LossPass: Absorbing Microbursts by Packet Eviction for Data Center Networks

Transactions on Cloud Computing

Gyuyeong Kim and Wonjun Lee

Network and Security Research Lab. (NetLab)

Korea University, Republic of Korea

Background – Data Center Networks (DCNs)

- Fundamental infrastructure for modern services
  - Abundant computing resources
  - Economy of scale

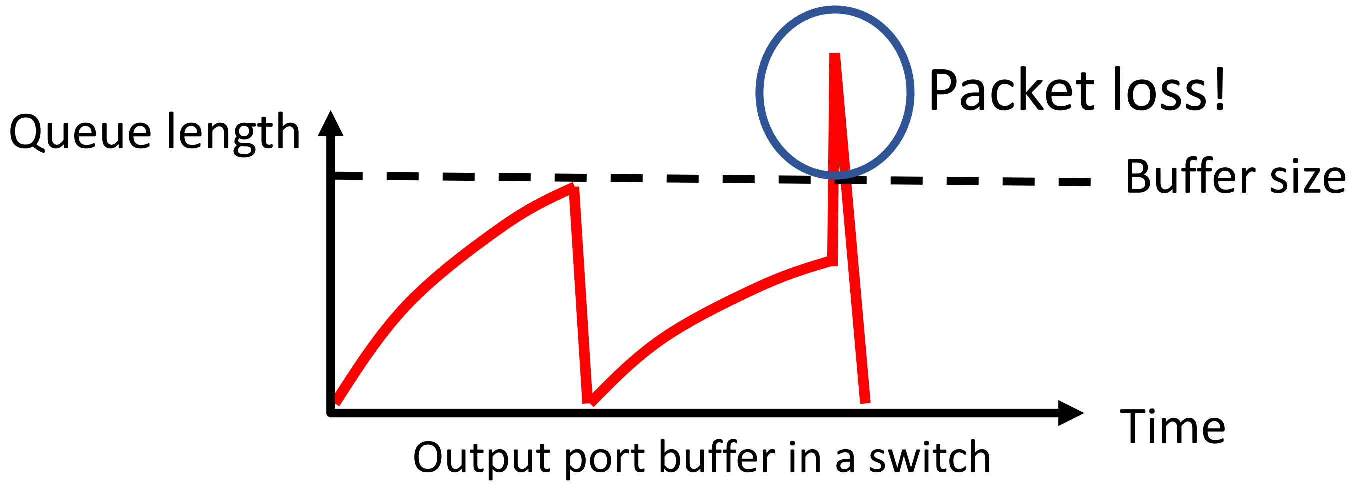
Microsoft data centers

Facebook data centers

NetLab
Network and Security Research Lab.

