MBECN: Enabling ECN with Micro-burst in Multi-queue Datacenter

Kexi Kang

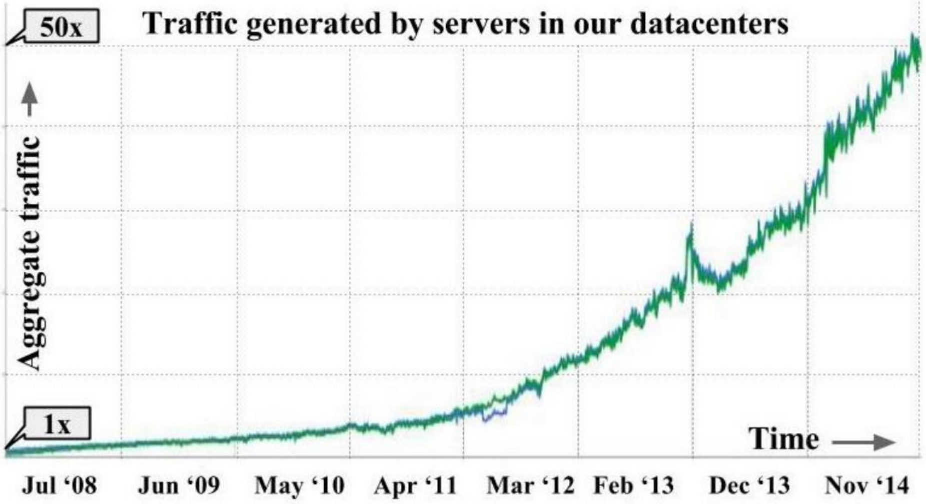
Southeast University

Outline

- Background & Motivation
- Analysis
- Solution
- Evaluation
- Conclusion

Data Center Network (DCN)

- Intra DC
  - – Distributed applications
    - High throughput & Low latency
  - – Growing traffic

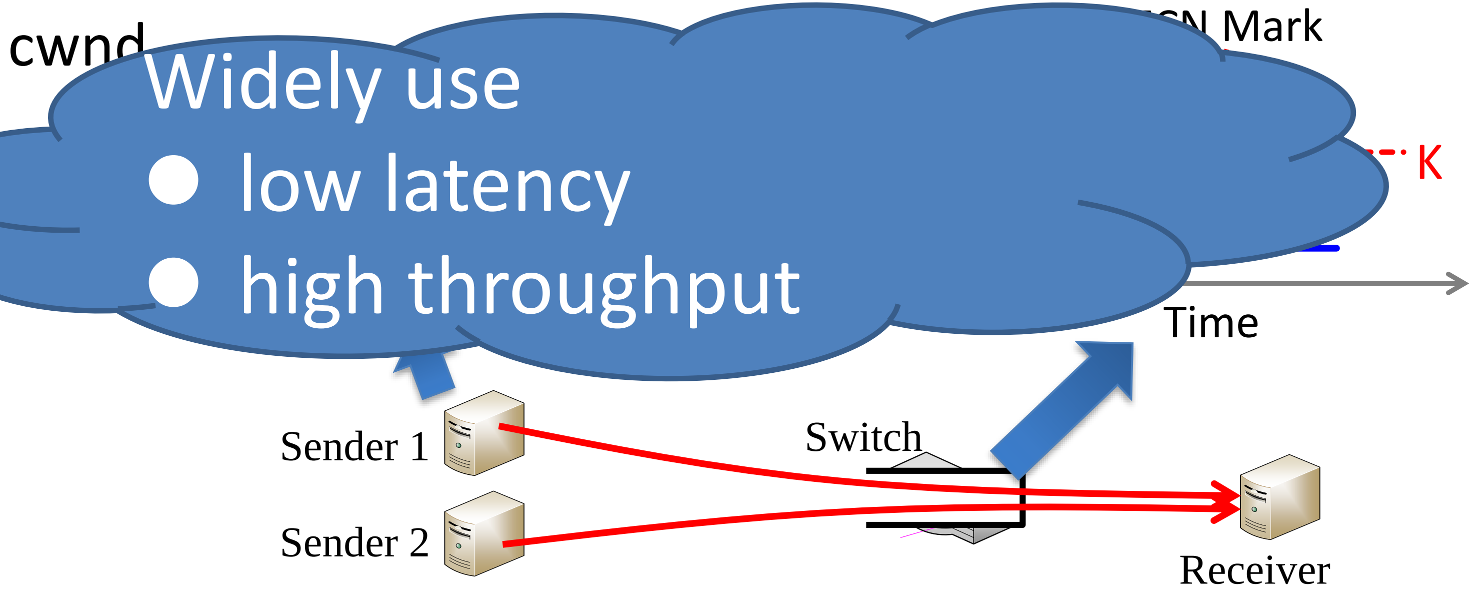


Multi-queue Data center

- Why Multi-queue?
  - Multi-queue Switch in industry
  - Multi-queue to isolate cloud services
- Main principle
  - Weighted Fair Share
- Main method
  - AQM (Active Queue Management)

Standard ECN marking in DCNs

- Standard ECN marking
  - DCTCP, ECN*, DCQCN, ……
  - Single ECN threshold, Instant queue length
    - If Qlen > K, mark packets with ECN
    - Senders slow down according to ECN feedbacks



