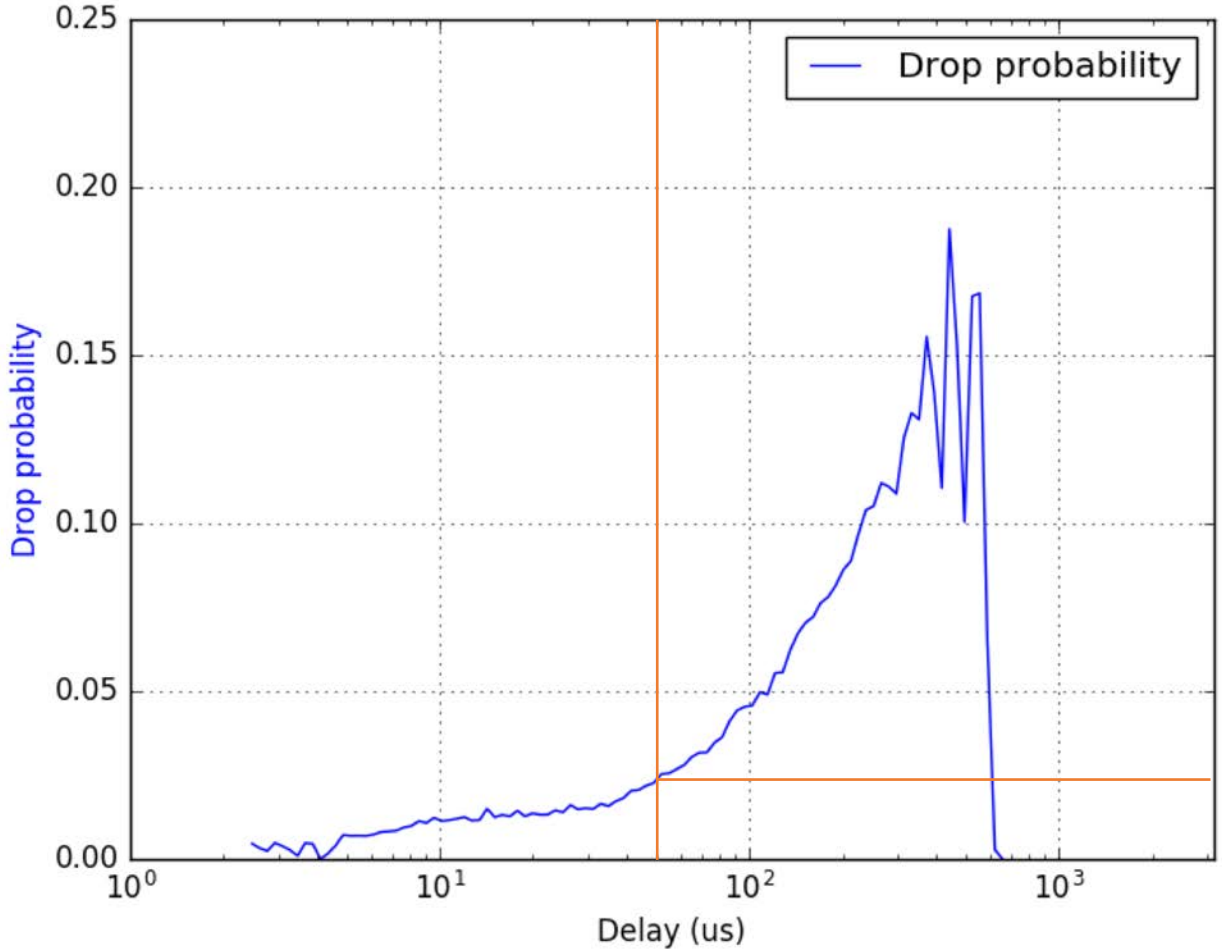


5 hops

60% load, fanout = 1, 2MB buffer

The Smart-Pause Rule

- When the OWD is greater than T units of time, Pause next packet from the flow at the edge for P usecs
  - Fix P:
    - T = 50 usecs
    - P = 20 usecs
  - Dynamic P:
    - T = 12,14,16 usecs for 1,3,5-hop flows
    - P = OWD – T
- Parameters could be dependent on paths and loads.



Overview

- What is smart-pause
- Evaluation of smart-pause
  - Baseline: DCQCN, TIMELY
  - Varying buffer sizes, load, incast fan-in

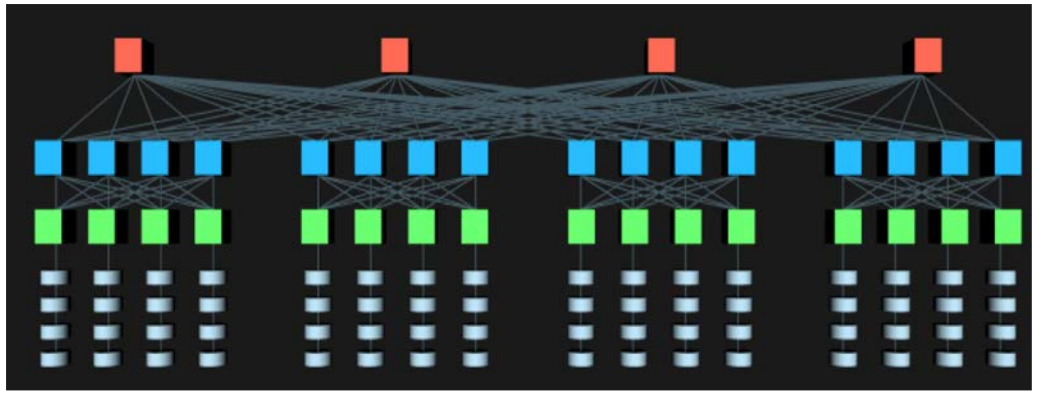
Experiment setup

- NS3 simulation implementing RoCE with DCQCN & TIMELY
- 40G, 3-stage Clos network, 1us link delay
- Switches have 2MB shared-buffer (256KB per port in avg)
- ECMP load balancing
- Traffic pattern:
  - Each end host generates new flows with Poisson arrival
  - Flow destination is random
  - Flow size is heavy-tailed distribution derived from [1][2]
    - 50% flows are single pkt msgs, e.g. key-value stores.
    - 15% flows are 200KB-3MB, e.g. storage traffic.
- Baselines:
  - DCQCN + no PFC
  - DCQCN + PFC

