BFC: Backpressure Flow Control

Prateesh Goyal, Preey Shah, Kevin Zhao,
Georgios Nikolaidis, Mohammad Alizadeh, Tom Anderson

Microsoft Research

MIT CSAIL

Indian Institute of Technology Bombay

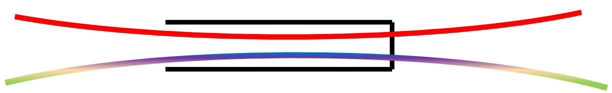
Paul G. Allen School of Computer Science & Engineering

intel

Lots of Existing Congestion Control Protocols

- Congestion control goals
  - High throughput
  - Low tail latency

Long-running flow



Cross-traffic flows
(Load: 60% of capacity, Flow size: Facebook Workload)

Time

- Swift
- HPCC
- Homa
- Express Pass
- Timely
- DCQCN
- DCTCP

Scheme	Norm. Throughput (%) (long-running flow)	99^{th} %ile Qdelay (μs) (short flows)
HPCC	57	23.9
DCQCN	25	30.4

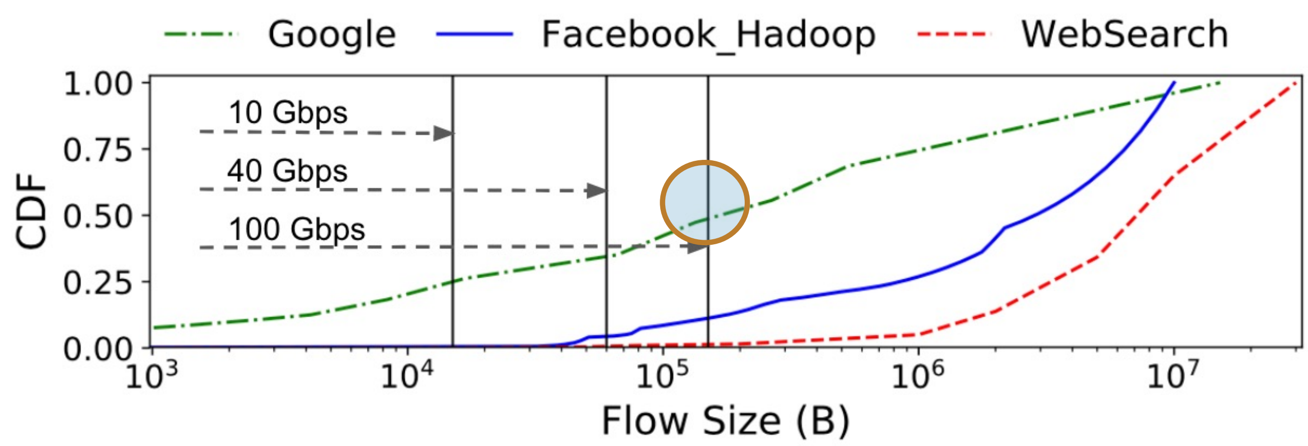
E2E Feedback Loops are too Slow for Datacenters

- High feedback delay: network round-trip-time (E2E RTT)
  - Acting on stale information can hurt performance
- Network conditions in datacenters are highly variable
  - High speed links (40/100 G)
  - Most flows are short: Bursty traffic

↑ Link Speed → ↑ Uncontrolled Traffic

- No feedback in the first RTT
  - Blind start → trade-off between under-utilization and congestion
- ↑ Uncontrolled traffic → ↑ packet drops

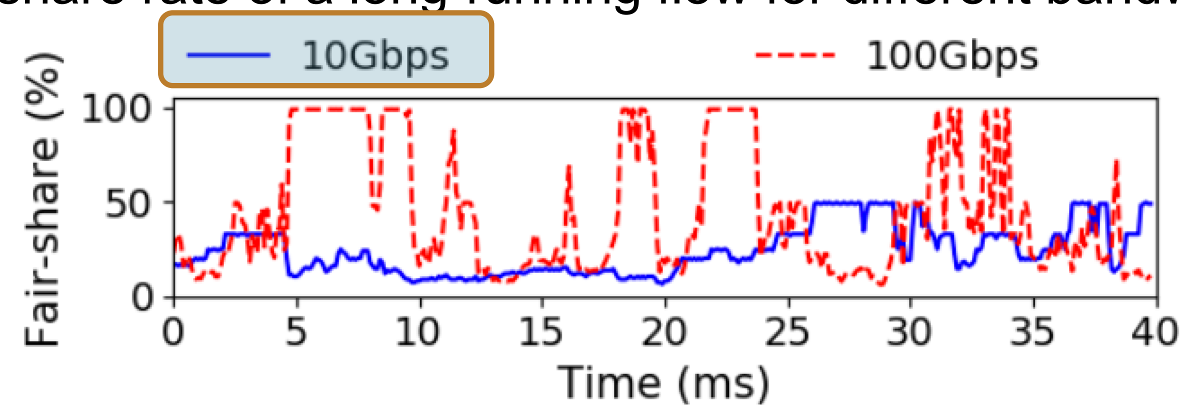
Cumulative traffic bytes contributed by flows of different sizes

