HPCC++: Enhanced High Precision Congestion Control

draft-miao-iccrg-hpccplus-00

Rui Miao, Hongqiang Harry Liu, Rong Pan, Jeongkeun Lee, Changhoon Kim, Barak Gafni, Yuval Shpigelman

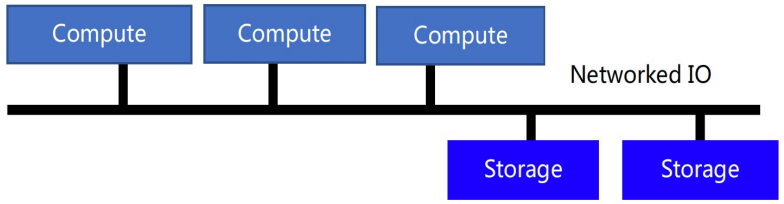
IETF-112 rtgwg

Nov 2021

Cloud desires hyper-speed networking

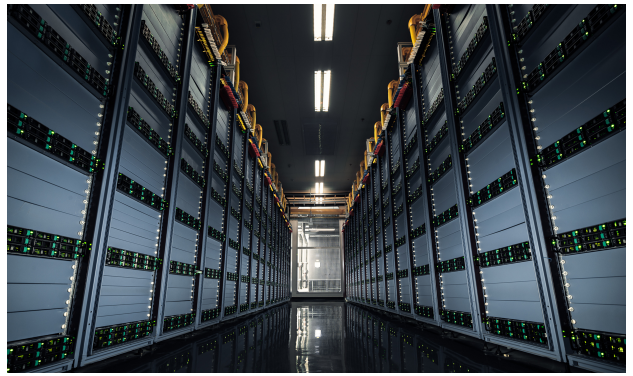
Today, clouds have
- bigger data to compute & store
- faster compute & storage devices
- more types of compute and storage resources

High-performance storage



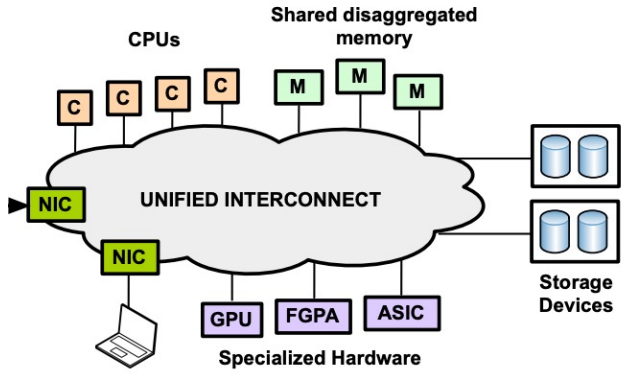
- Storage-compute separation is norm
- HDD→SSD→NVMe
- Higher-throughput, lower latency
- 1M IOPS / 50~100us