Widely use

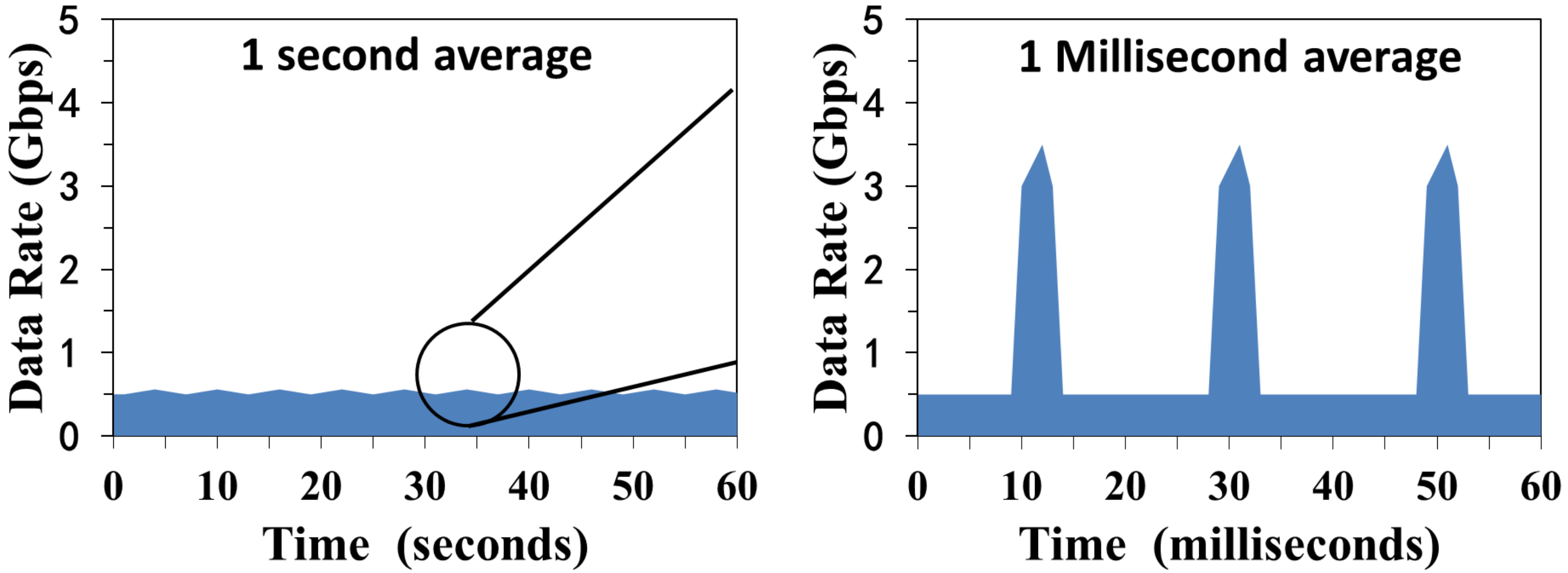
- low latency
- high throughput

Micro-burst in DCN

- Reducing CPU overhead: batching
  - **Large Segment Offload: TSO, GSO**
  - Receive Side Offload: RSC, LRO, GRO
  - Interrupt Coalescing (IC)
  - Jumbo Frame
  - …

Micro-burst in DCN

- One minute averages hide these short bursts.