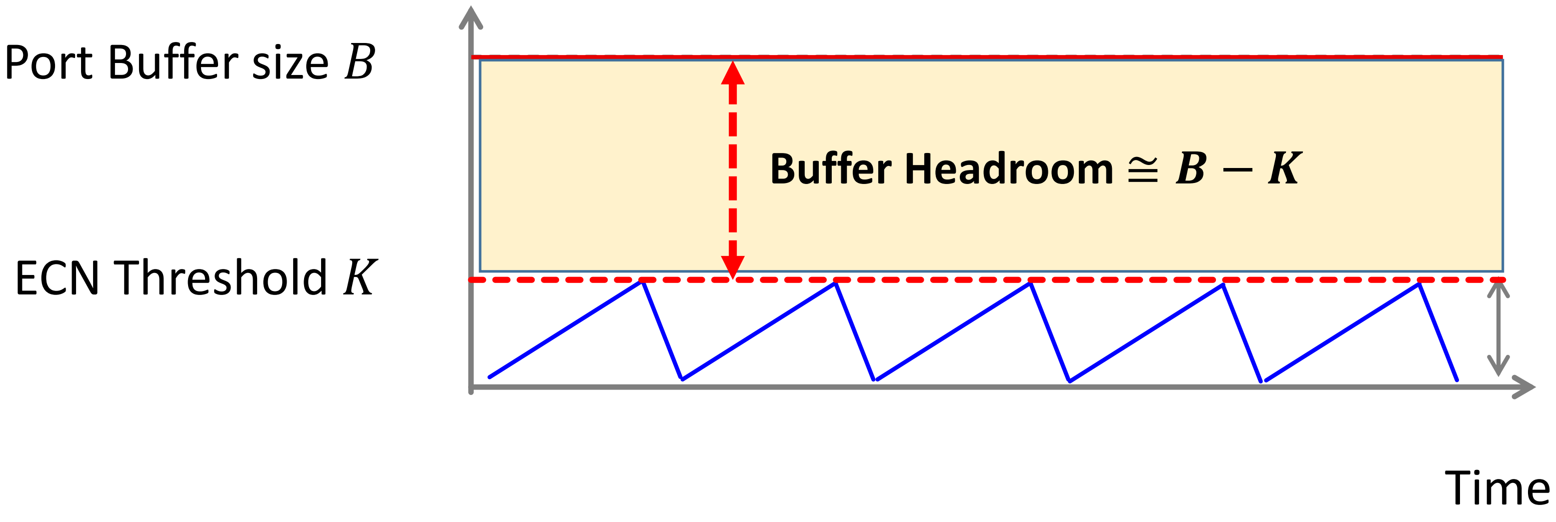
Motivation – Microbursts in DCNs

- Bursty traffic pattern consisting of many small flow packets
- Primary cause of transient congestion events in DCNs
- FCT of small flows can be lengthened multiple times due to timeout



Motivation – Explicit Congestion Notification (ECN)

- ECN is widely employed in many transport solutions
  - Marks packets if the instantaneous queue length exceeds the ECN marking threshold K
  - Maintains maximum queue length around K
- Leaves **buffer headroom** where microburst can be absorbed
  - More headroom, more burst tolerance



