Protocol-Independent Service Queue Isolation for Multi-Queue Data Centers

Gyuyeong Kim and Wonjun Lee

Network and Security Research Lab. (NetLab)

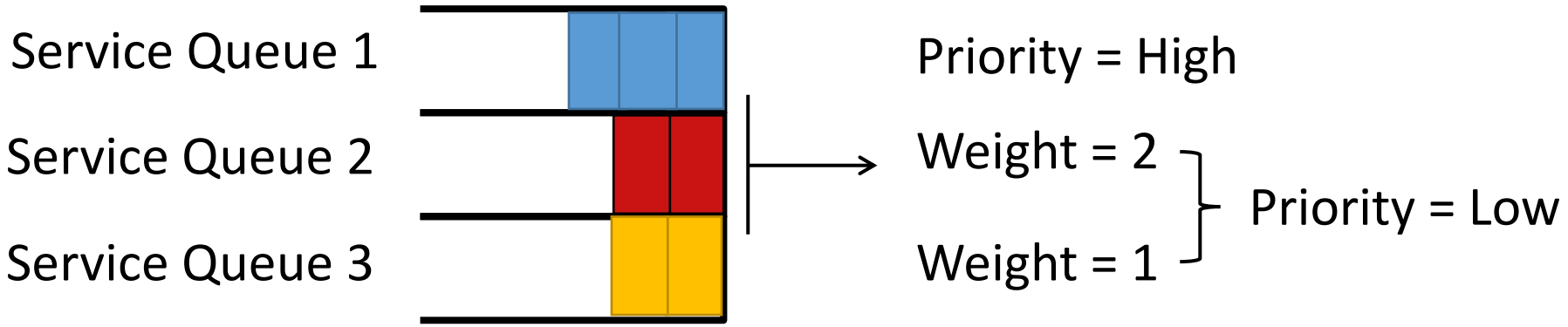
School of Cybersecurity

Korea University, Republic of Korea

IEEE ICDCS 2020, Singapore (Virtual), December 2020

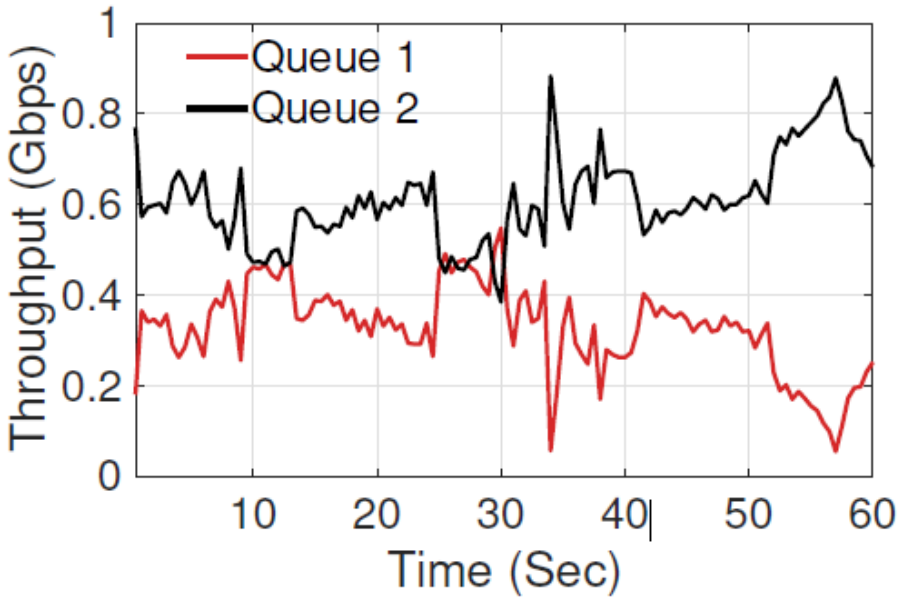
Motivation – Service Queues in Switches

- Modern cloud services have diverse network requirements
- Operators leverage service queues in switch ports to enforce network policy
  - Weighted fair schedulers (e.g. WRR, DRR) for isolation
  - SPQ scheduler to prioritize latency-sensitive flows