  } vs. DCQCN + no PFC + smart-pause

  - TIMELY + no PFC
  - TIMELY + PFC

  } vs. TIMELY + no PFC + smart-pause

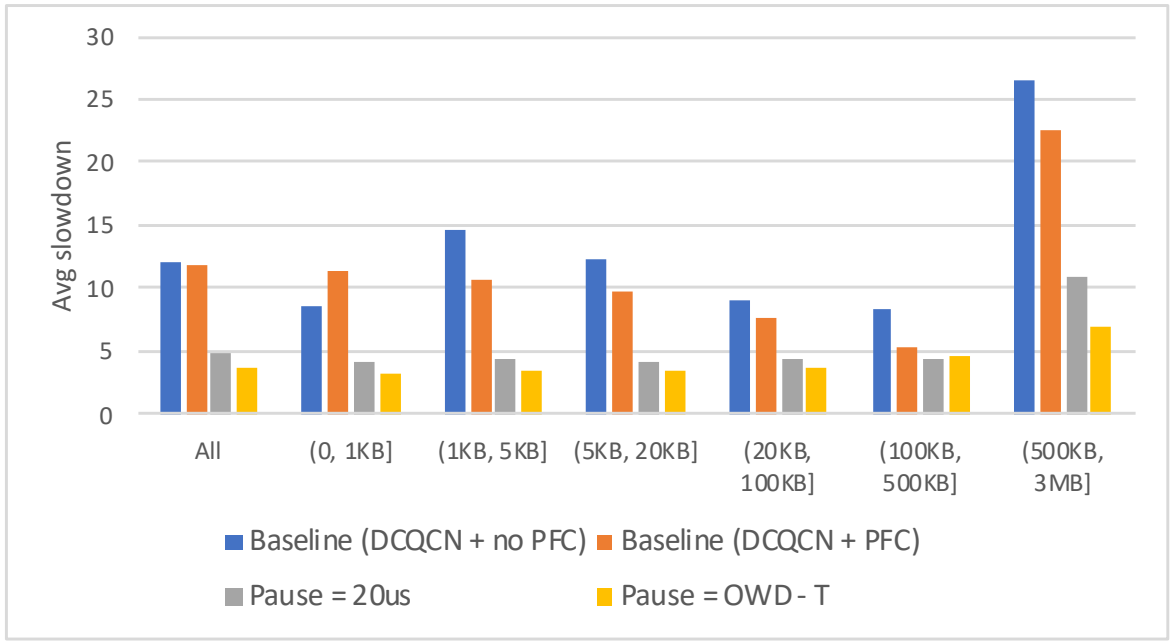


[1] *Revisiting Network Support for RDMA*, SIGCOMM'18
[2] *Network Traffic Characteristics of Data Centers in the Wild*, IMC'12

DCQCN, 60% load, fan-in = 1, 2MB buffer

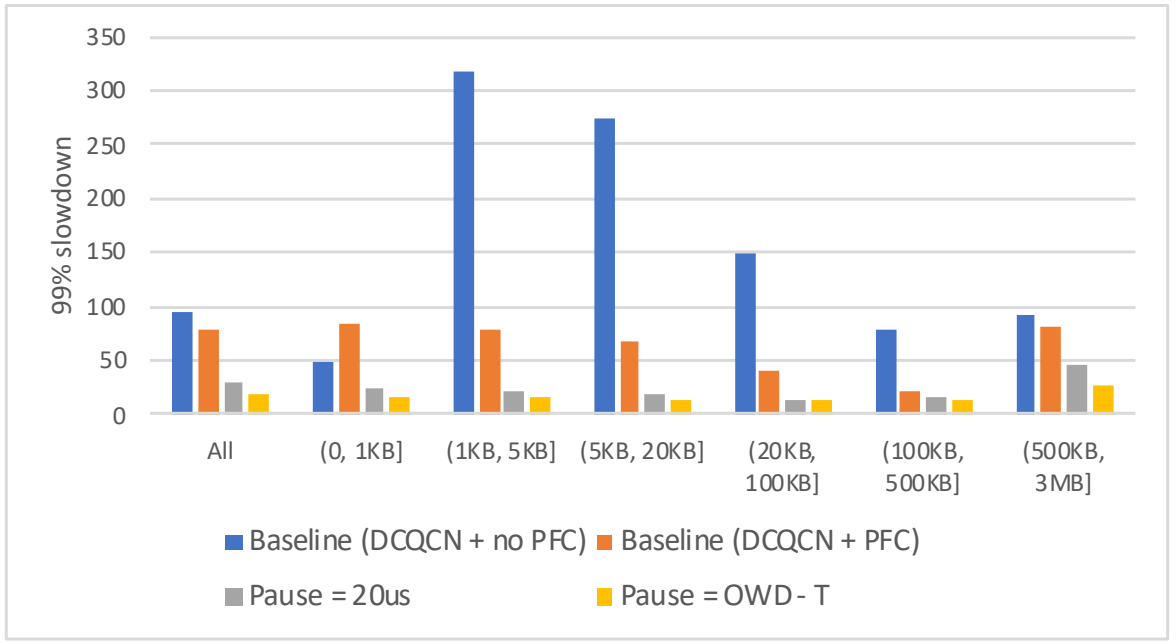
Average Slowdown

Improvement factor								
DCQCN + no PFC	Pause = 20us	2.5x	2.1x	3.4x	3.0x	2.1x	1.9x	2.4x
	Pause = OWD - T	3.2x	2.7x	4.4x	3.5x	2.4x	1.8x	3.8x
DCQCN + PFC	Pause = 20us	2.4x	2.7x	2.5x	2.4x	1.8x	1.2x	2.1x
	Pause = OWD - T	3.1x	3.5x	3.2x	2.8x	2.0x	1.1x	3.2x