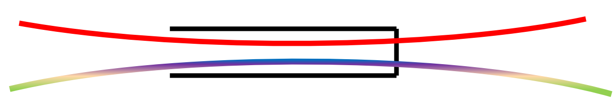


↑ Link Speed → ↑ Cross-traffic Variability

- Long flows can struggle to determine the right rate

Fair-share rate of a long-running flow for different bandwidths

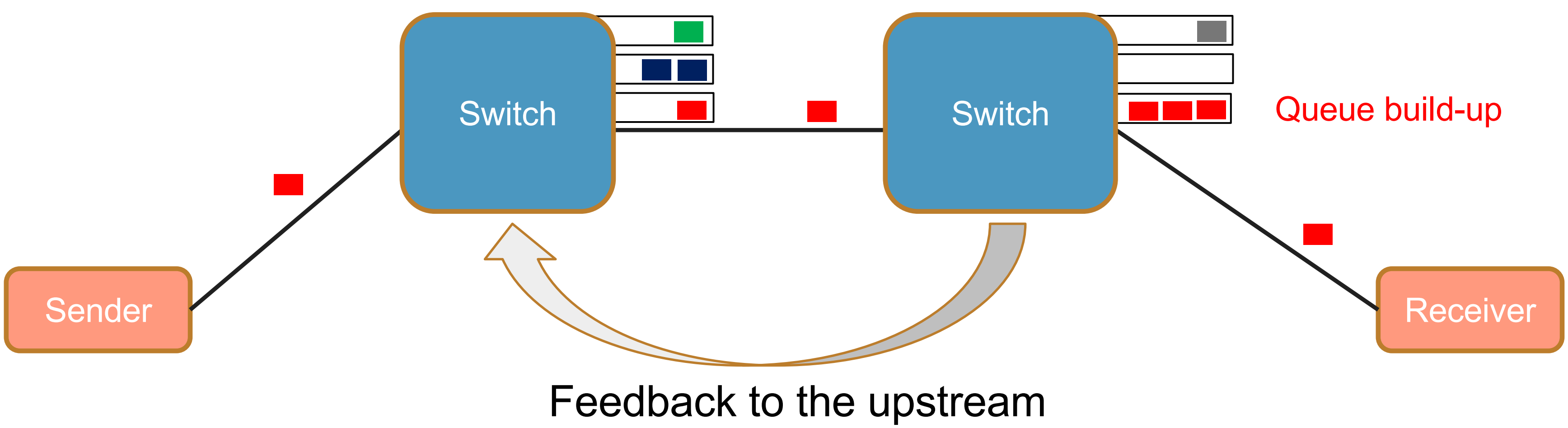




1 Long-running flow
Cross-traffic load: 60% of link capacity
Flow size: Facebook Workload

Its Time to Revisit per-hop per-flow Flow Control

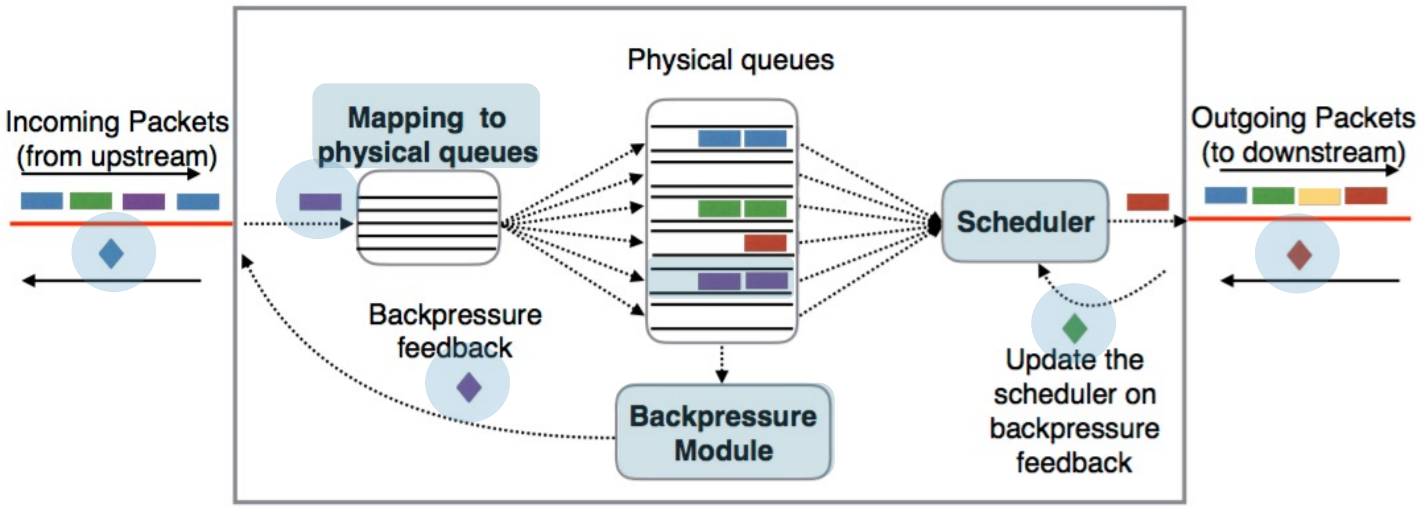
- Low buffering, low tail latency, high throughput
  - Faster reaction: 1-Hop RTT vs 1 E2E RTT
  - Per-flow queue → no head-of-line (HoL) blocking
    - Service rate of a flow is not unjustly affected by other flows



Challenges in per-hop per-flow Flow Control

Limited state, Limited # of queues, Limited programmability