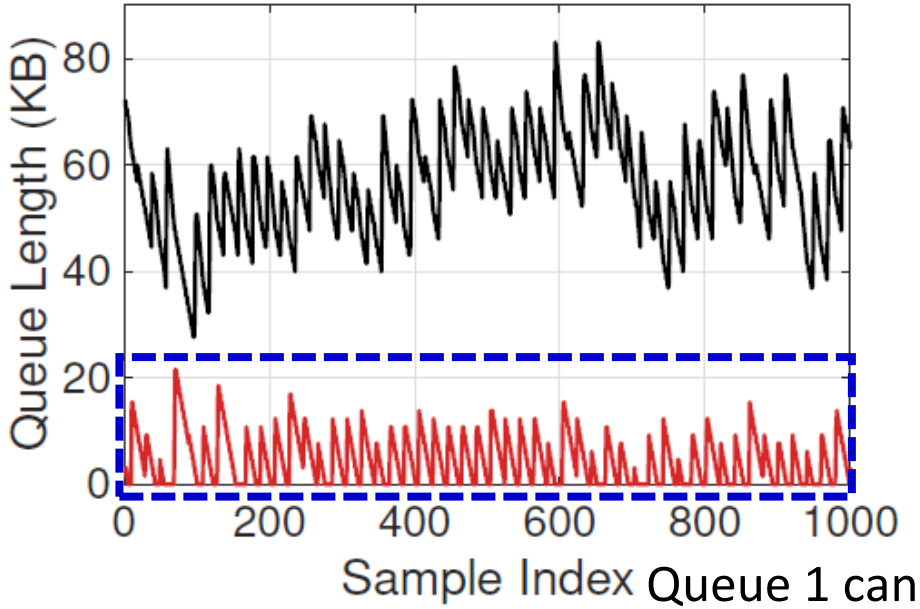


Motivation – Policy Violation by Unfair Buffer Sharing

- Aggressive queues monopolize the buffer, leading to network policy violation



Per-queue throughput



Queue length

Queue 1 cannot occupy enough buffer space

8 flows of queue 1 vs. 32 flows of queue 2

Motivation – Assumption of Existing Works

- Recent solutions leverage the power of ECN
  - MQ-ECN [NSDI'16], TCN [CoNEXT'16], PMSB [ICDCS'18]
  - Performs ECN marking to limit buffer occupancy of service queues
  - Assumption: all end-hosts use ECN-based transport protocols
    - Requirements: ECN-capable switches + ECN-enabled end-hosts

→ Fundamental dependency on ECN-based protocols (e.g. DCTCP)

Motivation – Why Protocol Dependency Matters?

- End-hosts cannot adapt to the advance in transport protocols
- Better non-ECN transport protocols have been proposed
  - **Delay-based protocols:** DX [ATC'15], TIMELY [SIGCOMM' 15]
  - **Credit-based protocol:** ExpressPass [SIGCOMM'17]
  - **INT-based protocol:** HPCC [SIGCOMM'19]

Q:How to isolate service queues in switch ports without dependency on transport protocols?

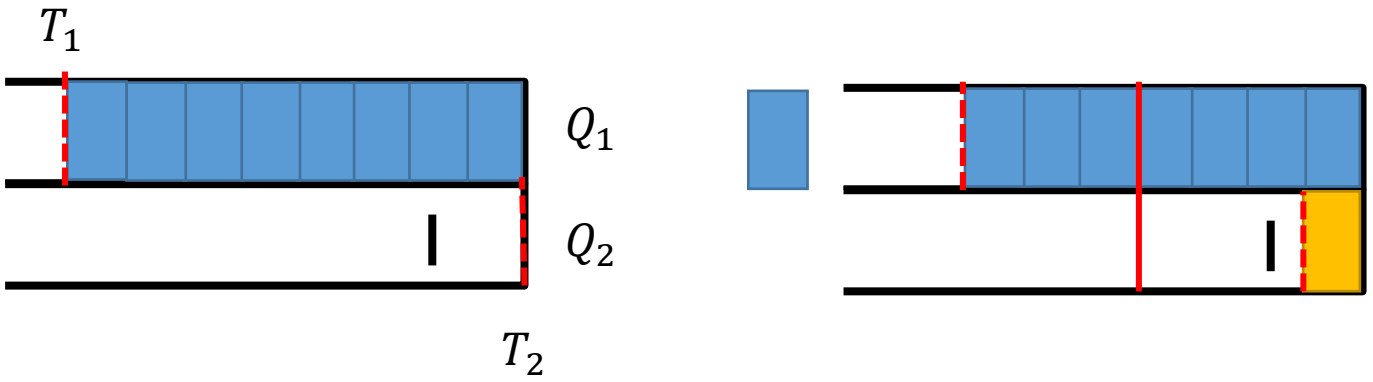
Design – Key Idea of DynaQ

- **DynaQ:** the first protocol-independent multi-queue management scheme

- Design guideline derived from the best-effort and Per-Queue Limit (PQL)
  - **Work conservation:** a service queue must be able to occupy the buffer larger than or equal to the BDP if there is free space in the port buffer
  - **Weighted fair sharing:** a service queue must be able to occupy buffer space larger than or equal to the weighted BDP regardless of other service queues
  - **We can meet only one of the two requirements at a time without dynamic multi-queue management**