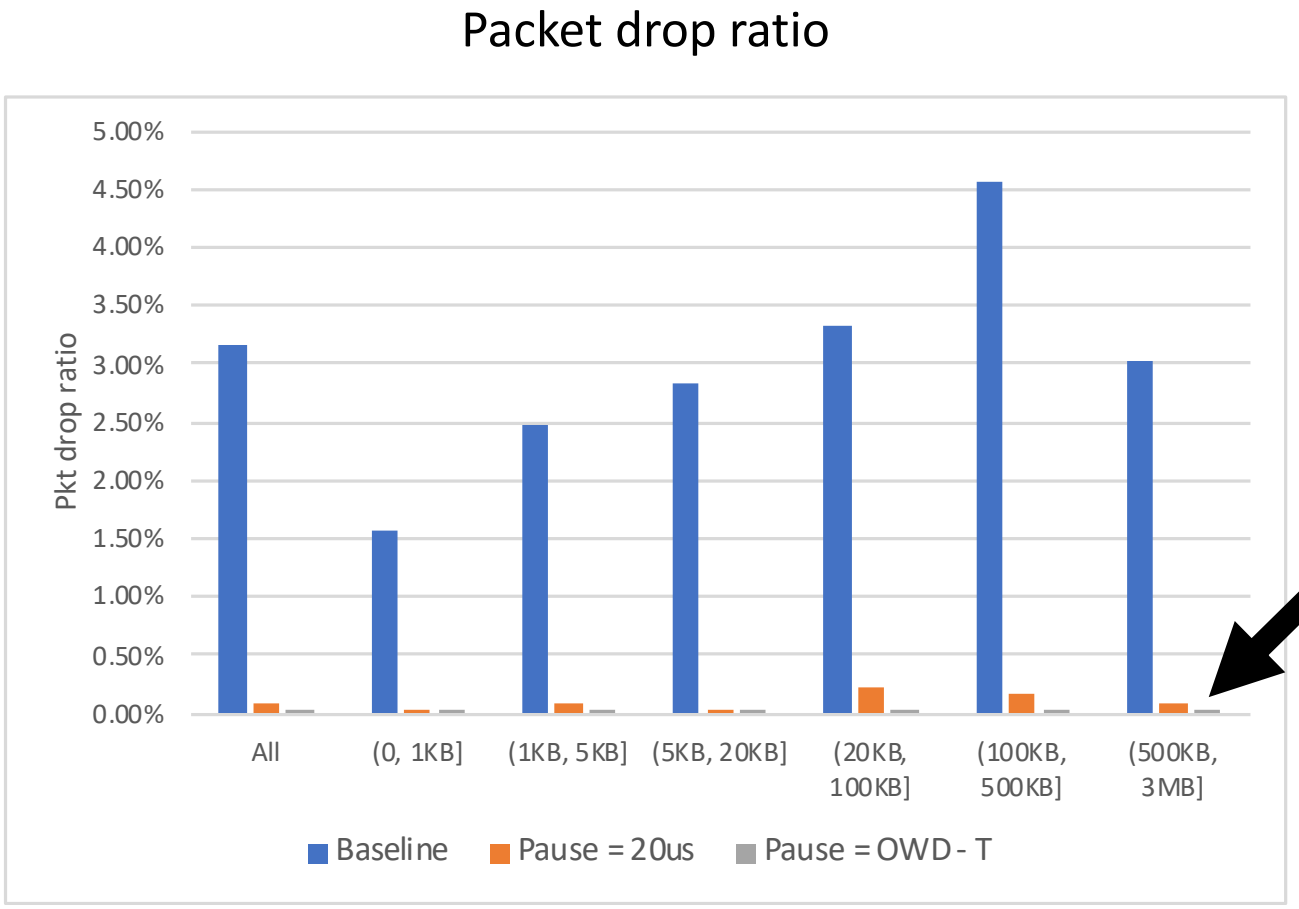


99% slowdown

Improvement factor								
DCQCN + no PFC	Pause = 20us	3.1x	2.2x	14.7x	14.9x	10.3x	5.3x	2.0x
	Pause = OWD - T	5.3x	3.1x	21.1x	19.6x	12.5x	5.7x	3.6x
DCQCN + PFC	Pause = 20us	2.6x	3.6x	3.6x	3.6x	2.8x	1.4x	1.8x
	Pause = OWD - T	4.5x	5.3x	5.2x	4.8x	3.4x	1.5x	3.2x



DCQCN, 60% load, fan-in = 1, 2MB buffer

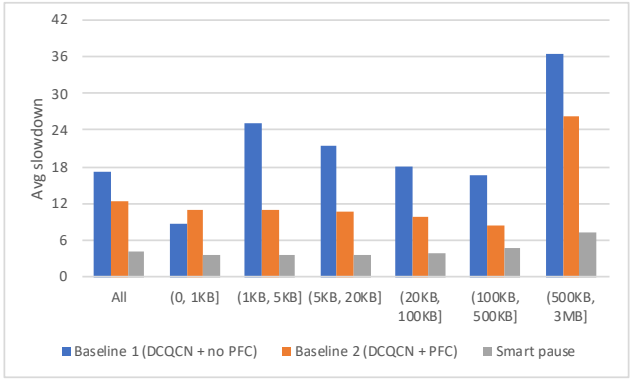


Pkt drop ratio goes to nearly 0

DCQCN (60% load, incast fan-in 1) + smart pause: various buffer sizes

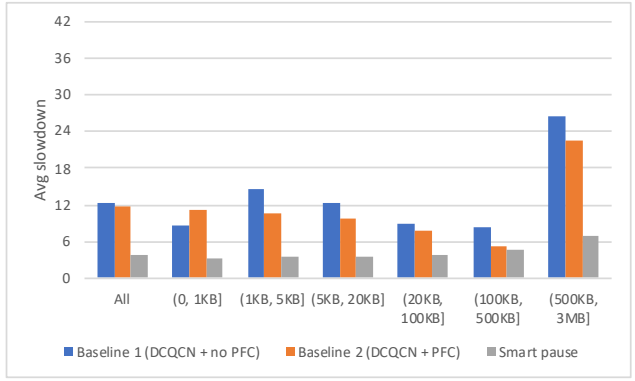
1MB buffer, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
4.2x	2.4x	6.9x	5.8x	4.7x	3.5x	5.1x
3.0x	3.0x	3.0x	2.9x	2.5x	1.8x	3.7x



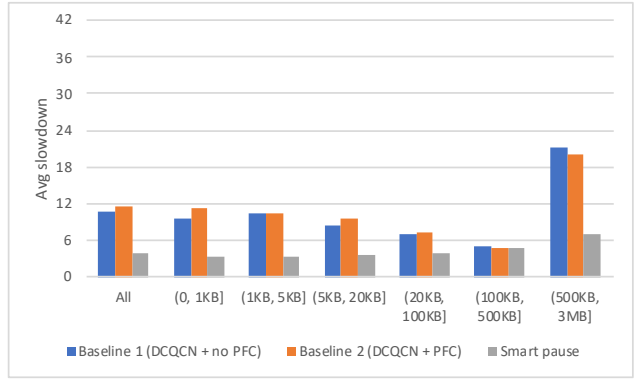
2MB buffer, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.2x	2.7x	4.4x	3.5x	2.4x	1.8x	3.8x
3.1x	3.5x	3.2x	2.8x	2.0x	1.1x	3.2x



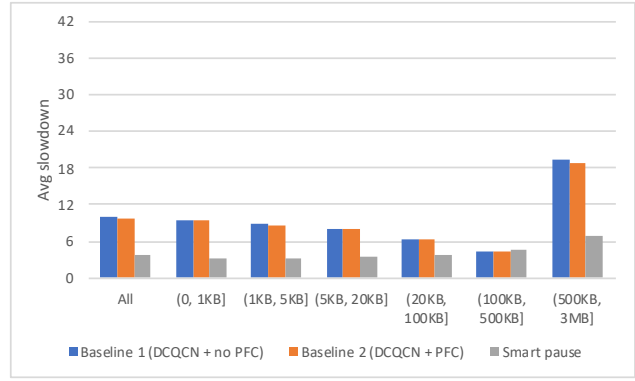
4MB buffer, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
2.8x	2.9x	3.1x	2.5x	1.8x	1.1x	3.1x
3.0x	3.5x	3.2x	2.8x	1.9x	1.0x	2.9x