Logical switch components (per-port)



Backpressure Flow Control (BFC)

- Approximate per-hop per-flow control
  - Minimal HoL blocking → low tail latency
- Pause flows aggressively and selectively
  - Low buffering, high utilization
- Feasible: Limited switch state and simple operations

Backpressure Flow Control (BFC)

- Key ideas
  1. Only track active flows
  2. Dynamic queue assignment
  3. Communicate state across switches

Idea 1: Only Track Active Flows

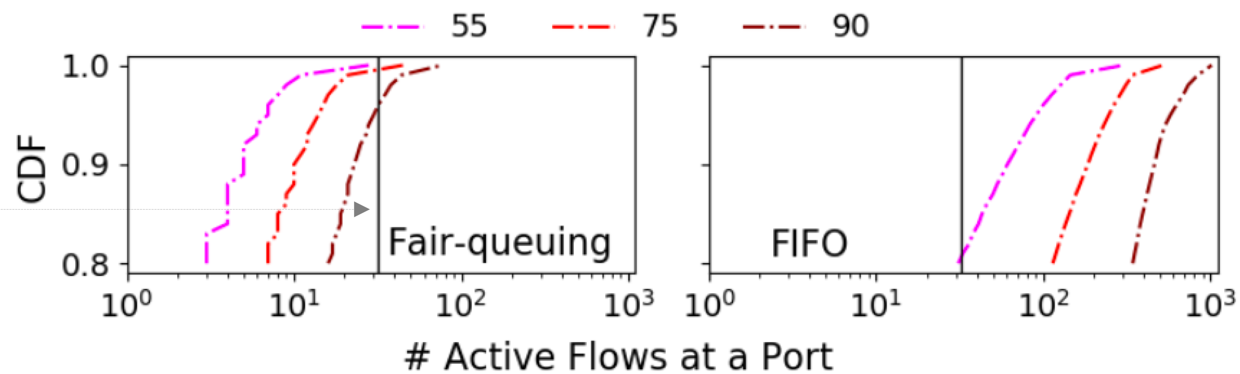
- Active flow: flow with packets queued at the switch

Idea 1: Only Track Active Flows

- Active flow: flow with packets queued at the switch
- Fair queueing → even smaller # of active flows

Active flows for different loads and scheduling policy.

