Optimizing Flow Completion Time via Adaptive Buffer Management in Data Center Networks

Sen Liu 1  
Yi Wang 3  
Xiang Lin 1  
Mohamed Adel Serhani 4  
Zehua Guo 2  
Yang Xu 1

1. Fudan University  
2. Beijing Institute of Technology  
3. Peng Cheng Lab  
4. United Arab Emirates University
DCNs ’s Challenge

- **Throughput-hungry** applications generate **long flows**, which always try to fill up the buffer of switches to achieve higher link utilization and thus introduce large queue.

- **Delay-sensitive** applications generate **short flows**, which requires small queue at the buffer of switches for their low latency communications.

**Trade-off:** high throughput and low latency communications.
I. most of flows in DCNs are short flows which are less than 1MB;
II. short flows are transmitted in a high concurrency and high burst manner;
III. the transmission of short flows shows obvious ON/OFF pattern.
Existing Schemes

(a). Single Queue

(b). Priority Queue

(c). Shared Priority Queue
Motivation Summary

- Most of the flows in DCNs are short flows
  High concurrency, high burstiness, ON/OFF pattern

- Long flows always try to fill up the buffer
  Large queueing delay and massive packet drops

- The current buffer management can not handle the ON/OFF traffic pattern well
  Poor performance of short flows
Design

Achieving efficient buffer management

- Boosting the performance of short flows in their ON periods
- Maintaining high link utilization for long flows in the OFF periods
Design

Achieving efficient buffer management

- Boosting the performance of short flows in their ON periods
- Maintaining high link utilization for long flows in the OFF periods

Cut-in Queue / CQ
Design

Achieving efficient buffer management

Cut-in Queue / CQ

Cut-in
the packets of short flows are enqueued ahead at switches to avoid massive queueing delay

Eviction
the enqueued packets of long flows should back off or dropped to leave switch buffer available for short flows
Design

① back off
② drop  Eviction
③ cut in  Cut-in Queue / CQ

(a). when packet of short flows coming
(b). step 1
(c). step 2
(d). step 3
Most of the current commercial switches are FIFO queues, which do not directly implement the above operations (back off and cut in).
Design --- CQ Architecture

Judging Function
identifying which type the incoming packet belongs to

Pulling Function
deciding whether packets in Back-off Queue should be pushed into Tx Queue

CQ overview
IMPLEMENTATION

Structure of CQ prototype in DPDK

More details in the paper
Testbed Evaluation

CQ prototype switch
Four Lenovo ThinkCentre K70 (Ubuntu 16.04)
A small many-to-one topology (10Gbps)
Testbed Evaluation

Avoid most of the packet drops

Reduce the FCT by up to 70% - short flow

Achieve high throughput - long flow
Testbed Evaluation

CQ achieves the lowest FCT
Simulation Evaluation

Simulator: NS2
Comparison: RED, PIAS, pFabric
Topology: Leaf-Spine

More results in the paper
Conclusion

◆ We reveal that the ON/OFF traffic pattern is common in DCNs.

◆ Current solutions do not consider this pattern and result in sub-optimal performance.

◆ We propose CQ, in which the short flow in ON periods can cut-into the head of enqueued long flow's packets, resulting in shorter queueing delay, while long flow can still leverage the whole buffer when short flows are in OFF period.
THANKS
Q&A