High-performance computation



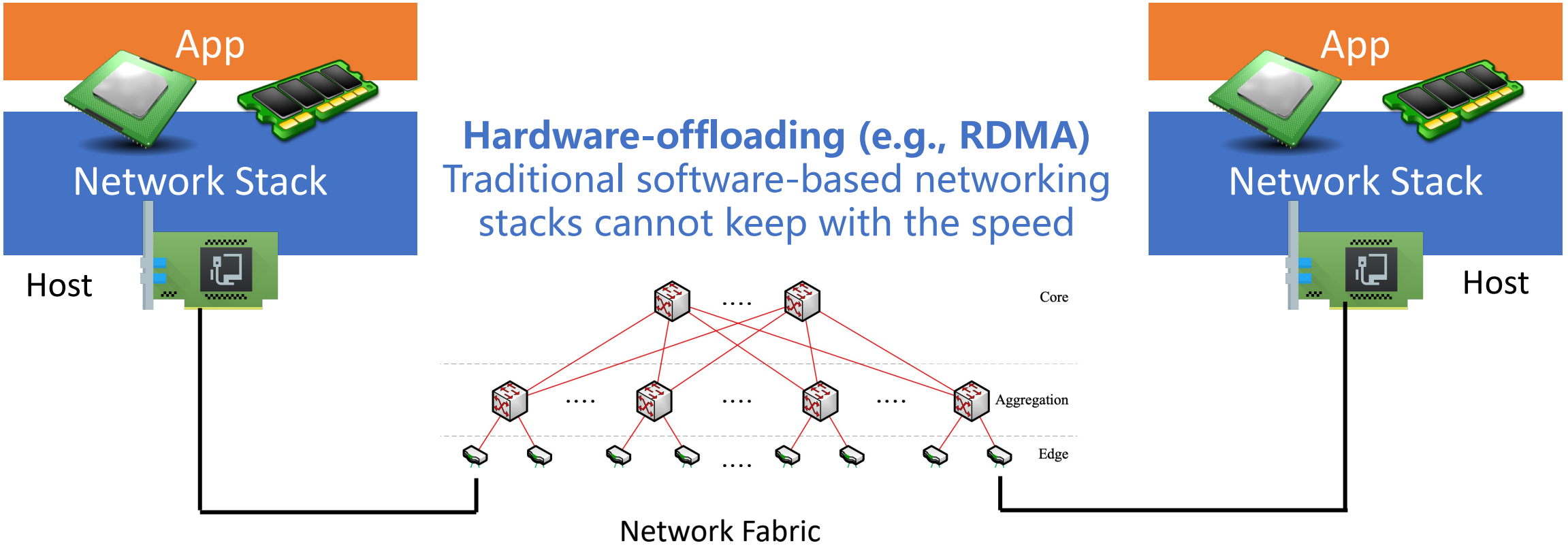
- Distributed deep learning, HPC
- CPU→GPU, FPGA, ASIC
- Faster compute, lower latency
- E.g. latency <10us

Resource disaggregation



- More network load
- Need ultra-lower latency: 3-5us, > 40Gbps (Gao Et.al. OSDI'16)

Hyper-speed network chips != hyper-speed networking



Hardware-offloading (e.g., RDMA)
Traditional software-based networking stacks cannot keep with the speed

Congestion control (CC)
Since, end hosts are aggressive, network is more vulnerable to congestion & packet loss

Realistic challenges in CC in high-speed networks

Challenges in current CC

- Operation challenge-1: PFC storm & deadlock
  - **Running lossy networks is desired, but there is a convergence challenge!!!**

Challenge-1: Slow Convergence

- Operation challenge-2: running multiple applications
  - **QoS queues are scarce resources!!!**