Queues in Tofino2 (32/100G port) (128/400G port)



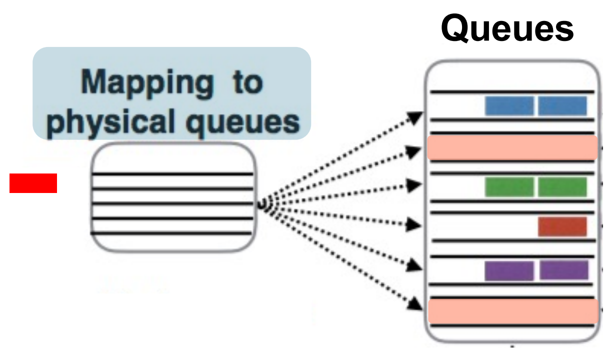
Google Workload
Bursty Log-normal flow inter-arrival
100Gbps port

Idea 2: Dynamic Queue Assignment

- Goal: Minimize HoL blocking
  - Collisions (flows sharing a queue) degrade performance
- Naive approach: Stochastically hash flows to queues ❌
  - Birthday paradox - Collisions with modest # active flows
  - E.g., 5 active flows, 32 queues → 28% chance of collision

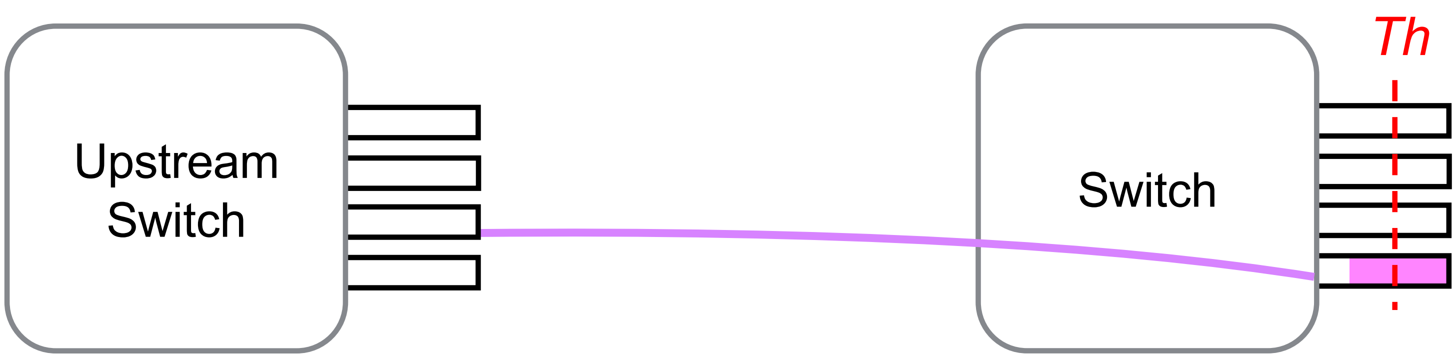
Idea 2: Dynamic Queue Assignment

- BFC: Dynamically assign new flows to empty queues
  - No collisions when # active flows < # of queues
  - Minimal HoL blocking → low tail latency