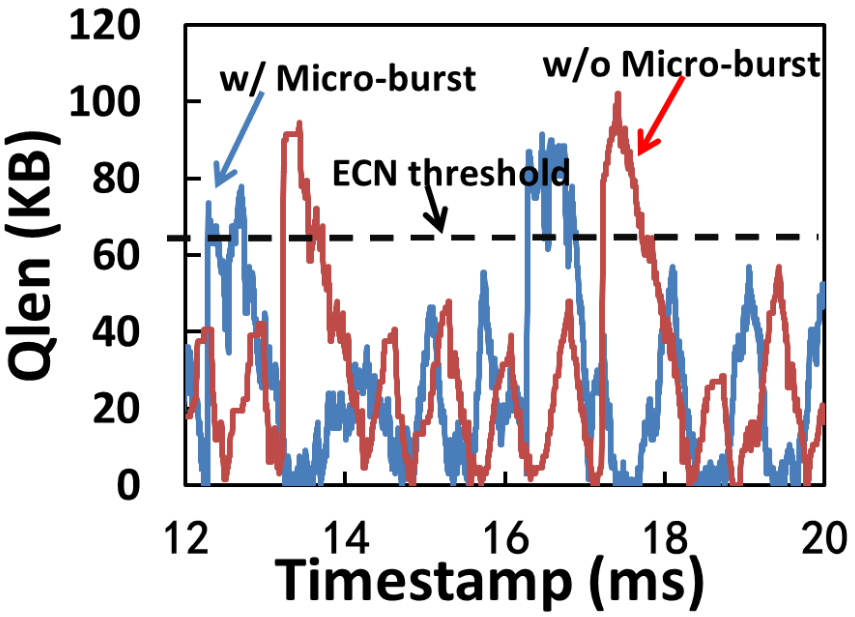


Outline

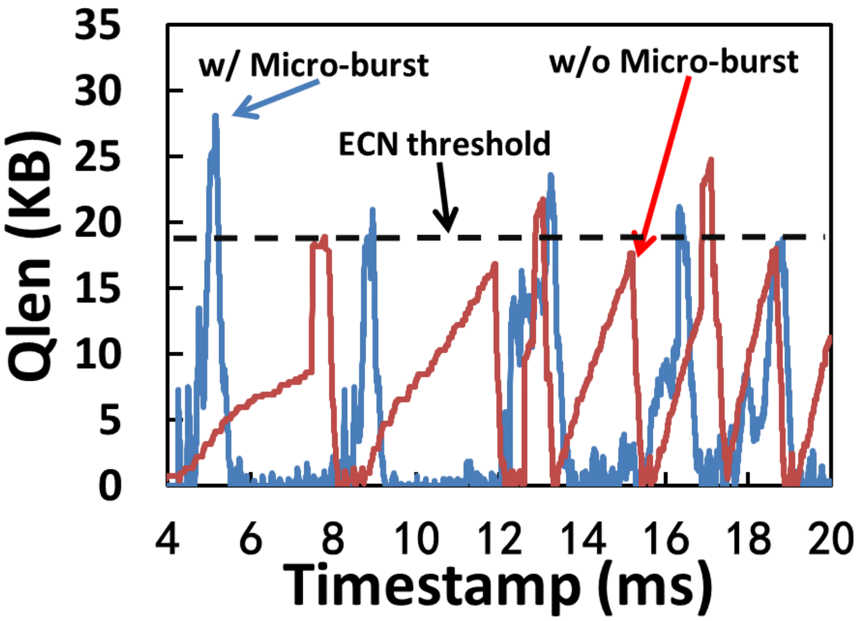
- Background & Motivation
- Analysis
- Solution
- Evaluation
- Conclusion