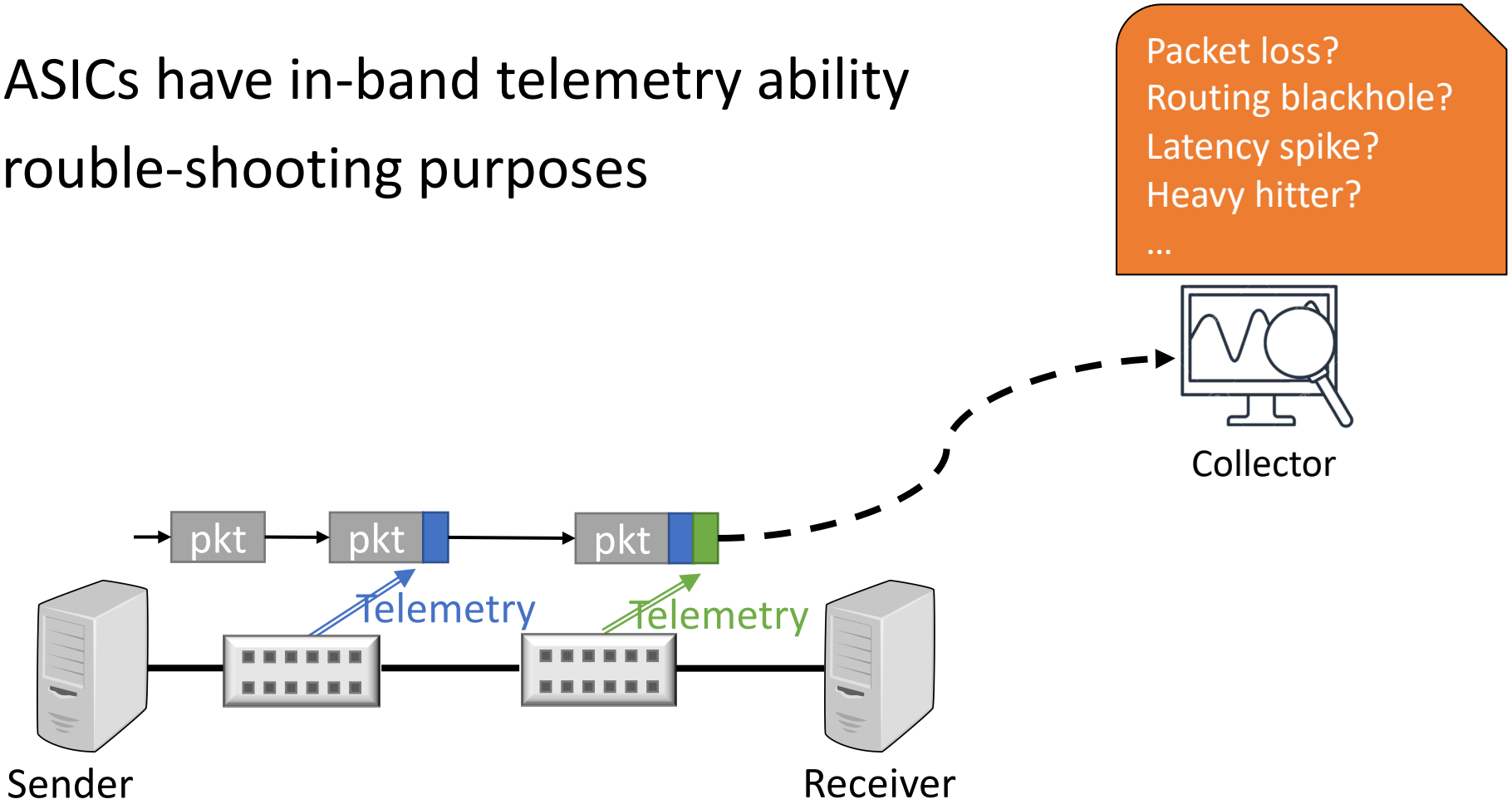
Challenge-2: Standing queue

- Operation challenge-3: complex parameter tuning
  - **DCQCN has at least 15 parameters to tune!!!**