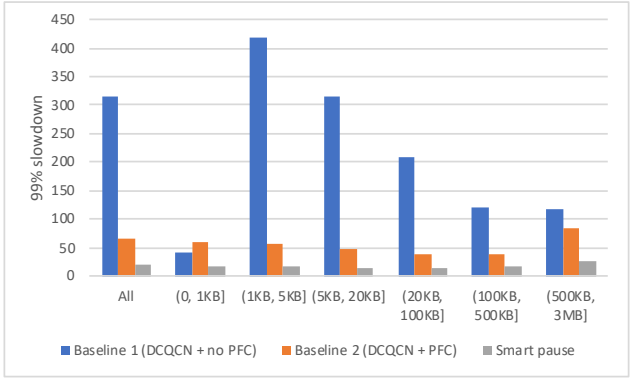
8MB buffer, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
2.6x	2.9x	2.7x	2.3x	1.7x	1.0x	2.8x
2.6x	2.9x	2.6x	2.3x	1.7x	1.0x	2.7x



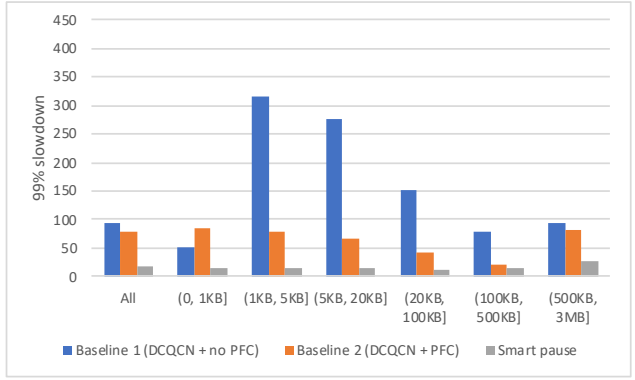
1MB buffer, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
17.1x	2.5x	27.0x	22.7x	16.9x	7.7x	4.4x
3.6x	3.7x	3.6x	3.5x	3.1x	2.5x	3.1x



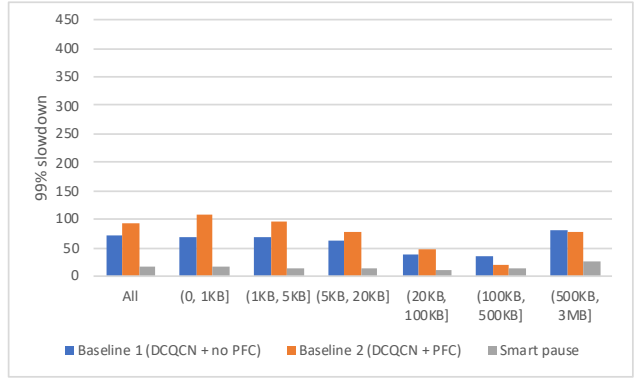
2MB buffer, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
5.3x	3.1x	21.1x	19.6x	12.5x	5.7x	3.6x
4.5x	5.3x	5.2x	4.8x	3.4x	1.5x	3.2x



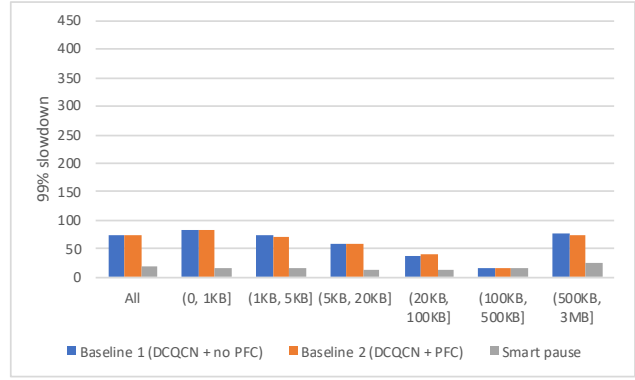
4MB buffer, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
4.0x	4.2x	4.6x	4.5x	3.4x	2.4x	3.2x
5.2x	6.7x	6.3x	5.6x	4.2x	1.3x	3.1x



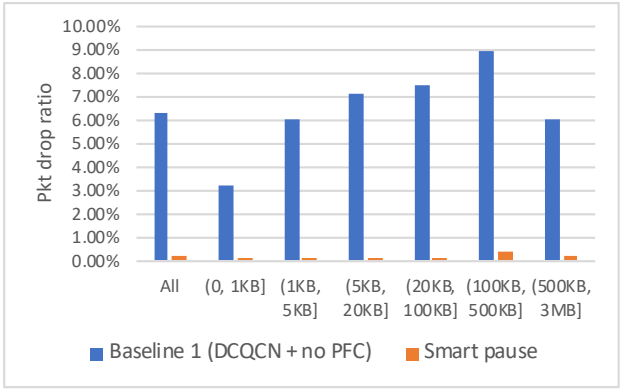
8MB buffer, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
4.2x	5.0x	4.8x	4.2x	3.3x	1.1x	3.0x
4.2x	5.2x	4.7x	4.2x	3.5x	1.0x	2.9x