Design – Key Idea of DynaQ

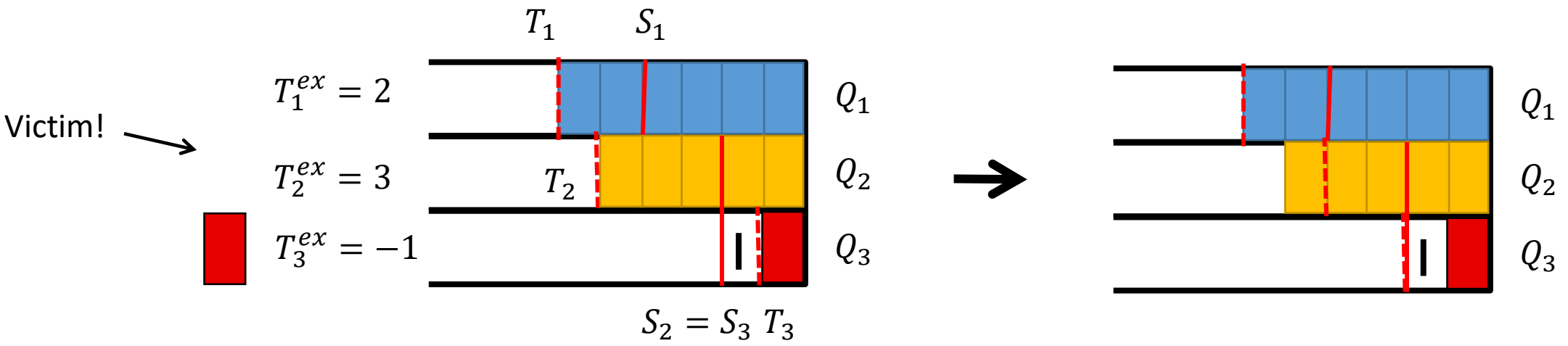
- Dynamically adjusts packet dropping threshold T_i every packet arrival
  - Allows a single queue to occupy free buffer space
  - Prevents the queue from taking the buffer of unsatisfied active queues



T_i: packet dropping threshold of queue i

Design – Victim Queue Selection

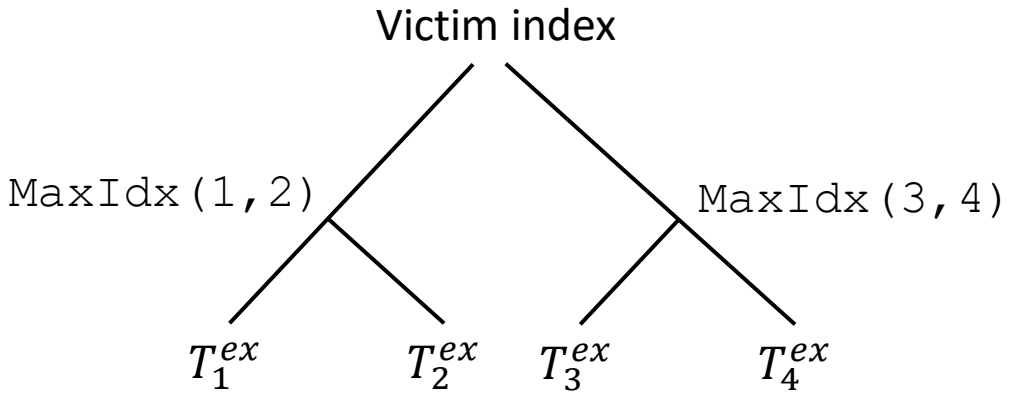
- Switch selects a service queue with the largest extra buffer size as the victim queue
  - T_i^{ex}: extra buffer size of queue i , $T_i - S_i$
  - S_i: satisfaction thresholod of queue i