Challenge-3: Heuristics in CC

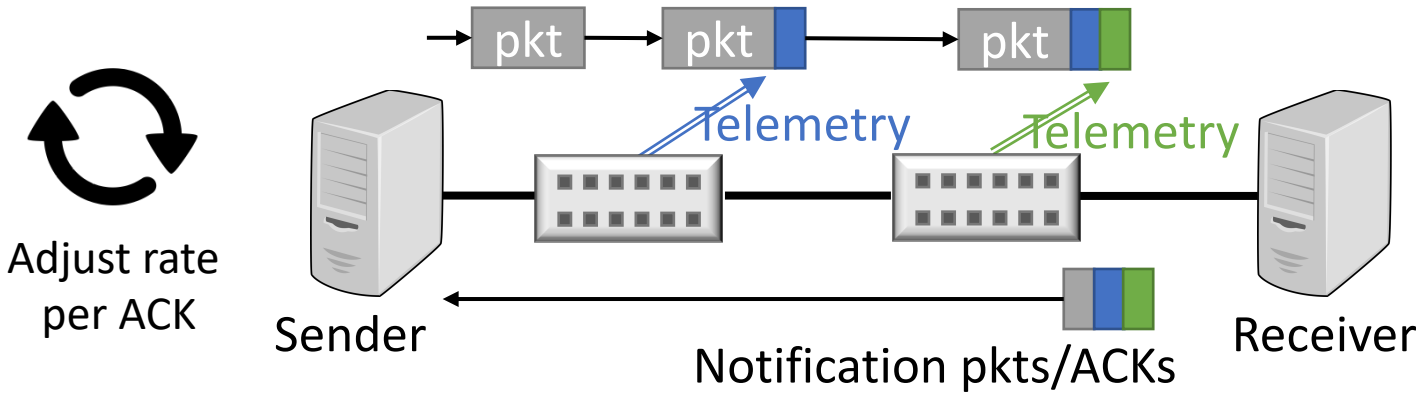
In-band telemetry

- New commodity ASICs have in-band telemetry ability
- Mainly used for trouble-shooting purposes



HPCC++: Enhanced High Precision Congestion Control

Can we use in-band telemetry as precise feedback for congestion control?



In-band telemetry format

- HPCC++ defines the algorithm of using telemetry information
  - including queue length, transmitted bytes, timestamp, link capacity, etc.
- Yet, the actual packet format is up to the environment

bits	31-24		23-16	15-8	7-0
0	Device-ID				PT
1	TID	congestion	Tx Bytes Cnt[39:32]	TTL	Queue ID
2	Rx Timestamp Sec - Upper				
3	Rx Timestamp Sec			Rx Timestamp Nano Upper	
4	Rx Timestamp Nano			Tx Timestamp Nano Upper	
5	Tx Timestamp Nano			Egress Queue Cell Cnt	
6	Src-Sys-Port			Dest-Sys-port	
7	Tx Bytes Cnt[31:0]				