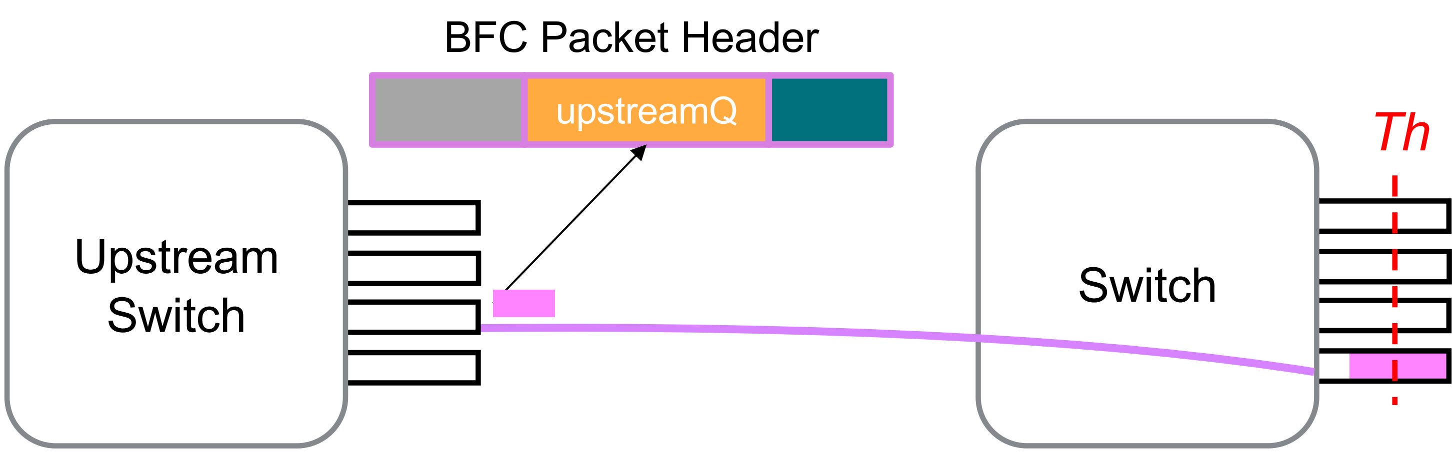
Idea 3: Communicate State across Switches

- Pause a flow (at the upstream) if
  - qLength at current switch > *Th*



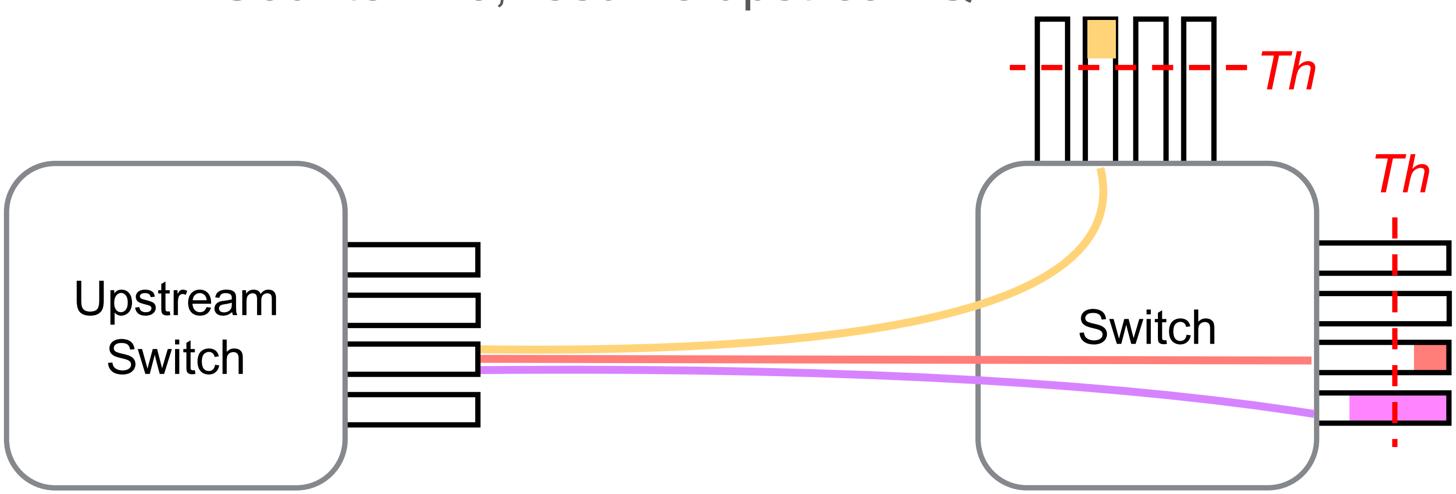
Idea 3: Communicate State across Switches

- Pause a flow (at the upstream) if
  - qLength at current switch > *Th*
- Header includes qAssigment at the previous hop (upstreamQ)
  - Switch pauses the upstreamQ directly (on a packet arrival) if
    - qLength at current switch > *Th*