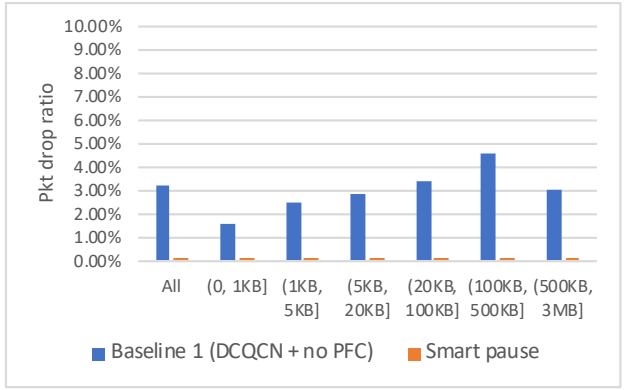


DCQCN (60% load, incast fan-in 1) + smart pause: various buffer sizes

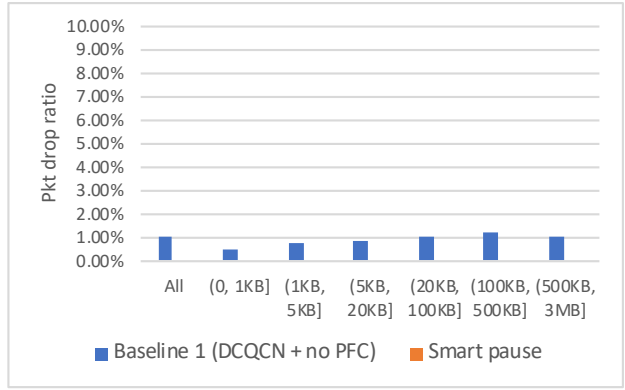
1MB buffer
% of packets dropped



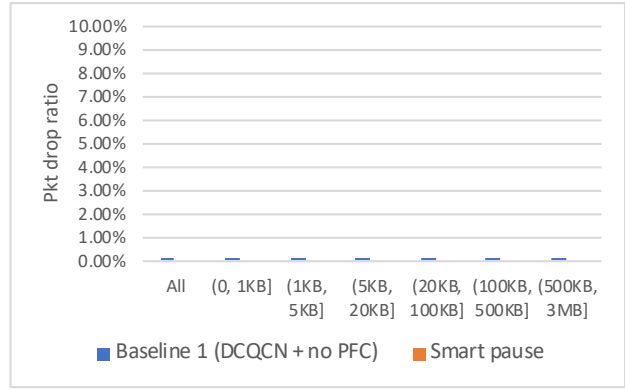
2MB buffer
% of packets dropped



4MB buffer
% of packets dropped



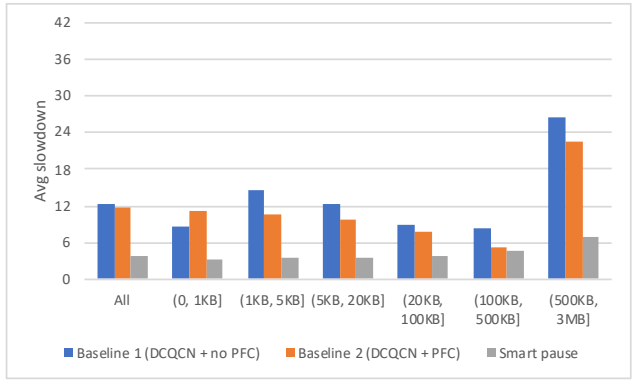
8MB buffer
% of packets dropped



DCQCN (2MB buffer, incast fan-in 1) + smart pause: various loads

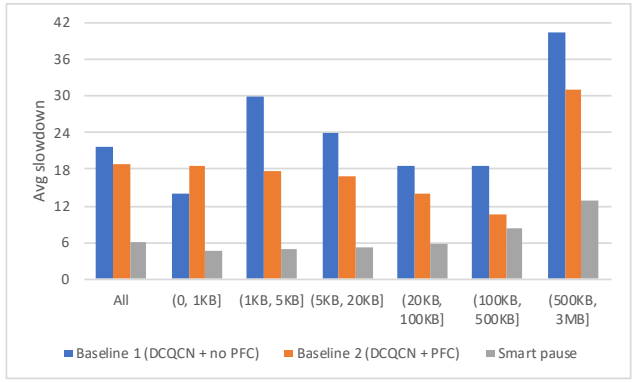
60% load, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.2x	2.7x	4.4x	3.5x	2.4x	1.8x	3.8x
3.1x	3.5x	3.2x	2.8x	2.0x	1.1x	3.2x



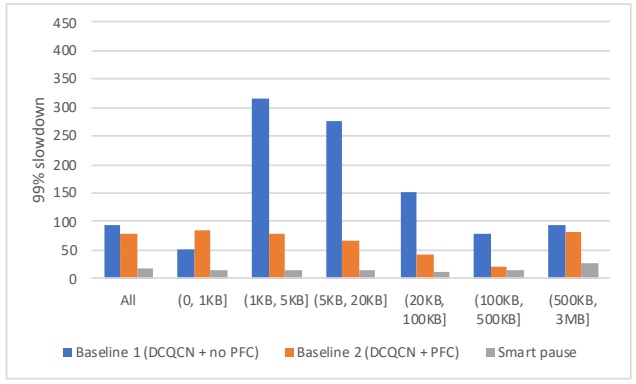
80% load, avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.6x	2.9x	6.0x	4.6x	3.3x	2.2x	3.2x
3.1x	3.9x	3.5x	3.3x	2.4x	1.2x	2.4x



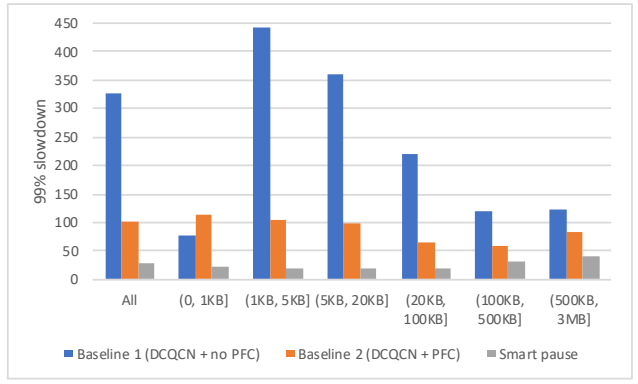
60% load, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
5.3x	3.1x	21.1x	19.6x	12.5x	5.7x	3.6x
4.5x	5.3x	5.2x	4.8x	3.4x	1.5x	3.2x



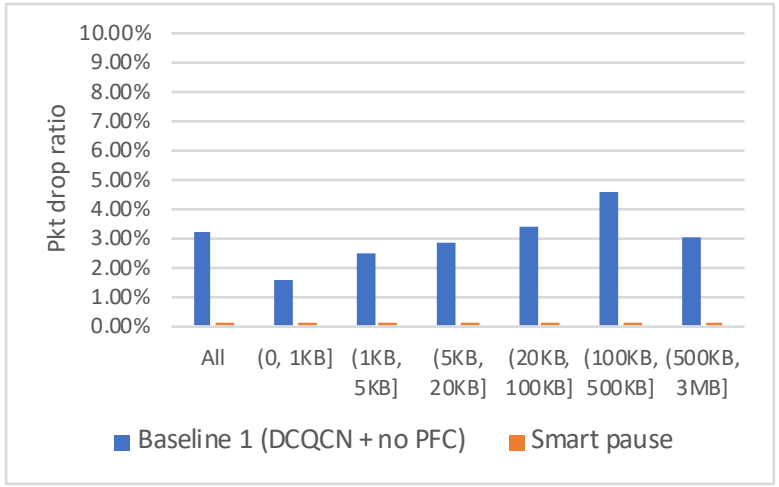
80% load, 99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
11.1x	3.6x	22.5x	20.0x	11.6x	3.9x	2.9x
3.5x	5.3x	5.3x	5.5x	3.4x	1.9x	2.0x