Design – Victim Queue Search without Loops

- Finding the victim queue can be done through linear search
  - Unfortunately, switching ASICs prevent loop operations to guarantee a deterministic processing delay

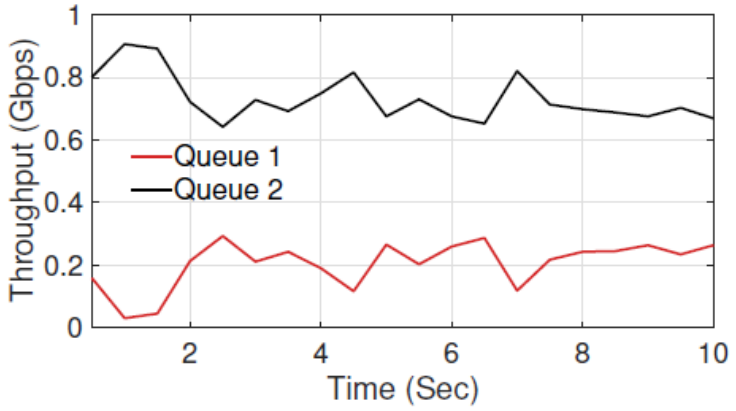
- DynaQ uses binary search with $O(\log n)$ complexity
  - Victim queue index = `MaxIdx(MaxIdx(1,2),MaxIdx(3,4))`



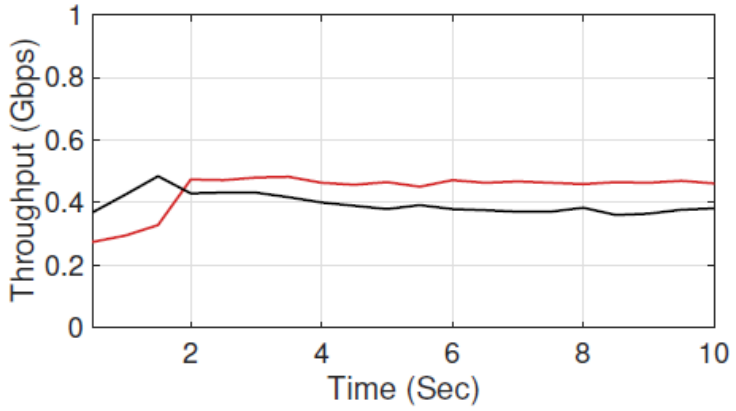
Evaluation

- Software implementation
  - A software prototype as a Linux qdisc module on a server-emulated switch
- Testbed setup
  - 5 servers connected to a server-emulated switch
  - Switch is with two Intel I350-T4 v2 1GbE NICs
  - Emulates Broadcom 56538 ASIC with 85KB per-port buffer
- ns-2 simulation setup
  - Broadcom Trident+ ASIC with 10Gbps links and 192KB port buffers
  - Broadcom Trident 3 ASICs with 100Gbps links and 1MB port buffers
- Compared schemes
  - BestEffort, PQL, PMSB, and TCN

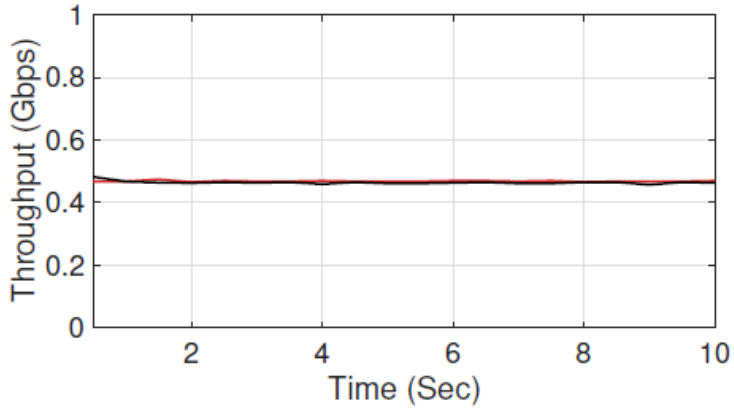
Evaluation – Convergence and Queue Evolution