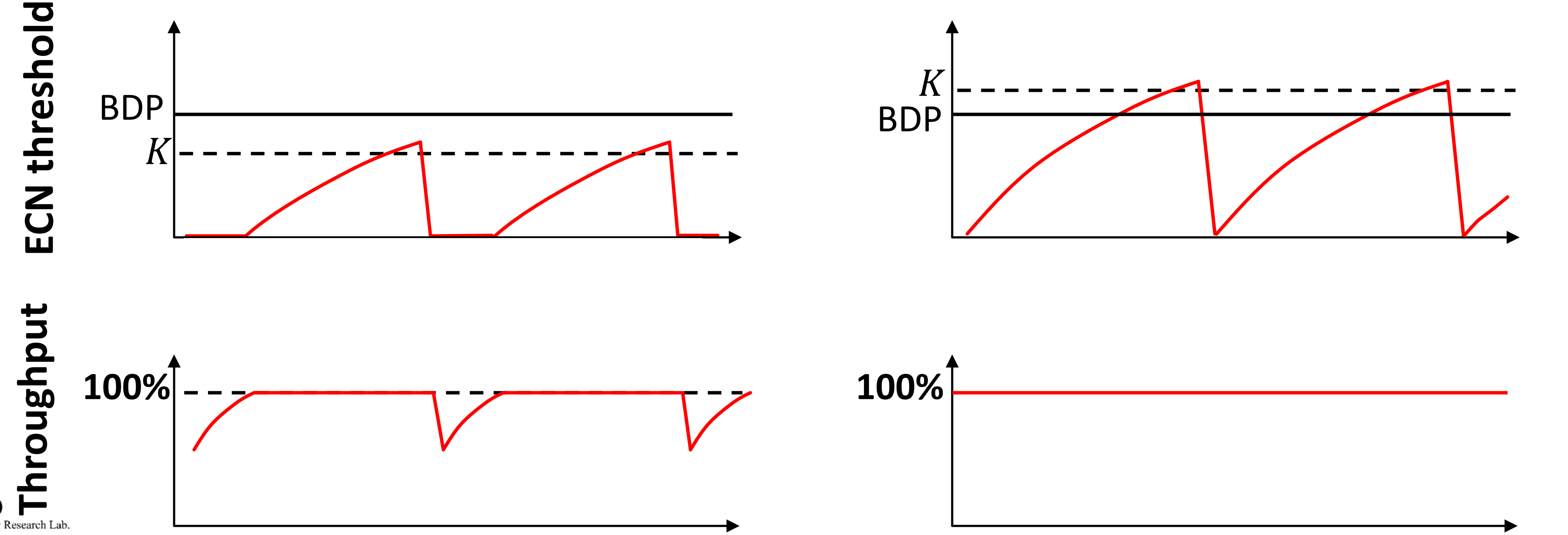
Motivation – Tradeoff of ECN

- Buffer headroom causes a tradeoff between latency and throughput
- Switch requires at least $C \times RTT$ (Bandwidth-Delay Product, BDP) of buffer space to saturate the bottleneck capacity

$$K < C \times RTT \qquad\qquad K \geq C \times RTT$$

High burst tolerance but throughput loss

Line-rate throughput but poor burst tolerance



NetLab
Network and Security Research Lab.

Motivation - Problem and Requirements

- **Q:**How to absorb microbursts as many as possible while maintaining line-rate throughput?

- **Minimized tail latency:** should minimize the tail FCTs of small flows
- **Line-rate throughput:** the link capacity should be fully utilized anytime
- **No headroom:** should not reserve buffer headroom to absorb microbursts
- **Being practical:** should be inexpensive to implement

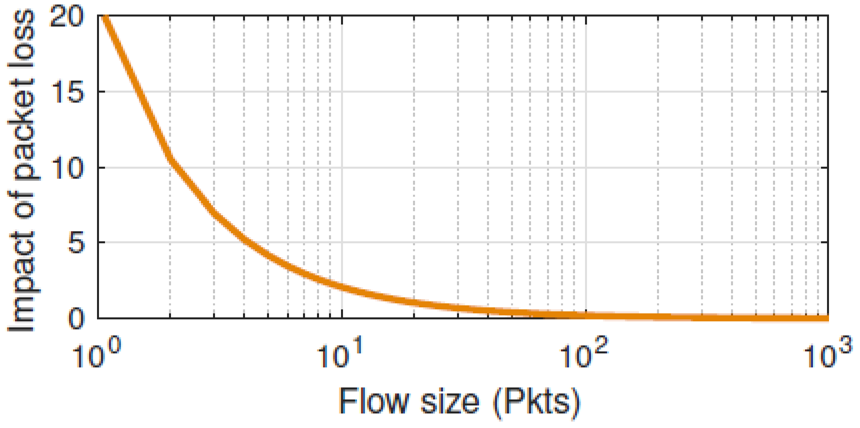
Design – Key Idea of LossPass

- LossPass passes packet loss of small flows to large flows to avoid timeout of small flows
- When buffer overflow occurs, the switch **evicts the buffered large flow packet** if the arriving packet is a small flow packet
- Key insight: as the flow size increases, the impact of packet loss on FCT decreases

Degree of the FCT reduction of small flows **>** Degree of the FCT increase of large flows

Gain **Loss**

It's a good deal!

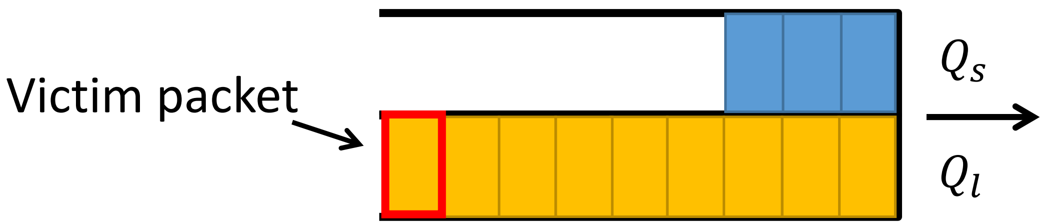


The impact of single timeout as the flow size increases

NetLab
Network and Security Research Lab.