Example format of in-band telemetry used by HPCC++

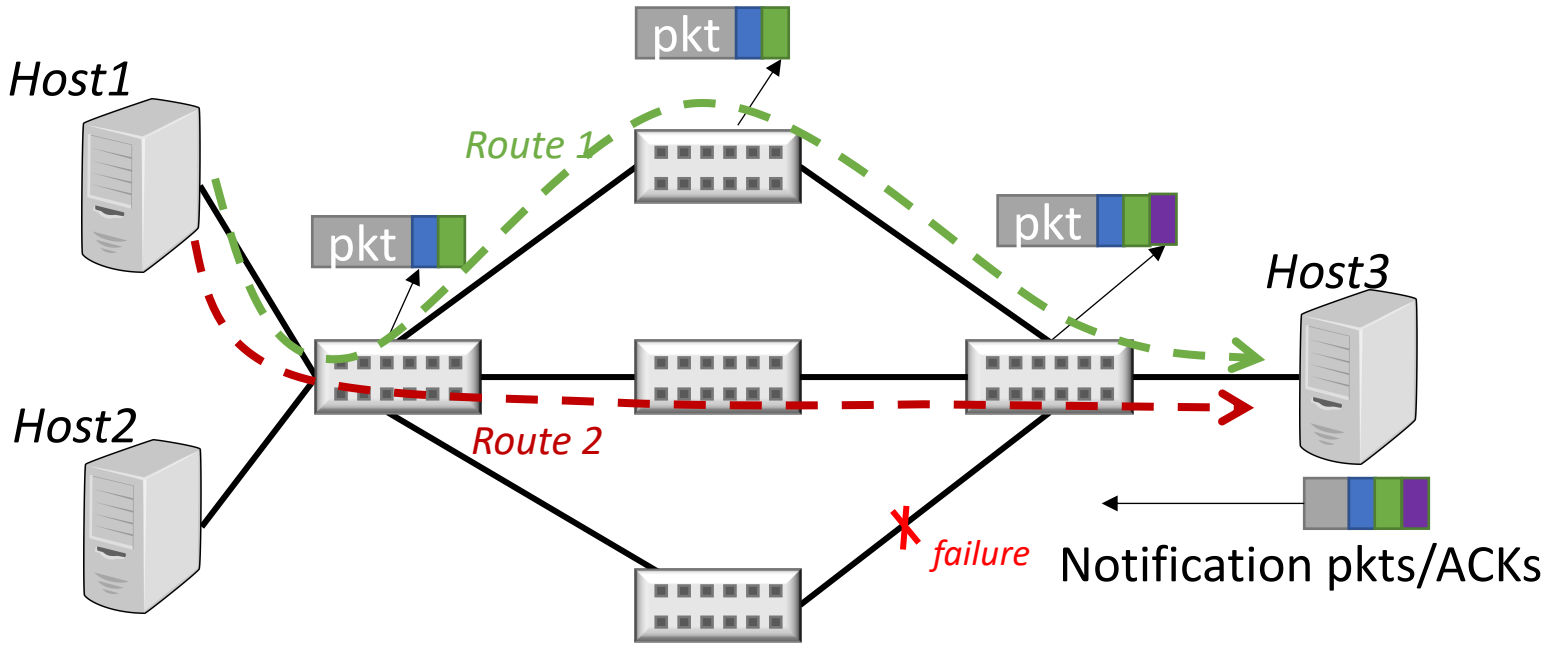
HPCC++ solves all there problems

Using in-band telemetry as the precise feedback

- Fast convergence
  - Sender knows the precise rate to adjust to, on every ACK
- Near-zero queue
  - Feedback does not only rely on queue
- Few parameters
  - Precise feedback, so no need for heuristics which requires many parameters