BestEffort

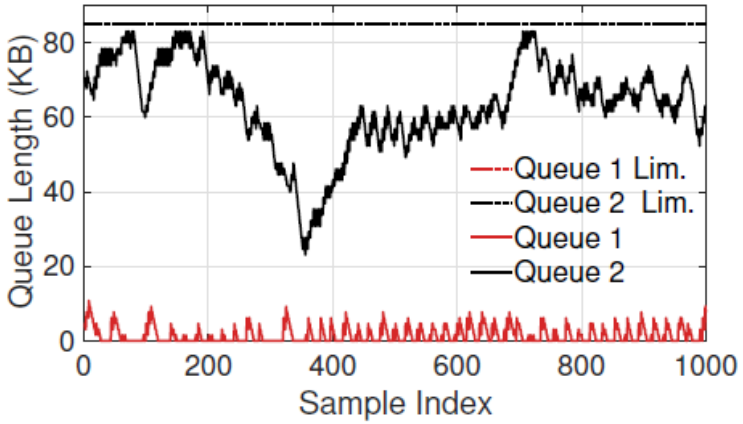


PQL

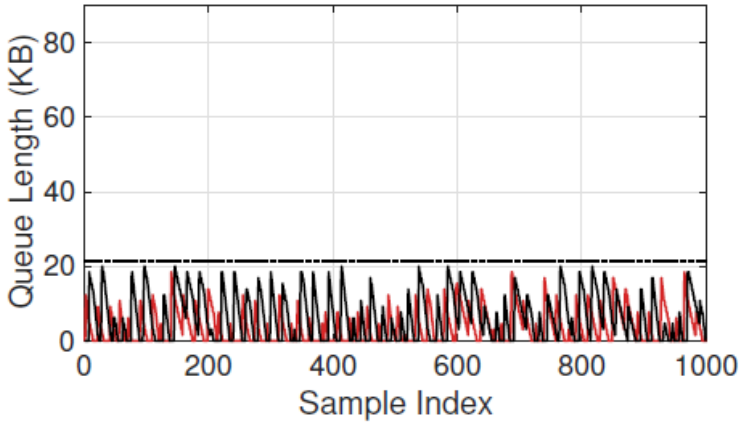


DynaQ

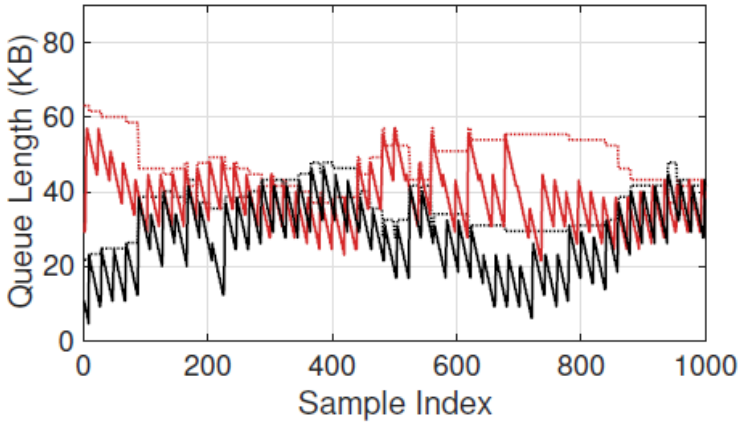
Throughput convergence of two active queues