Design – How to find the victim with low complexity?

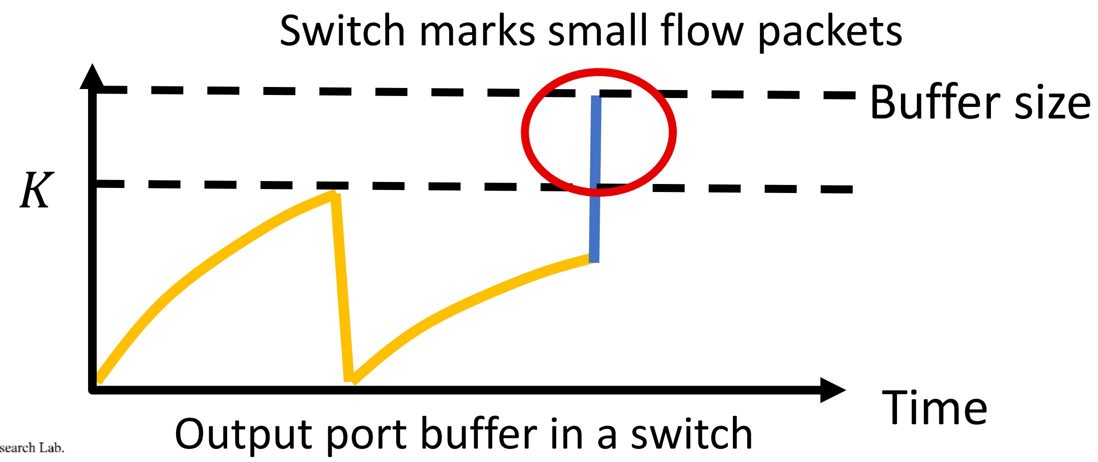
- A queue is an unsorted list consisting of a mixture of small and large flows
- Finding a large flow packet requires $O(n)$ complexity
- Our approach
  - Leverages two service queues in the port
    - Assigns different queues Q_s and Q_l for small and large flows, respectively
  - We can find the victim packet directly by pointing to the tail packet at Q_l
  - Switch can classify packets with DSCP field in the IP header
    - DSCP values can be tagged at end-hosts

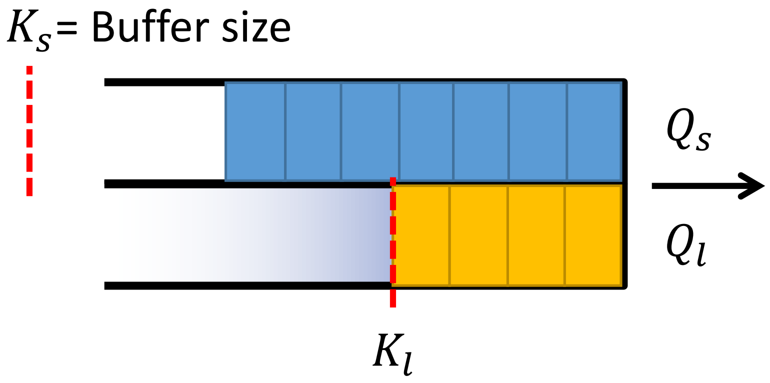


NetLab
Network and Security Research Lab.

Design – How can we use LossPass with ECN?