Effect of Micro-burst on Standard ECN

- Micro-burst causes serious **queue oscillations**



DCTCP

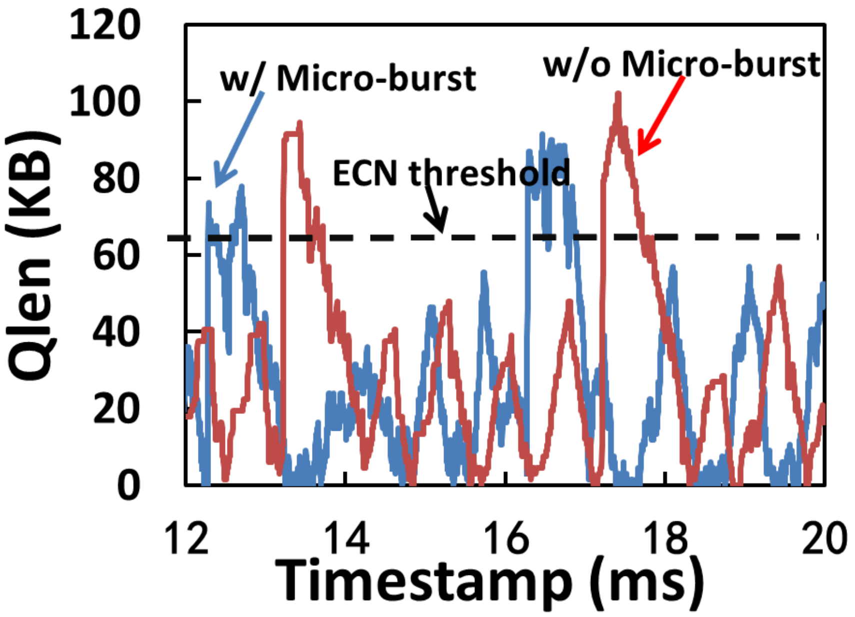


ECN*

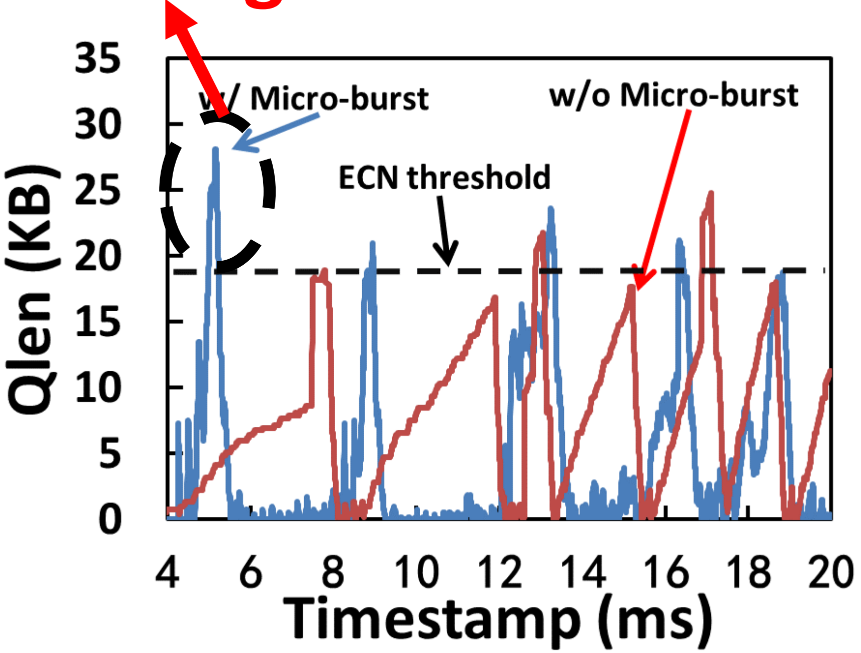
Effect of Micro-burst on Standard ECN

- Micro-burst causes serious **mismarkings**

mismarking



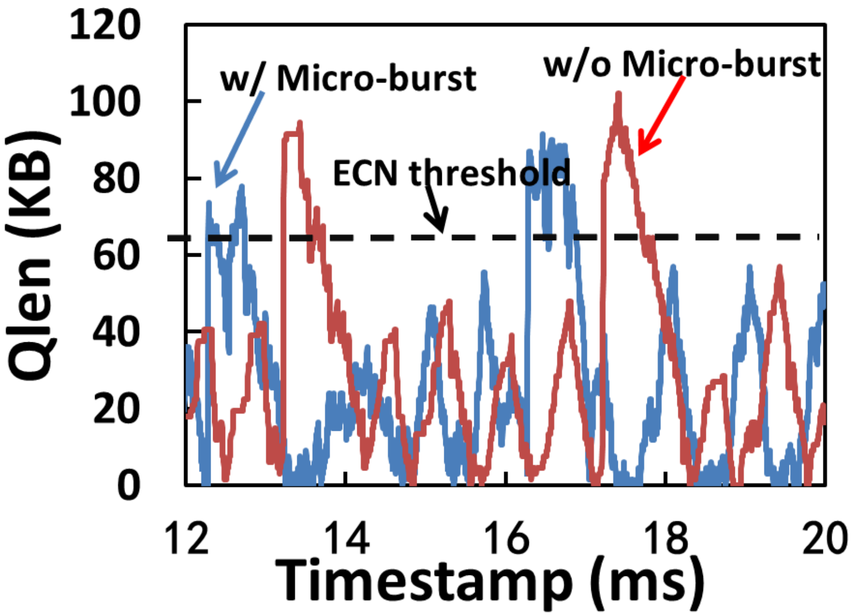
DCTCP



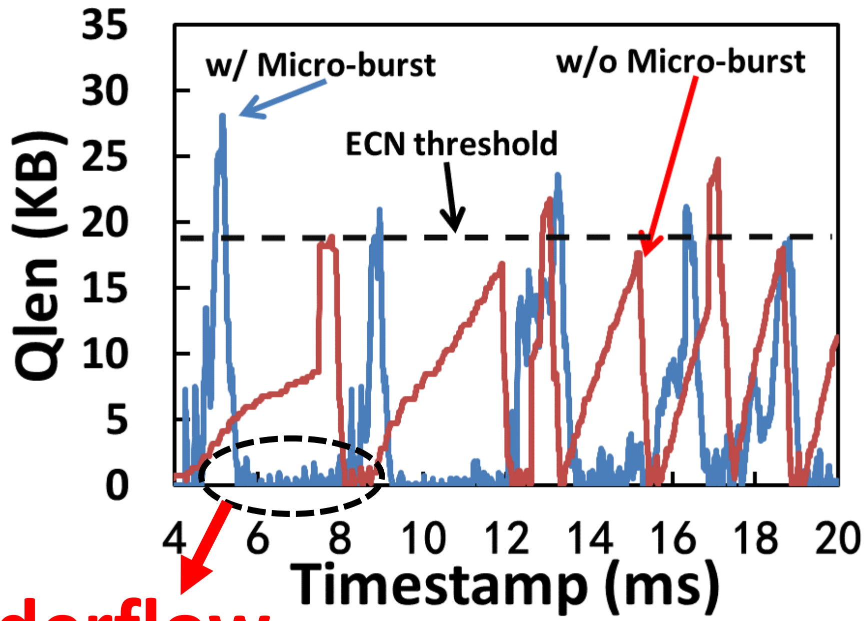
ECN*

Effect of Micro-burst on Standard ECN

- Micro-burst causes serious **buffer underflow**



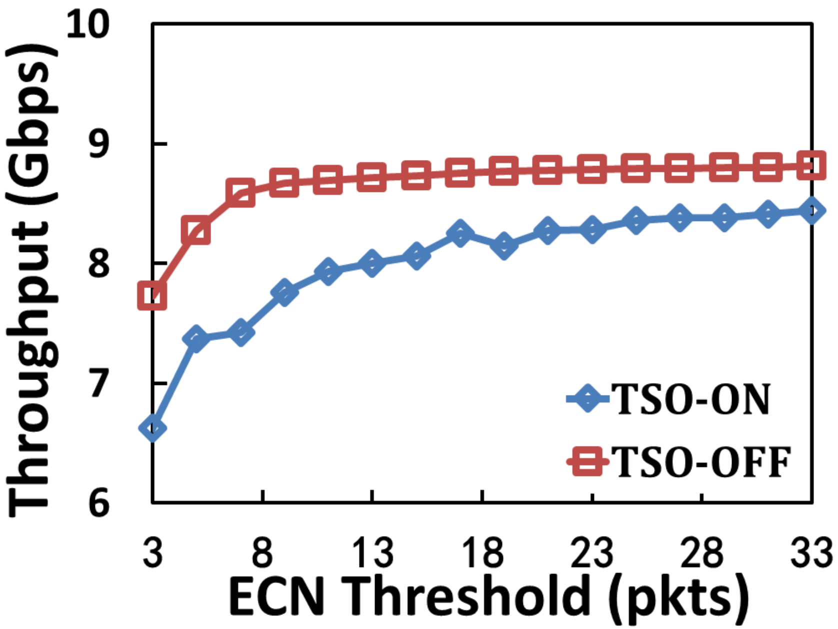
DCTCP



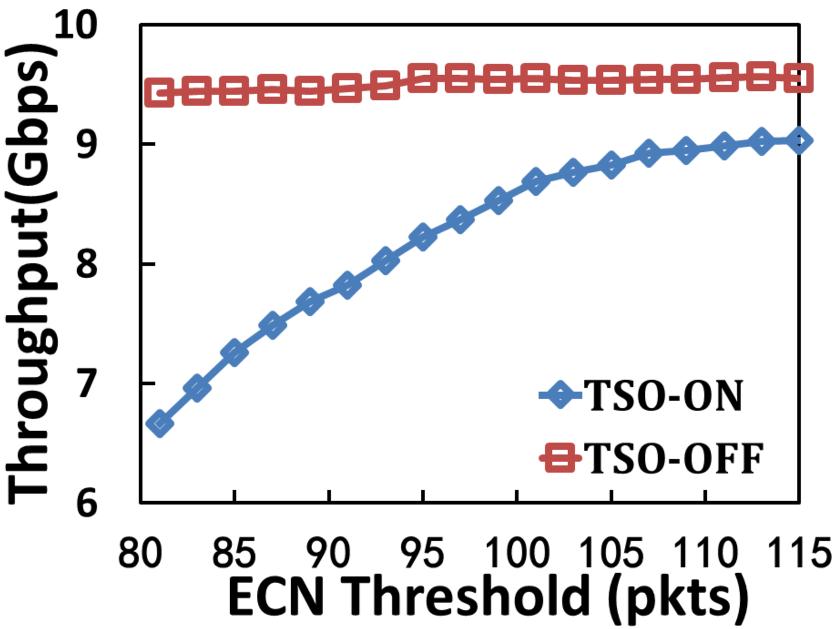
ECN*

Effect of Micro-burst on Standard ECN

- Serious buffer underflow results in **throughput loss**



DCTCP



ECN*

How to set ECN threshold

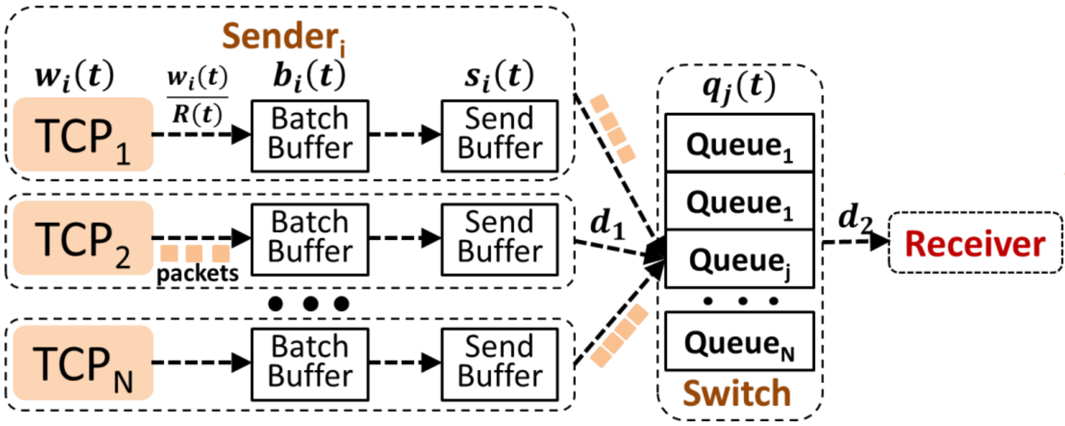
- Higher ECN threshold
  - Enough room to absorb micro-burst
- Dynamic ECN threshold
  - Adapt to dynamic network
  - Deal with packet backlog

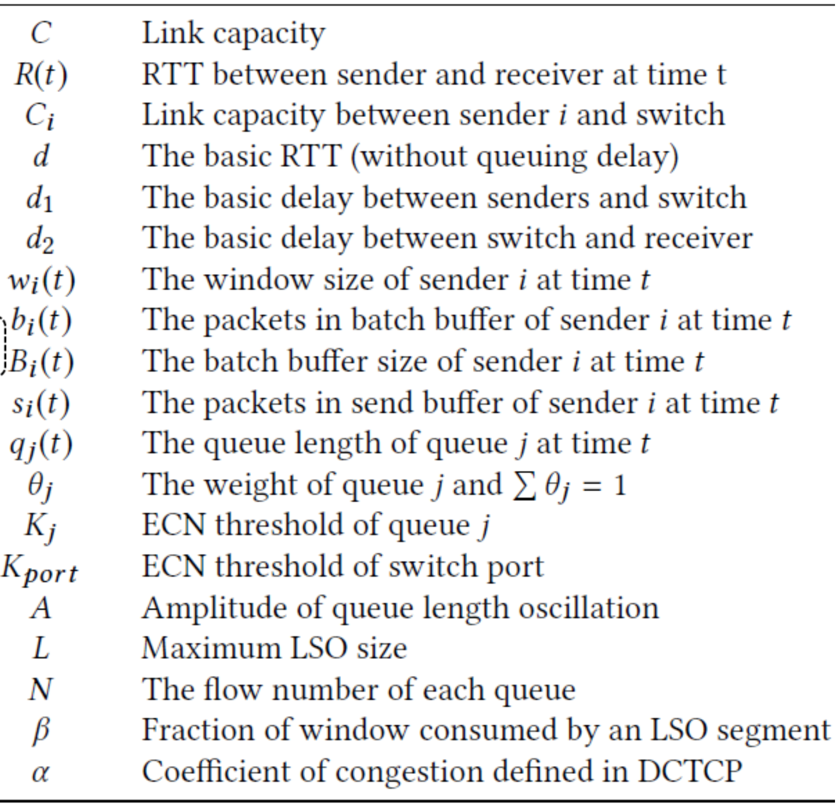
Outline

- Background & Motivation
- Analysis
- Solution
- Evaluation
- Conclusion

Steady-state Analysis

- Model





C	Link capacity
$R(t)$	RTT between sender and receiver at time t
C_i	Link capacity between sender i and switch
d	The basic RTT (without queuing delay)
d_1	The basic delay between senders and switch
d_2	The basic delay between switch and receiver
$w_i(t)$	The window size of sender i at time t
$b_i(t)$	The packets in batch buffer of sender i at time t
$B_i(t)$	The batch buffer size of sender i at time t
$s_i(t)$	The packets in send buffer of sender i at time t
$q_j(t)$	The queue length of queue j at time t
θ_j	The weight of queue j and $\sum \theta_j = 1$
K_j	ECN threshold of queue j
K_{port}	ECN threshold of switch port
A	Amplitude of queue length oscillation
L	Maximum LSO size
N	The flow number of each queue
β	Fraction of window consumed by an LSO segment
α	Coefficient of congestion defined in DCTCP