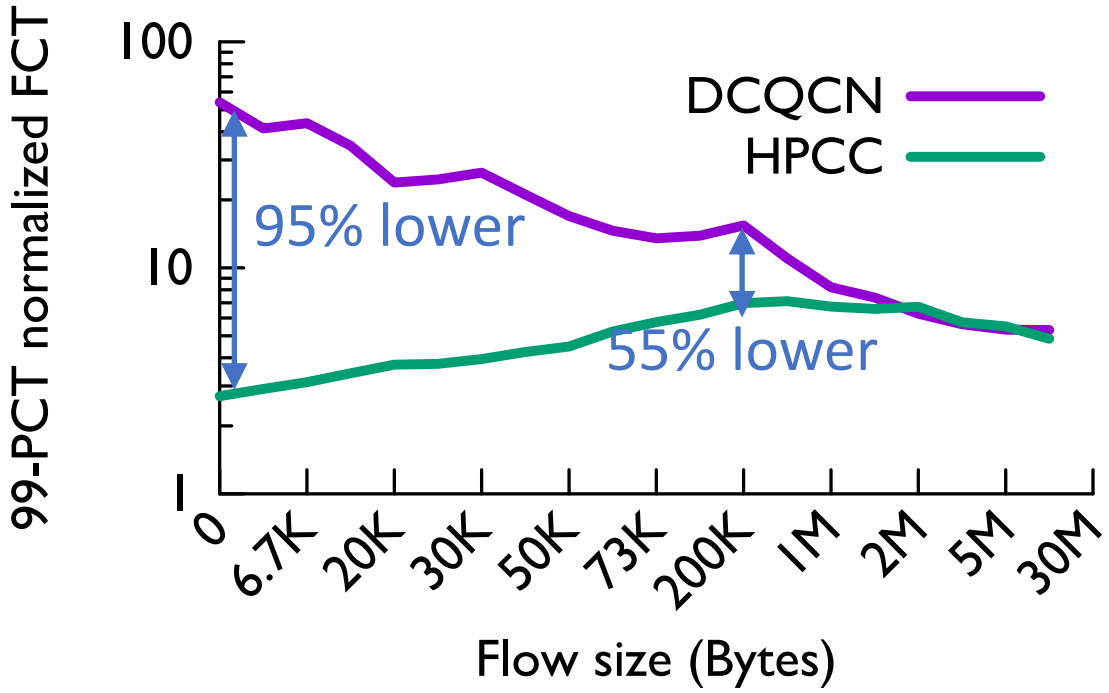
Smart routing with HPCC++

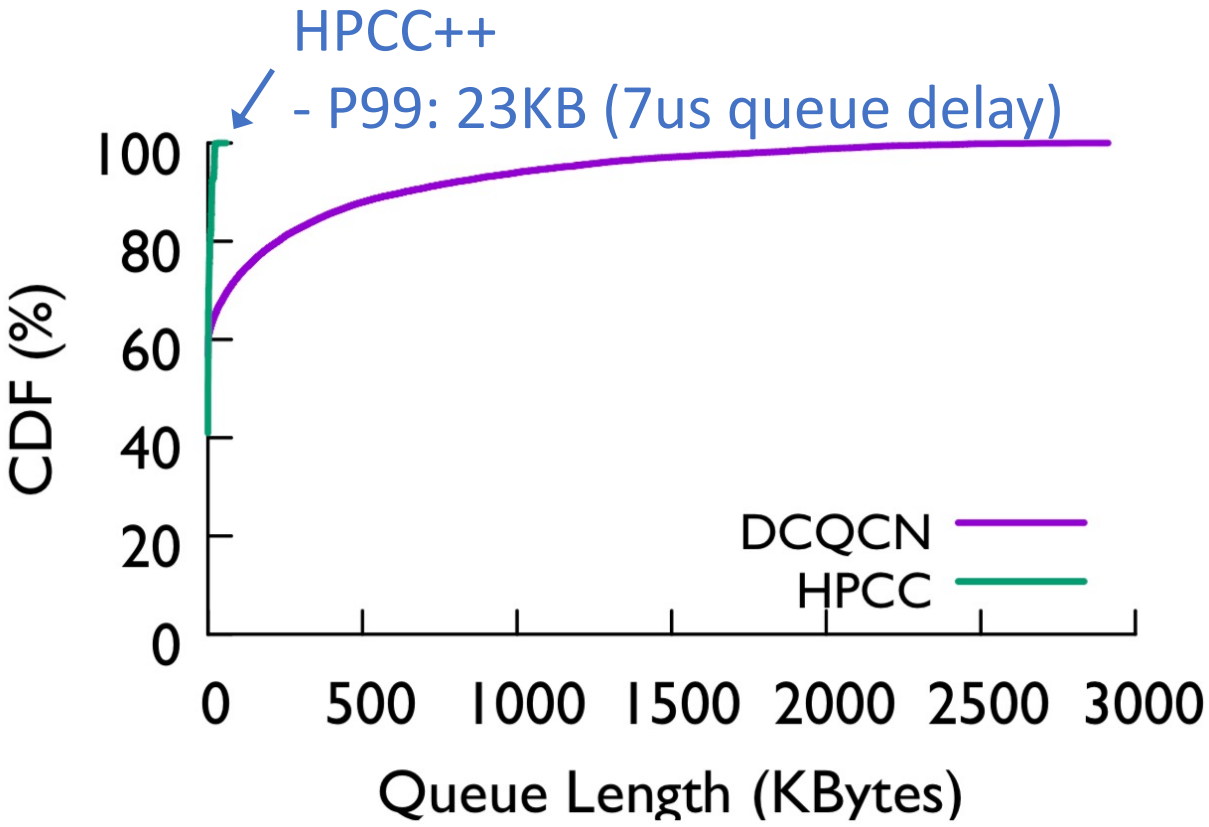
- Route condition changes drastically in DC
  - e.g, incast, failure, re-routing, …
- HPCC++ offers a precise *"view"* of a route's capacity
- Joint decision on routing and traffic allocation by combining views of routes



HPCC++ achieves lower FCT and near-zero queue

- In testbed, vs. DCQCN (hardware-based, widely used in industry)
  - Web search traffic at 50% load
- Vs. other CC (unavailable in HW) in simulation. HPCC performs better





Thank You