- ECN tries to decrease sending rates of small flows since the standard per-port ECN marking regards microbursts as the cause of congestion
- Our approach
  - Selective ECN marking by leveraging per-queue ECN marking
  - K_l=recommended value, K_s= Buffer size
  - Only marks large flow packets





Implementation

- Hardware implementation
  - Should be implemented at the end of ingress pipeline
  - Clock 1: checks DSCP value, packet size, and minimum buffer size
  - Clock 2: removes the victim by freeing the memory

- Software implementation
  - A software prototype as a Linux qdisc module on a server-emulated switch
  - `qdisc_dequeue_tail()` to evict the victim
  - `skb_peek_tail()` to obtain the tail packet size

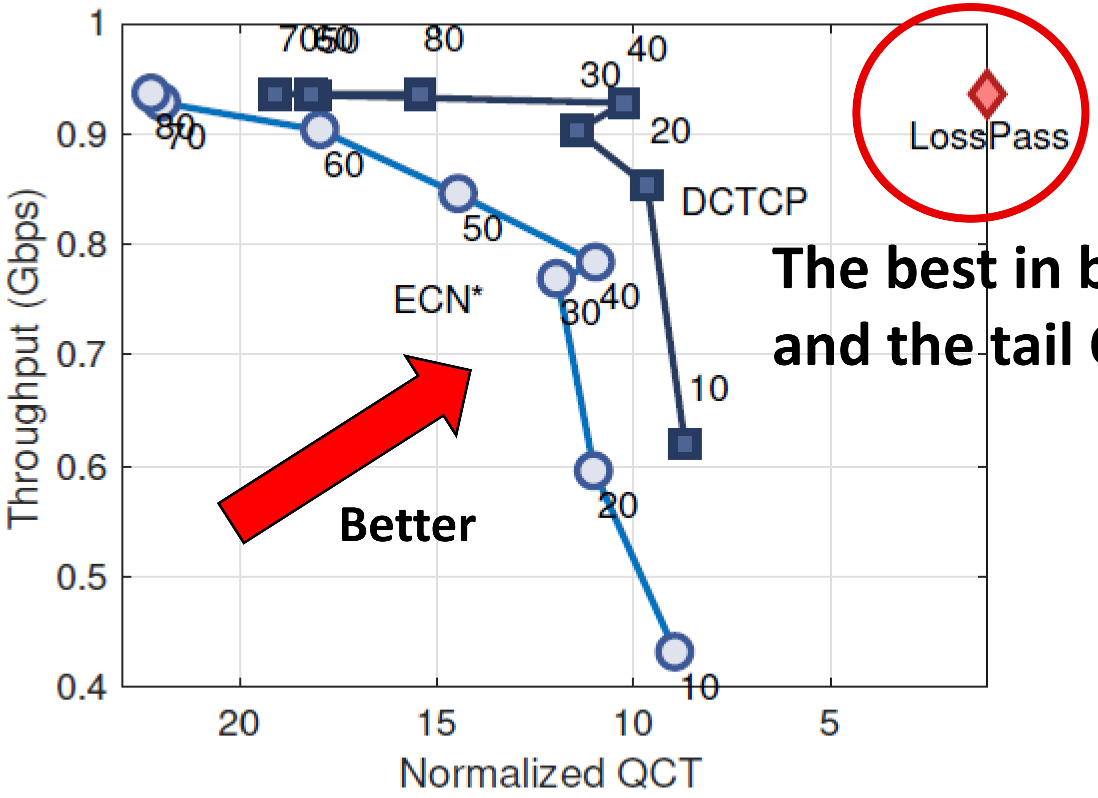
Evaluation

- Testbed experiments
  - 1Gbps in testbed experiments
  - Memcached KVS microbenchmarks
  - Realistic workloads from Microsoft data centers
    - Web search
    - Data mining
  - DCTCP by default
- Compared schemes
  - ECN with standard marking threshold
  - PIAS: the state-of-the-art flow scheduling solution (NSDI'15)

[PIAS] Wei Bai, Kai Chen, Hao Wang, Li Chen, Dongsu Han, and Chen Tian, "Information-agnostic flow scheduling for commodity data centers," in *Proc. of USENIX NSDI*, 2015.