Steady-state Analysis

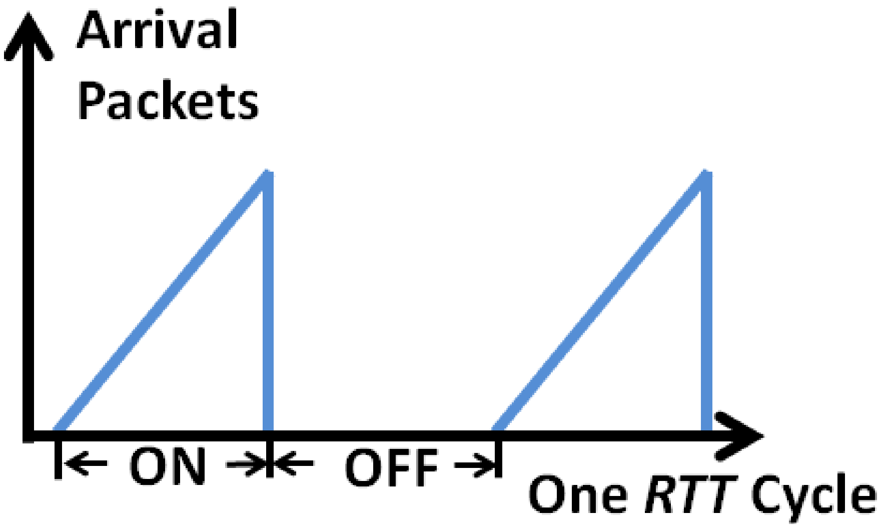
- Two basic constraints
  - $q_{min}^{j}(t) \leq K_j \leq q_{max}^{j}(t)$

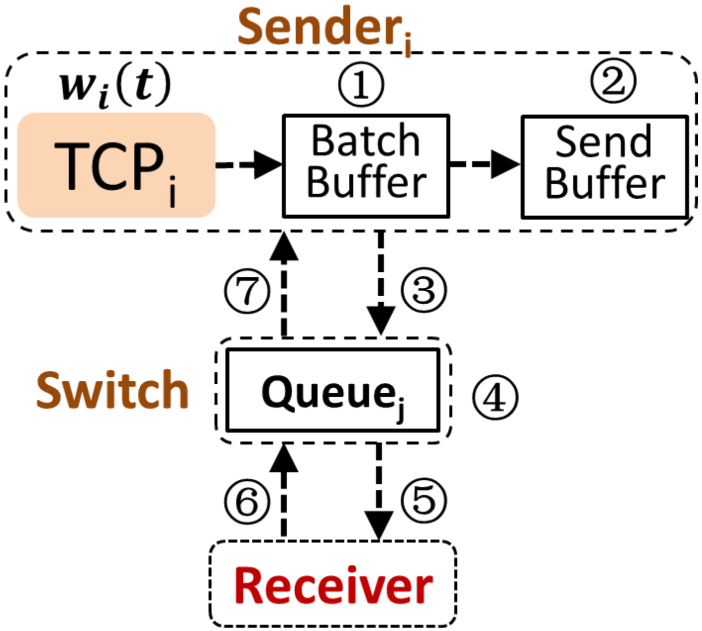
    Enough room to absorb micro-burst
  - $q_{min}^{j}(t) \geq 0$

    Avoid queue underflow

Steady-state Analysis

- Two lemmas
  - Packets distribution based on ON/OFF pattern





Steady-state Analysis

- Lemma 1

$$\boldsymbol{q^{j}_{min}(t) = Nw_i(t) - \theta_j Cd - NB_i(t)}$$

- Lemma 2

$$\boldsymbol{q^{j}_{max}(t) = Nw_i(t) - \theta_j Cd - (\theta_j \frac{C}{C_i} + 1)B_i(t)}$$

Steady-state Analysis

- ECN*
  - Based on two constraints and two lemmas

$$K_j^{bound} = \frac{(1+\beta)N-2\Phi\beta}{N(1-\beta)}\theta_j Cd$$

  - Where $\Phi = \theta_j \frac{C}{C_i} + 1$

Steady-state Analysis

- DCTCP
  - Based on two constraints and two lemmas

$$\boldsymbol{K_j^{bound}} = \begin{cases} \dfrac{(N-\Phi\beta)^2}{8\beta} + \dfrac{\beta(N-\Phi)}{N(1-\beta)}\theta_j Cd & if \;\; w^* \leq \dfrac{K_j + \theta_j Cd}{N(1-\beta)} \\ \dfrac{\Psi + \sqrt{\Psi^2 + 8\theta_j Cd\Psi}}{4} & if \;\; w^* > \dfrac{K_j + \theta_j Cd}{N(1-\beta)} \end{cases}$$