BestEffort



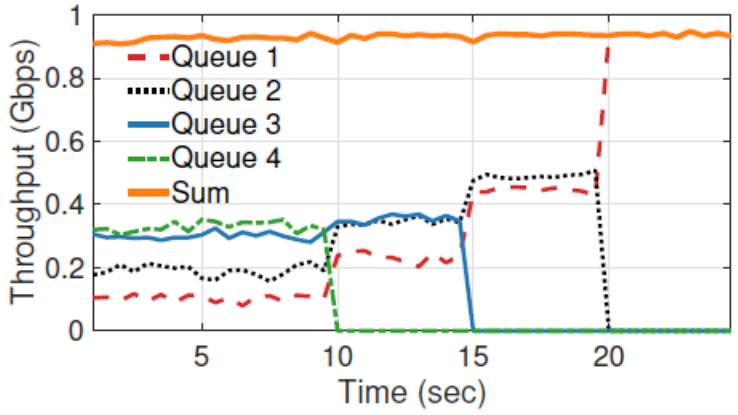
PQL



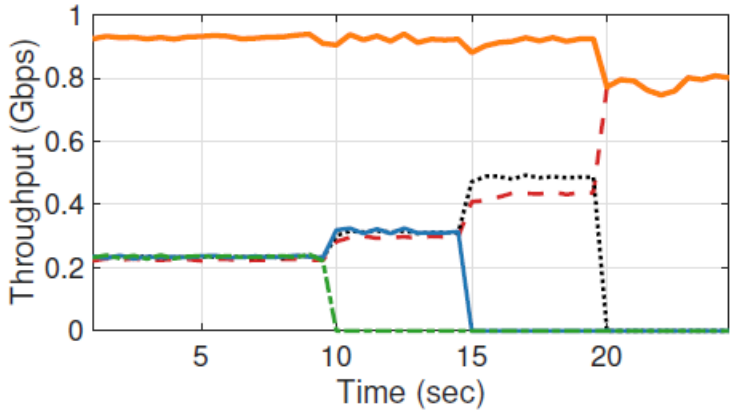
DynaQ

Queue length evolution of two active queues

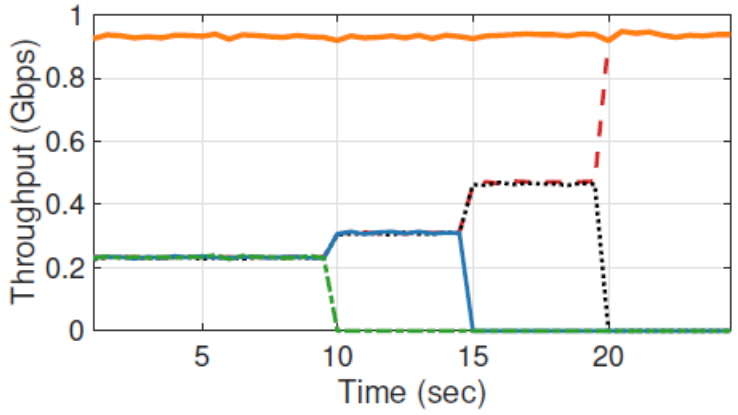
Evaluation – Bandwidth Sharing



BestEffort

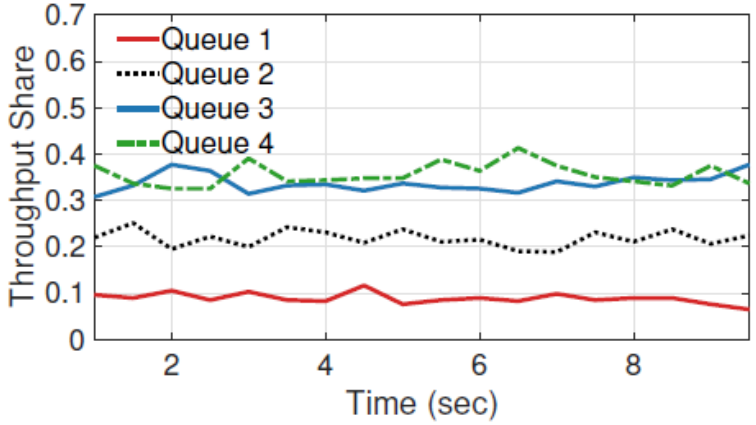


PQL

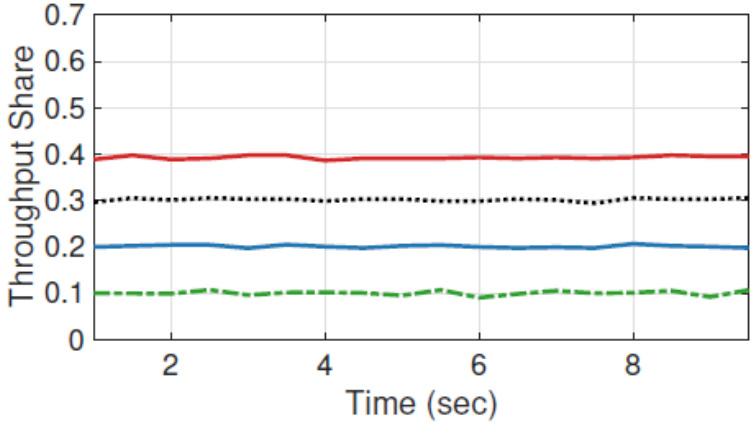


DynaQ

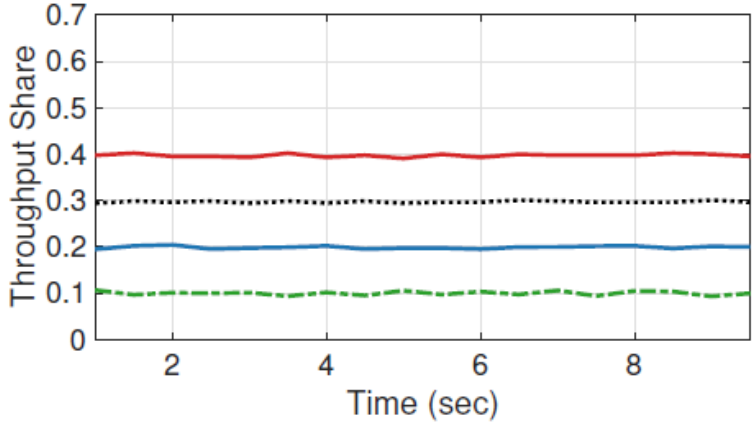
Bandwidth sharing with equal weights



BestEffort



PQL

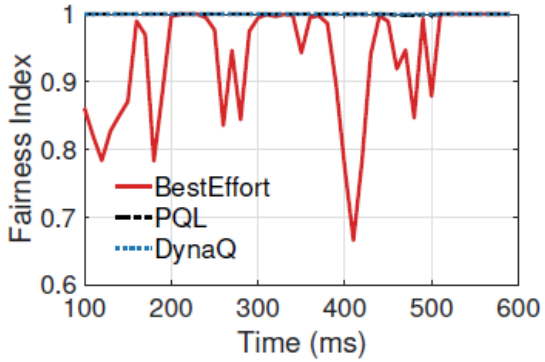


DynaQ

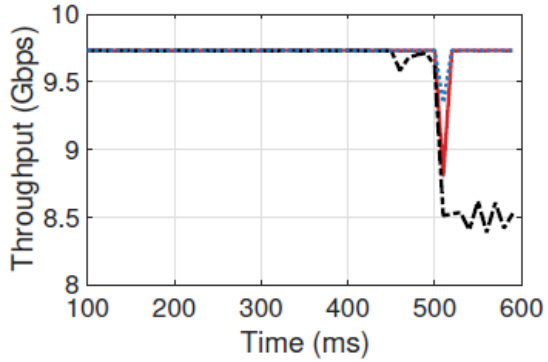
Bandwidth sharing with queue weights of 4:3:2:1

Evaluation – Impact of Link Capacity

- DynaQ is robust to link capacity