Idea 3: Communicate State across Switches

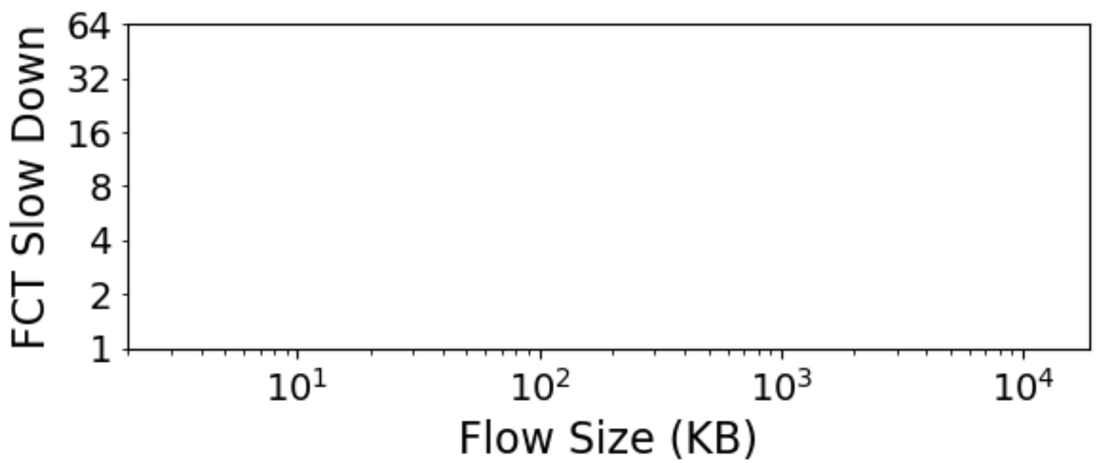
- Resume an upstreamQ if
  - qLength at current switch < *Th*, ∀ **flows** from the upstreamQ
- For each upsteamQ,
  - Count # of packets queued that exceeded *Th (on packet arrival)*
    - If Counter = 0, resume upstreamQ



Evaluation

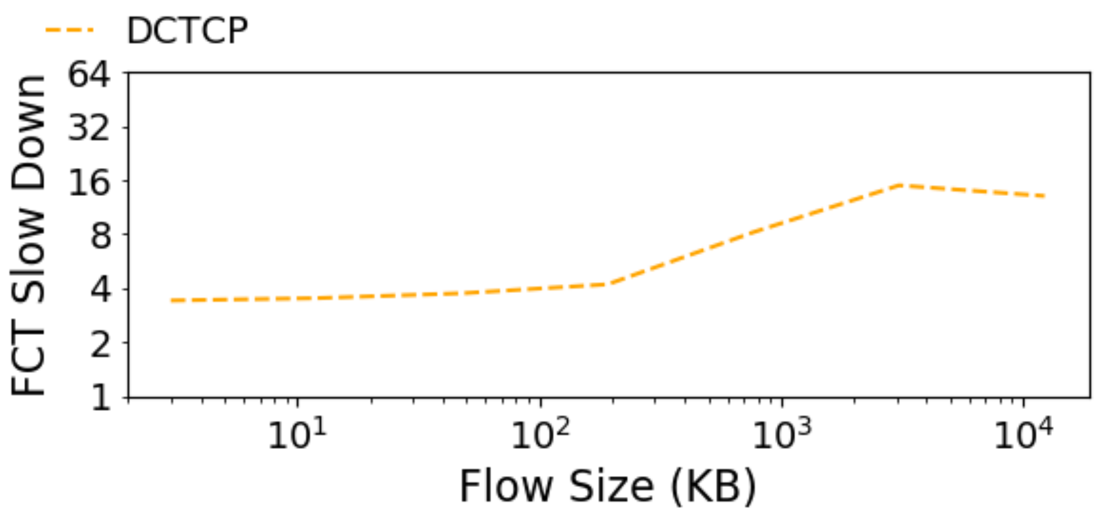
- Tofino2 (proof-of-concept)
  - P4-based programmable switch
  - Pause/resume queues from the *dataplane* at line rate (400 Gbps)
- NS-3
  - Large-scale packet-level simulations
  - Vary: Traffic load, incast degree, flow size distribution