  - Where $\Phi = \theta_j \dfrac{C}{C_i} + 1$, $\Psi = N(N-\Phi)(1-\beta)$

$$w^* = \frac{N-\Phi\beta}{8\beta} + \frac{\theta_j Cd + K_j}{N-\Phi\beta}$$

MBECN—Threshold Baseline

- Base on ideal GPS model
  - When $\sum K_j^{bound} > K_{port}$

$$K_j^{baseline} = K_j^{bound} - \theta_j \times (\sum K_j^{bound} - K_{port})$$

  - When $\sum K_j^{bound} \leq K_{port}$

$$K_j^{baseline} = K_j^{bound} + \theta_j \times (K_{port} - \sum K_j^{bound})$$

MBECN—Threshold Baseline

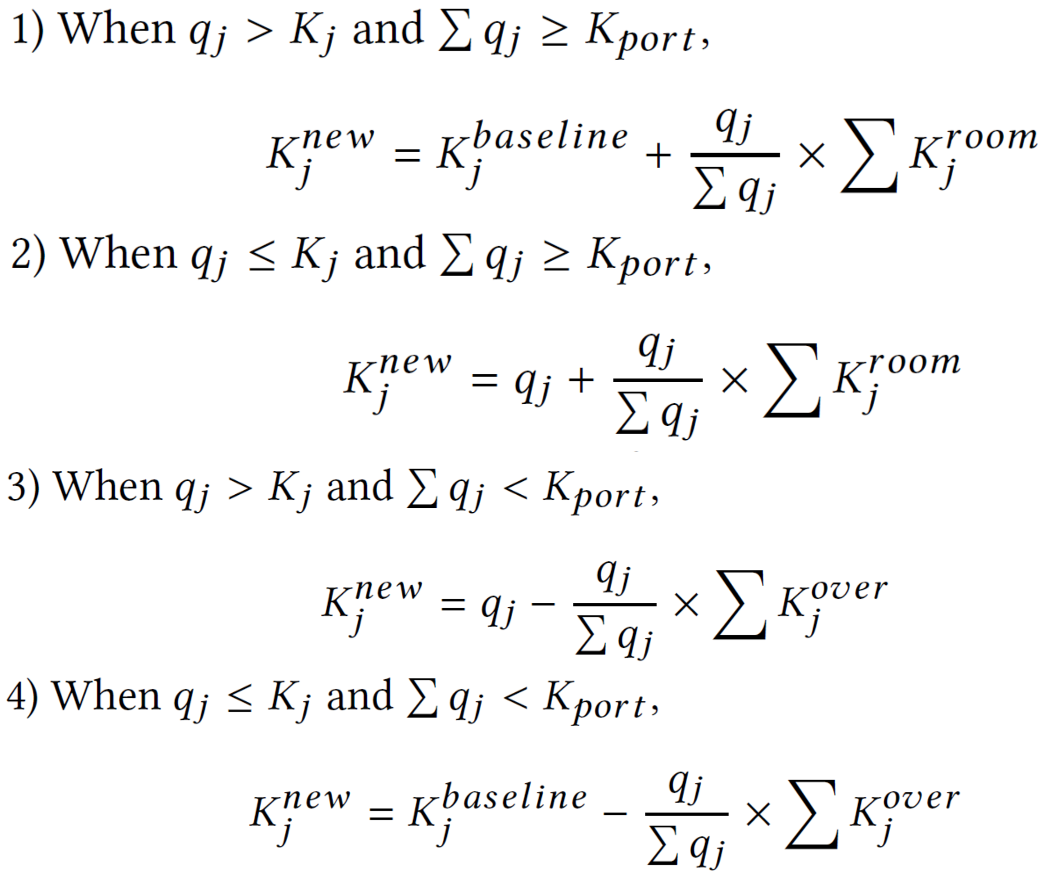
- Why need to tune the threshold?
  - Queue isolations
  - Avoid mismatch between input rate and output rate

MBECN—Dynamically Adjust

- Aim to fully utilize the room buffer
- Predefine：
  - K_j^{room}: The room buffer of $queue_j$
    If $q_j \geq K_j, K_j^{room} = 0$; If $q_j < K_j, K_j^{room} = K_j - q_j$
  - K_j^{over}: The part of $queue_j$ overflow
    If $q_j \geq K_j, K_j^{over} = q_j - K_j$; If $q_j < K_j, K_j^{over} = 0$