  - BestEffort fails to achieve fair sharing
  - PQL cannot maintain line-rate throughput

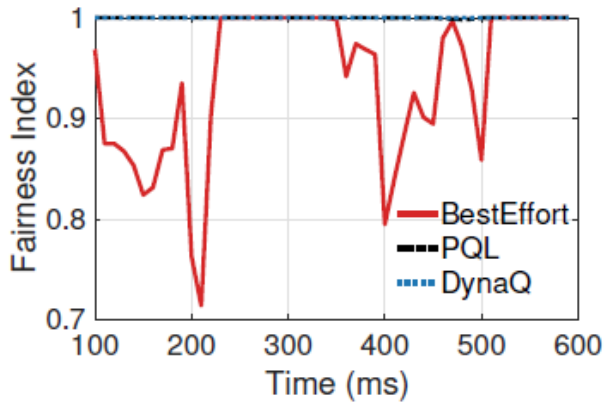


Fairness index

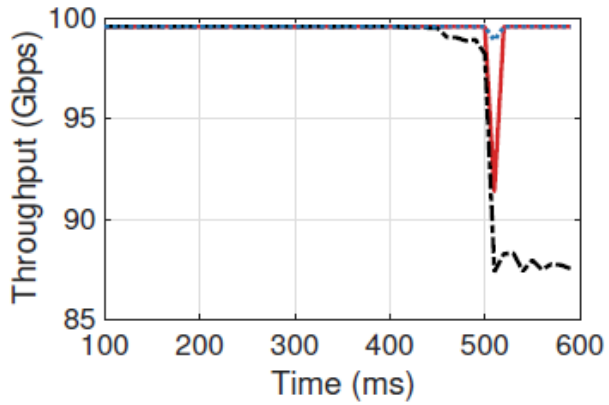


Aggregate throughput

10Gbps links



Fairness index



Aggregate throughput

100Gbps links

Conclusion

- **Problem:** how to isolate service queues without protocol dependency?

- **DynaQ**: a multi-queue management solution that adjusts packet dropping thresholds dynamically
  - **Key idea:** allows a queue to occupy free buffer space but protects unsatisfied active queues

- **Results**
  - Preserves weighted fair sharing and work conservation at the same time
  - Robust to link capacity