Evaluation: Simulation



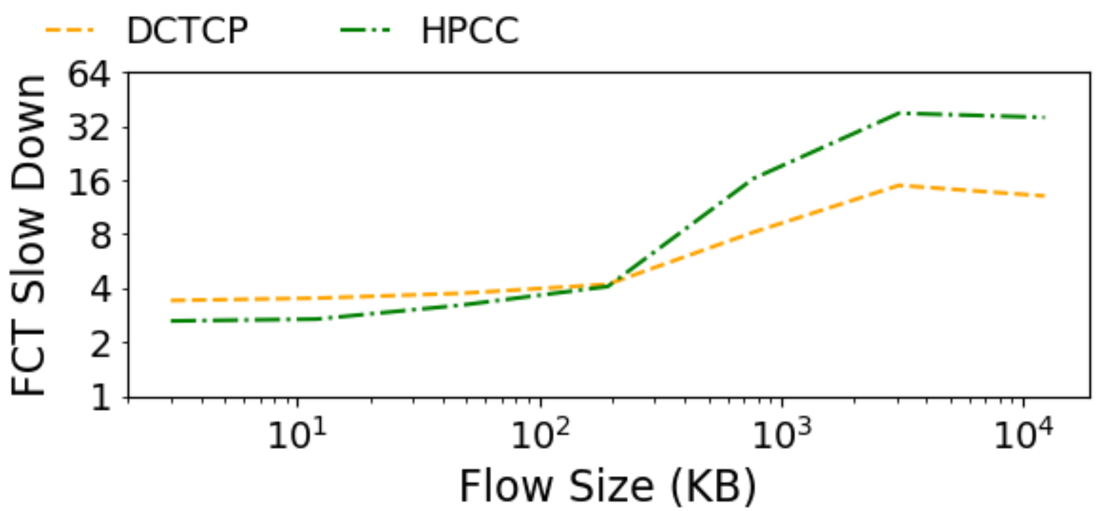
Topology: 2-level Clos (128 leaf servers)
100Gbps links, BDP: 100 KB
Flow sizes: Facebook Workload
Network load: 60%

Evaluation: Simulation



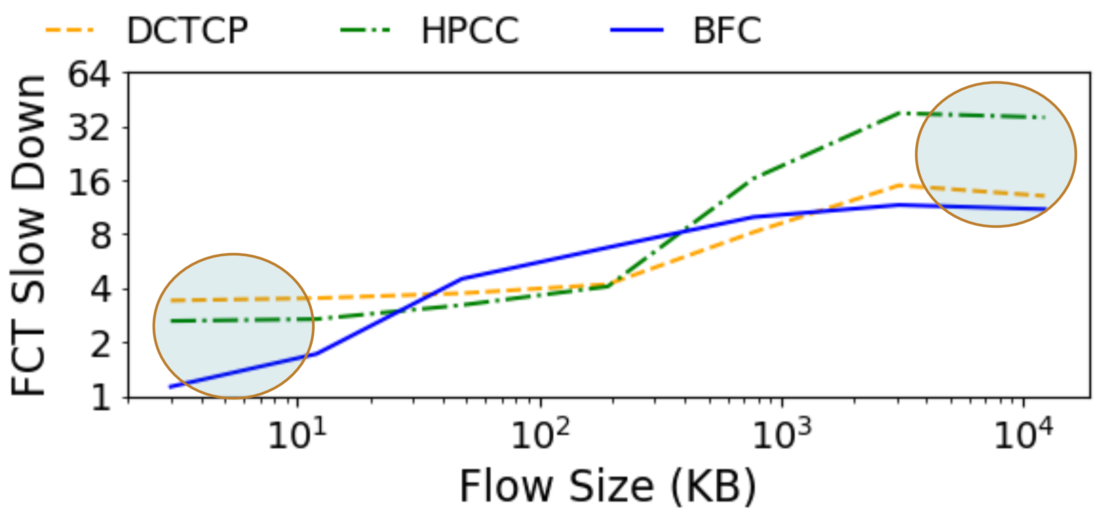
Topology: 2-level clos (128 leaf servers)
100Gbps links, BDP: 100 KB
Flow sizes: Facebook Workload
Network load: 60%