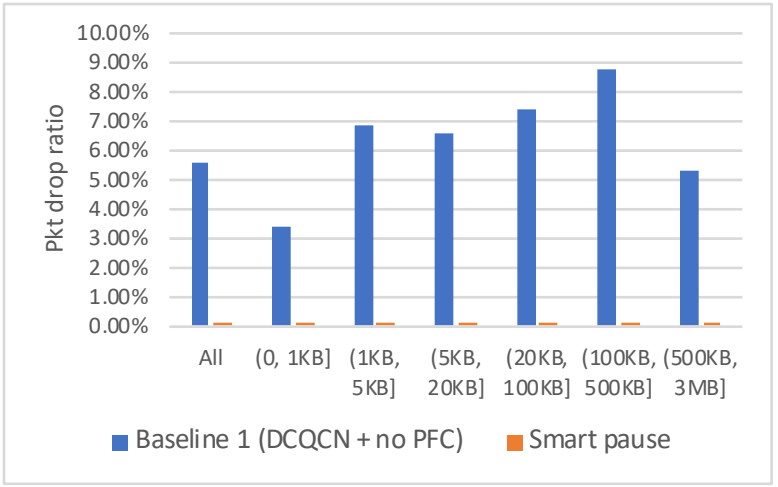


DCQCN (2MB buffer, incast fan-in 1) + smart pause: various loads

60% load
% of packets dropped



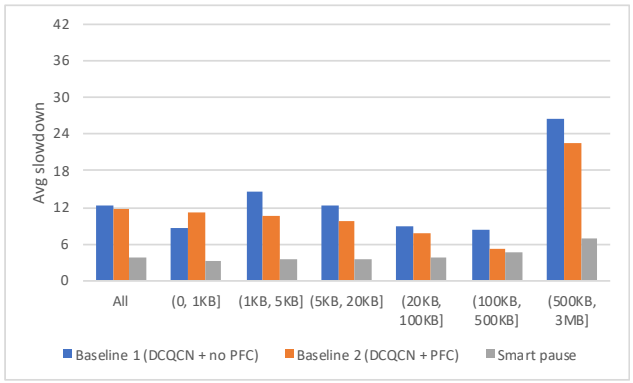
80% load
% of packets dropped



DCQCN (2MB buffer, 60% load) + smart pause: various incast fan-in

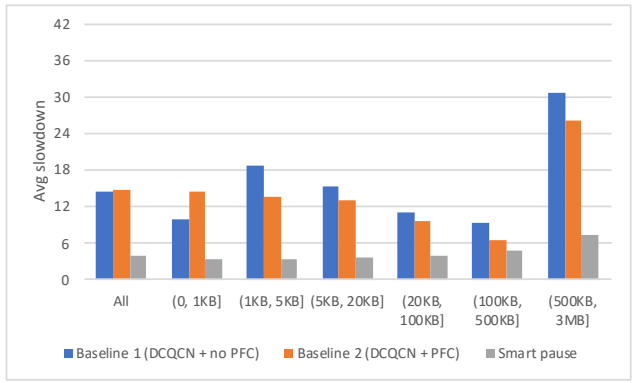
fan-in = 1,
avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.2x	2.7x	4.4x	3.5x	2.4x	1.8x	3.8x
3.1x	3.5x	3.2x	2.8x	2.0x	1.1x	3.2x



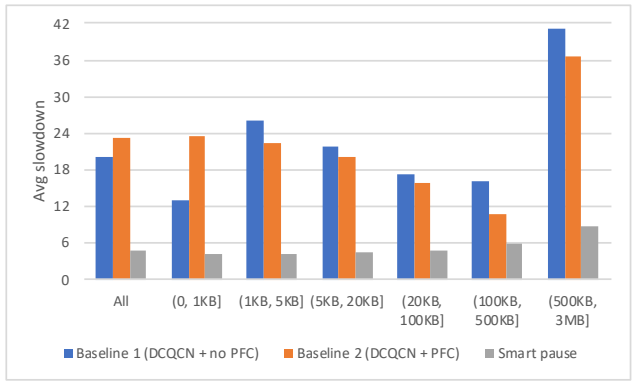
fan-in = 1(50%), 2(30%), 4(20%),
avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.6x	2.9x	5.3x	4.2x	2.8x	2.0x	4.2x
3.7x	4.2x	3.9x	3.6x	2.5x	1.3x	3.6x



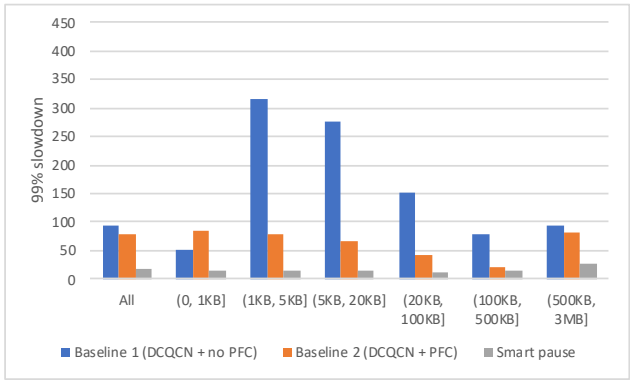
fan-in = 1(30%), 4(40%), 10(30%),
avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
4.1x	3.1x	6.0x	4.9x	3.6x	2.8x	4.6x
4.8x	5.7x	5.2x	4.5x	3.3x	1.9x	4.1x



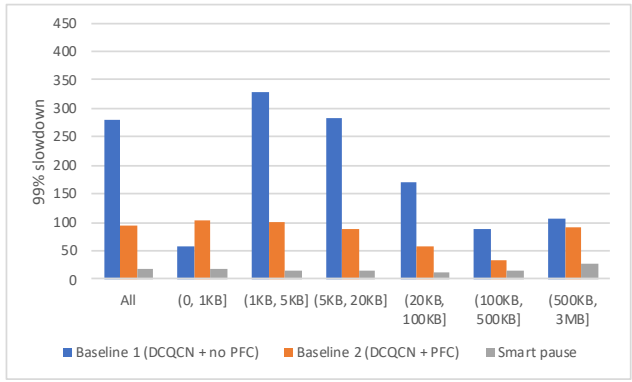
fan-in = 1,
99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
5.3x	3.1x	21.1x	19.6x	12.5x	5.7x	3.6x
4.5x	5.3x	5.2x	4.8x	3.4x	1.5x	3.2x



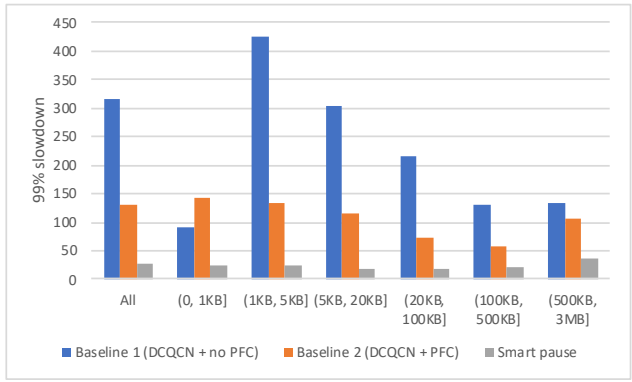
fan-in = 1(50%), 2(30%), 4(20%),
99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
14.5x	3.3x	20.4x	20.1x	13.5x	6.3x	3.8x
5.0x	5.9x	6.2x	6.3x	4.7x	2.4x	3.3x