Evaluation – Memcached Experiments

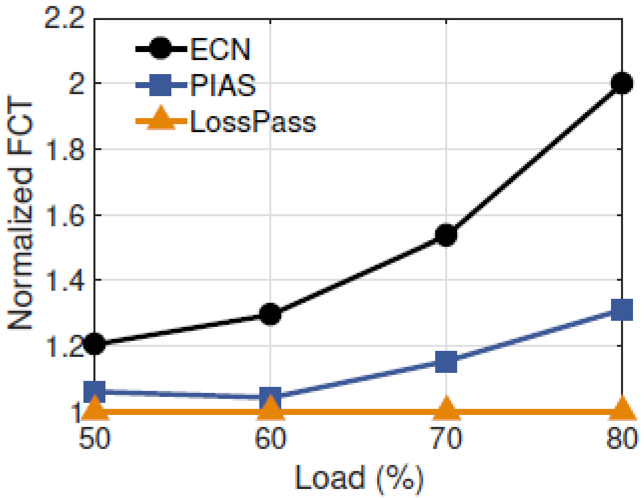
- Measures the aggregate throughput using `iperf`
- Measures the 99th percentile QCT of 1K memcached queries



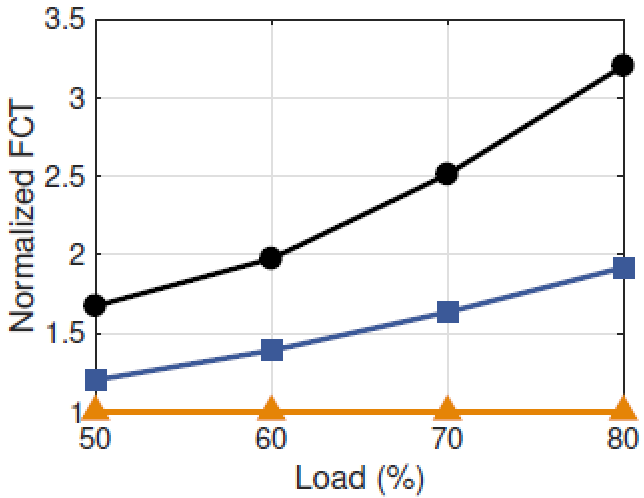
The best in both throughput and the tail QCT!

Evaluation – Workload-Driven Experiments

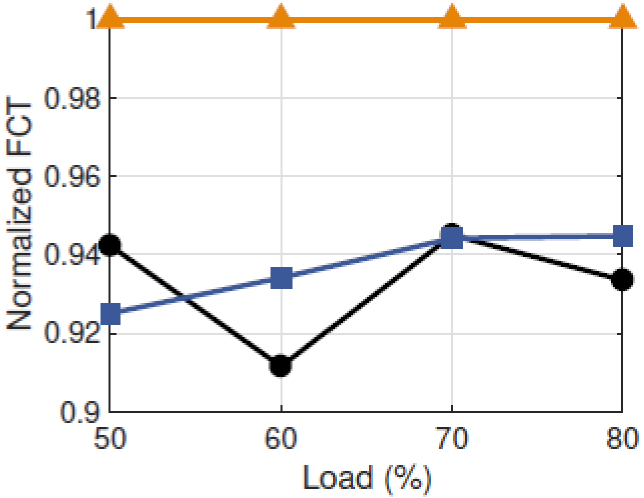
- Sends requests with data size generated by workloads
- 5K flows by varying traffic loads from 50% to 80%



(0, 100KB]: Avg



(0, 100KB]: 99th percentile



(10MB, ∞]: Avg

LossPass improves the FCT of small flows while degrading that of large flows slightly

NetLab Network and Security Research Lab.

Summary of LossPass

- **Problem:** how to absorb microbursts as many as possible with line-rate throughput?
- **LossPass**: a buffer sharing solution that implements packet eviction by addressing practical design issues
  - Finding the victim packet with low complexity by leveraging two priority queues
  - Providing ECN compatibility through selective ECN marking
- **Results**
  - Memcached experiments
    - Improves the 99th percentile QCT by up to 22.24x compared to ECN
    - Maintains line-rate throughput
  - Workload-driven experiments
    - Better than PIAS by up to 3.20x in the 99th percentile FCT