Evaluation: Simulation



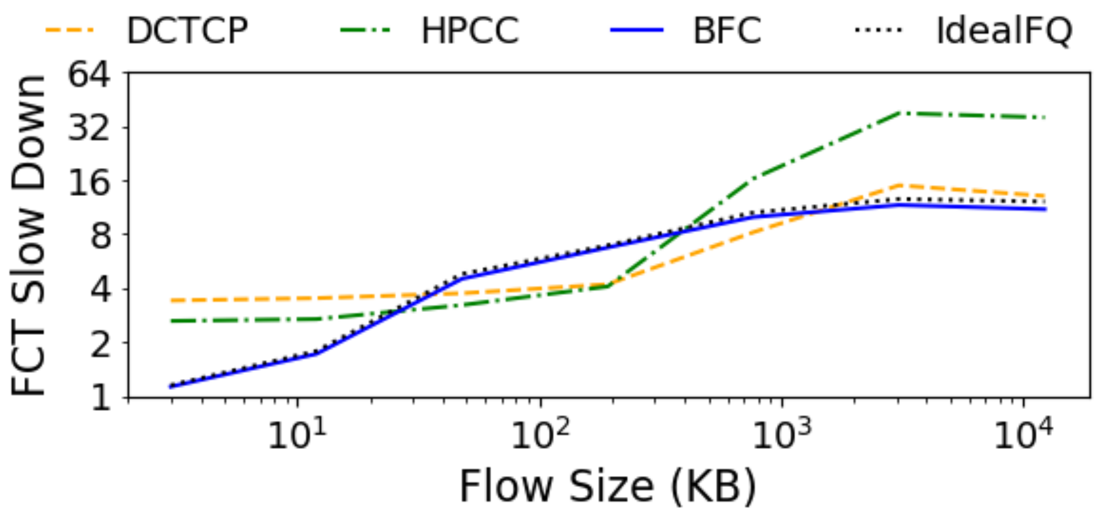
Topology: 2-level clos (128 leaf servers)
100Gbps links, BDP: 100 KB
Flow sizes: Facebook Workload
Network load: 60%

Evaluation: Simulation



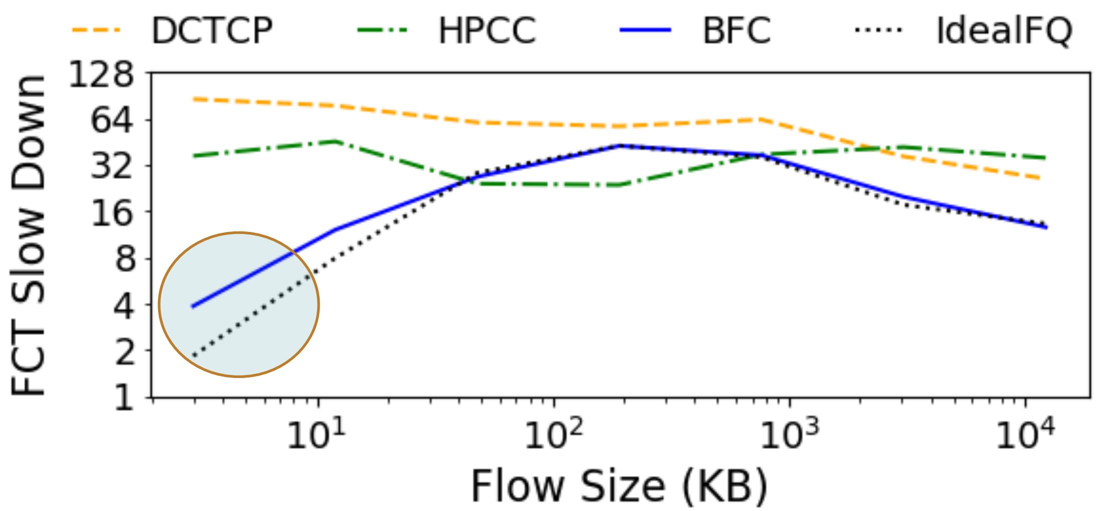
Topology: 2-level clos (128 leaf servers)
100Gbps links, BDP: 100 KB
Flow sizes: Facebook Workload
Network load: 60%

Evaluation: Simulation



Topology: 2-level clos (128 leaf servers)
100Gbps links, BDP: 100 KB
Flow sizes: Facebook Workload
Network load: 60%

Evaluation: Simulation (Incast)



2-level clos topology
Flow sizes: Facebook Workload
Load: 55 % + 5% 100-1 incast
Aggregate size of an incast: 20MB
New incast every 0.5 ms

Backpressure Flow Control (BFC)

- Key ideas
  1. Only track active flows
  2. Dynamic queue assignment
  3. Communicate state across switches

Thank You

- Per-hop per-flow flow control is great
  - Low buffering
  - Low tail latency
  - High Throughput
- Per-hop per-flow flow control is feasible