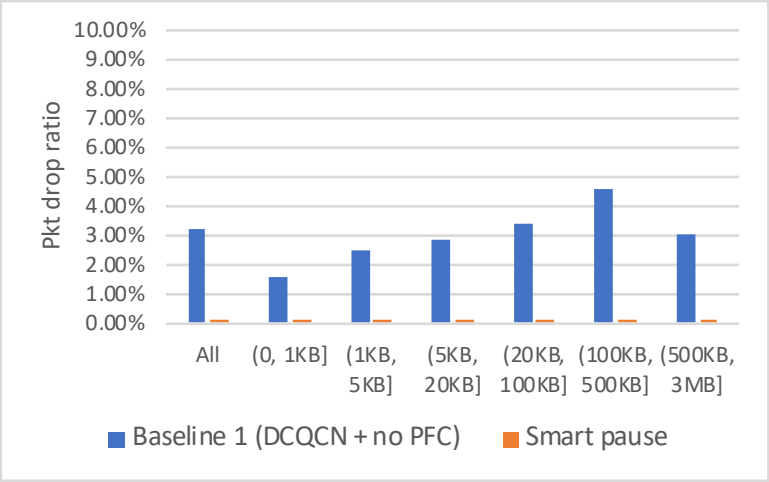
fan-in = 1(30%), 4(40%), 10(30%),
99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
11.8x	3.6x	18.3x	15.8x	12.5x	6.6x	3.7x
4.8x	5.8x	5.7x	6.1x	4.3x	2.9x	2.9x

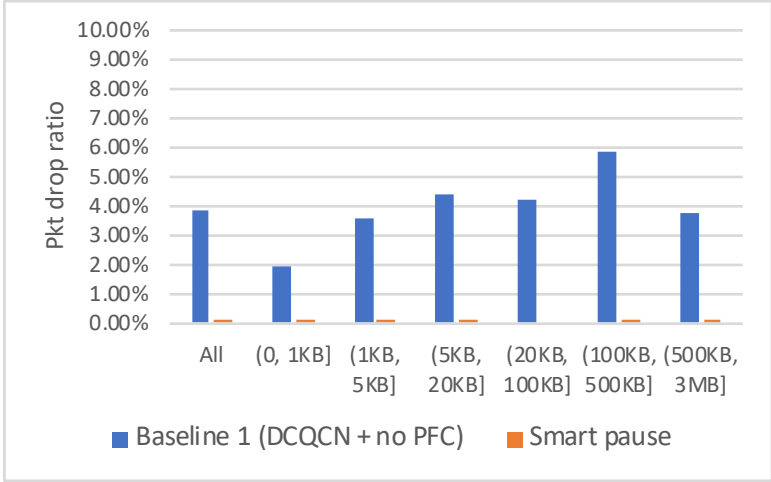


DCQCN (2MB buffer, 60% load) + smart pause: various incast fan-in

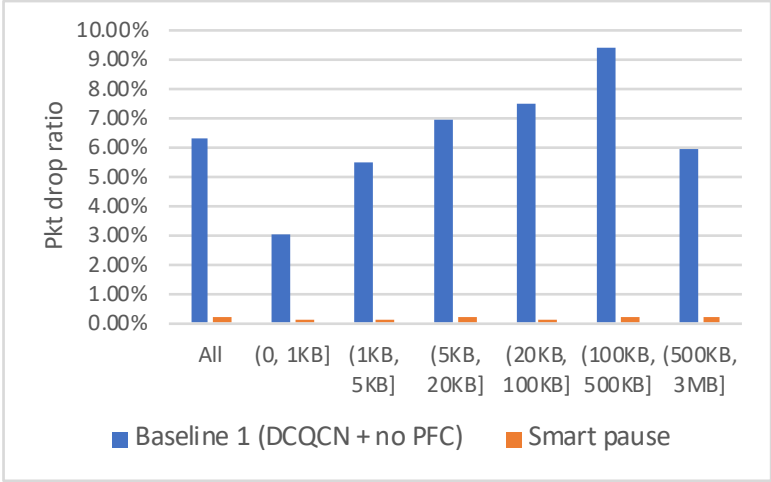
fan-in = 1,
% of packets dropped



fan-in = 1(50%), 2(30%), 4(20%),
% of packets dropped



fan-in = 1(30%), 4(40%), 10(30%),
% of packets dropped

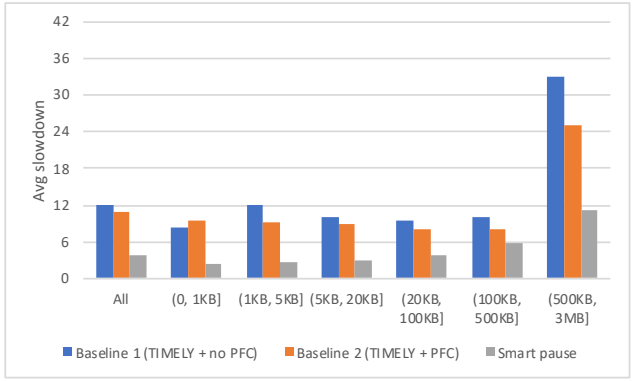


TIMELY + smart pause

60% load, fanout = 1, 2MB buffer

Avg slowdown

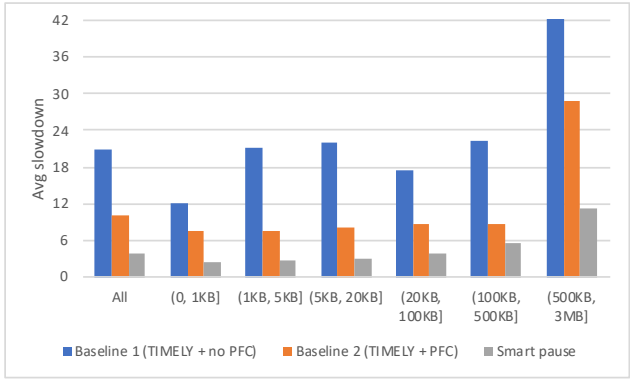
All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
3.3x	3.4x	4.6x	3.4x	2.6x	1.8x	3.0x
2.9x	3.9x	3.5x	3.1x	2.2x	1.4x	2.2x



60% load, fanout = 1, 1MB buffer

Avg slowdown*

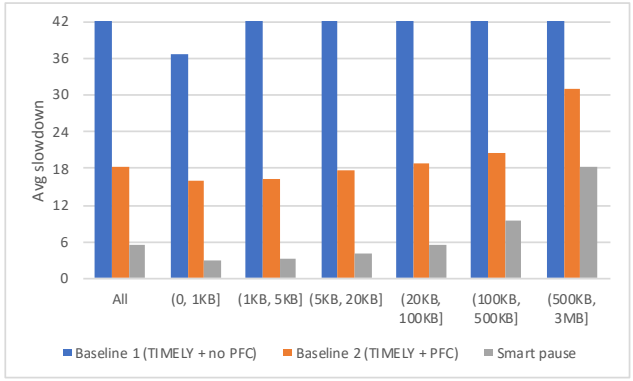
All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
5.6x	5.0x	8.2x	7.6x	4.7x	4.0x	5.3x
2.7x	3.1x	2.9x	2.8x	2.3x	1.5x	2.5x



80% load, fanout = 1, 2MB buffer

Avg slowdown*

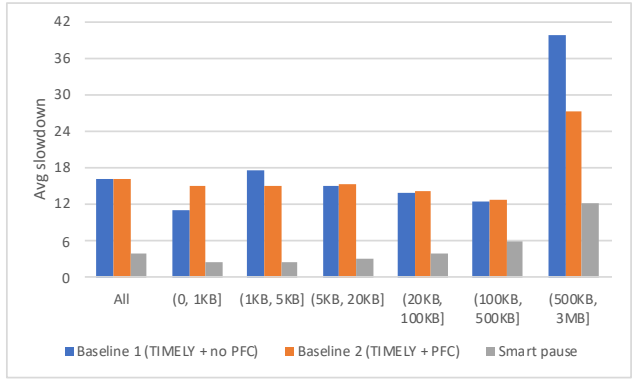
All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
13.6x	12.1x	26.1x	25.9x	20.6x	10.2x	8.4x
3.4x	5.3x	4.9x	4.4x	3.4x	2.2x	1.7x



60% load, fanout = 1(50%), 2(30%), 4(20%), 2MB buffer

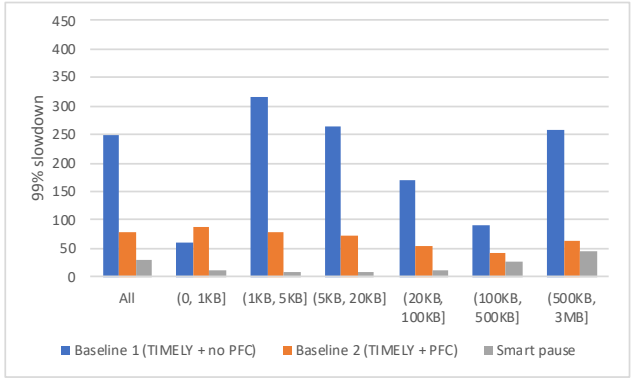
Avg slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
4.1x	4.4x	6.6x	5.0x	3.6x	2.1x	3.2x
4.2x	6.1x	5.6x	5.1x	3.6x	2.2x	2.2x



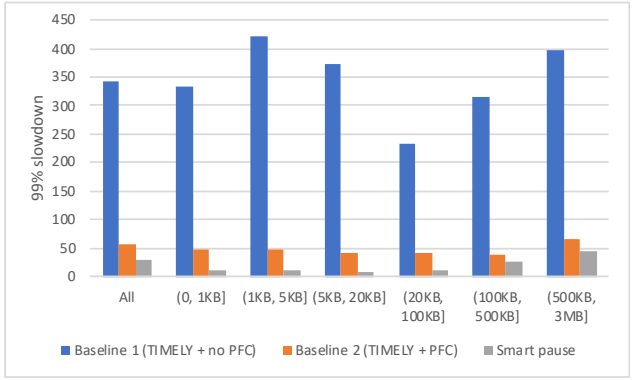
99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
8.5x	5.8x	33.3x	29.8x	14.4x	3.6x	5.7x
2.7x	8.6x	8.3x	8.3x	4.6x	1.6x	1.4x



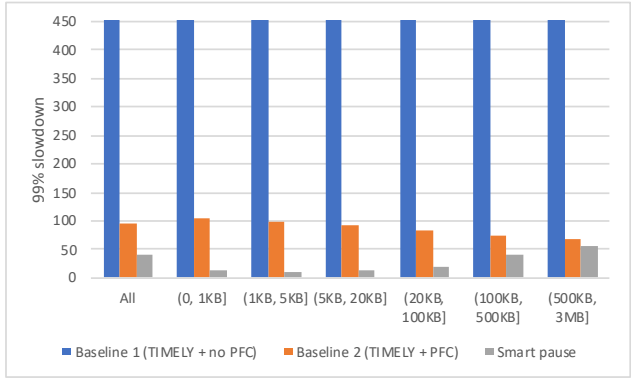
99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
11.6x	33.1x	43.6x	42.5x	19.9x	12.2x	9.0x
2.0x	4.8x	4.8x	4.7x	3.4x	1.5x	1.5x



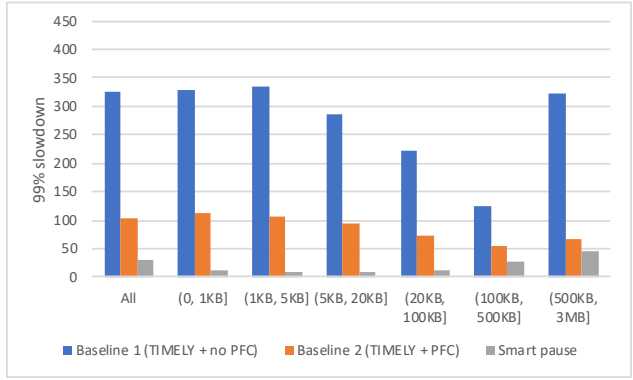
99% slowdown**

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
22.1x	54.2x	101x	95.6x	61.5x	15.1x	9.7x
2.4x	8.7x	8.6x	7.3x	4.0x	1.8x	1.2x



99% slowdown

All	(0, 1KB]	(1KB, 5KB]	(5KB, 20KB]	(20KB, 100KB]	(100KB, 500KB]	(500KB, 3MB]
10.4x	29.0x	32.3x	29.7x	19.1x	4.7x	6.9x
3.3x	9.9x	10.4x	9.7x	6.4x	2.1x	1.4x



*: y-axis is capped at 42. **: y-axis is capped at 450.

Conclusion

- Smart-Pause: Move Pause to the edge of the network
  - On a per-packet basis using on a novel measure of “path- and load-dependent” congestion
  - Acts independently of congestion control
- Smart-Pause on ECN-based (DCQCN) & delay-based (TIMELY) congestion control algorithm:
  - 3 – 5x reduction in average slowdown
  - 3 – 17x reduction in 99% slowdown
  - Drop ratio goes nearly to 0
  - Evaluations on other congestion control algorithms are ongoing