MBECN—Dynamically Adjust

- Heuristic algorithm



1) When $q_j > K_j$ and $\sum q_j \geq K_{port}$,

$$K_j^{new} = K_j^{baseline} + \frac{q_j}{\sum q_j} \times \sum K_j^{room}$$

2) When $q_j \leq K_j$ and $\sum q_j \geq K_{port}$,

$$K_j^{new} = q_j + \frac{q_j}{\sum q_j} \times \sum K_j^{room}$$

3) When $q_j > K_j$ and $\sum q_j < K_{port}$,

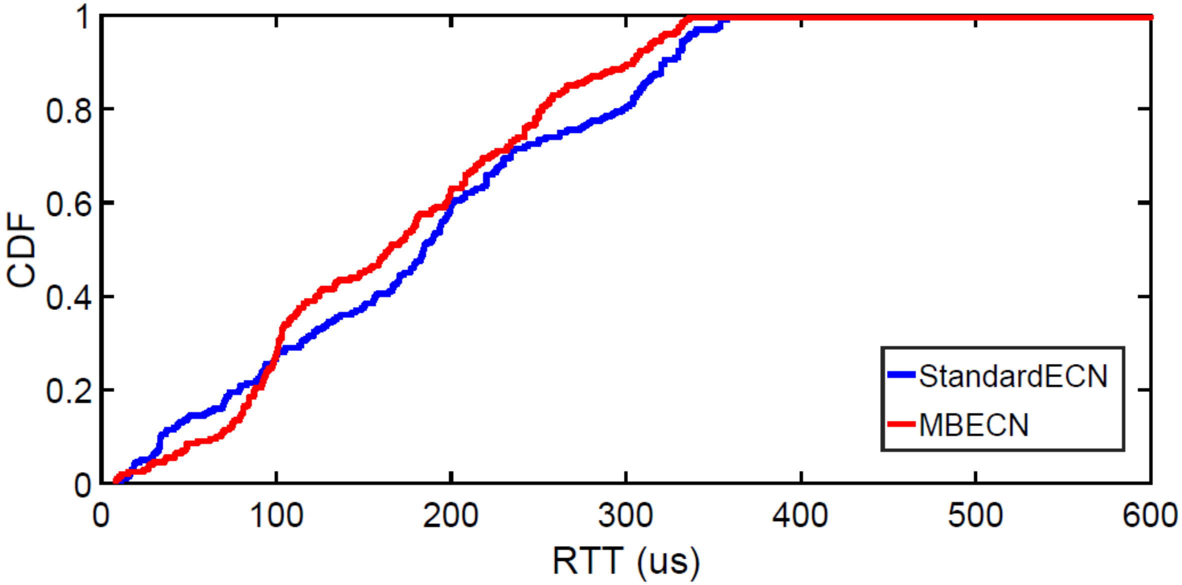
$$K_j^{new} = q_j - \frac{q_j}{\sum q_j} \times \sum K_j^{over}$$

4) When $q_j \leq K_j$ and $\sum q_j < K_{port}$,

$$K_j^{new} = K_j^{baseline} - \frac{q_j}{\sum q_j} \times \sum K_j^{over}$$

MBECN—Dynamically Adjust

- Could dynamical threshold bring extra latency?



MBECN—Dynamically Adjust

- Reasons:
  - – Expand threshold for high-load queues
    - More scheduling time or rounds
  - – Heuristic algorithm needs several CPU cycles

Outline

- Background & Motivation
- Analysis
- Solution
- Evaluation
- Conclusion

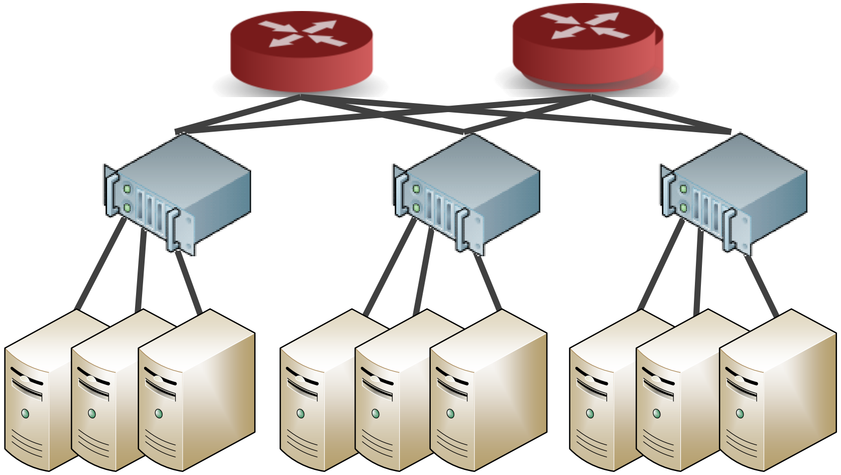
Evaluations

- Large-scale Simulations
  - – NS-2
  - – Realistic workload

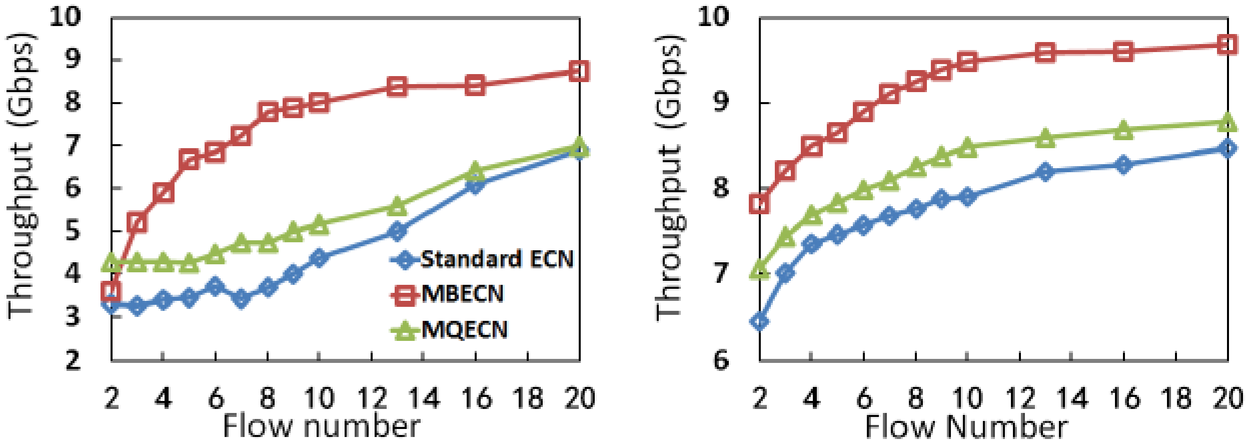
- Testbed Experiments
  - – Server-emulated Switch
  - – Realistic workload

Simulations

- Topology
  - 144 hosts, 12 leaf (ToR) switches and 6 spine (Core) switches.



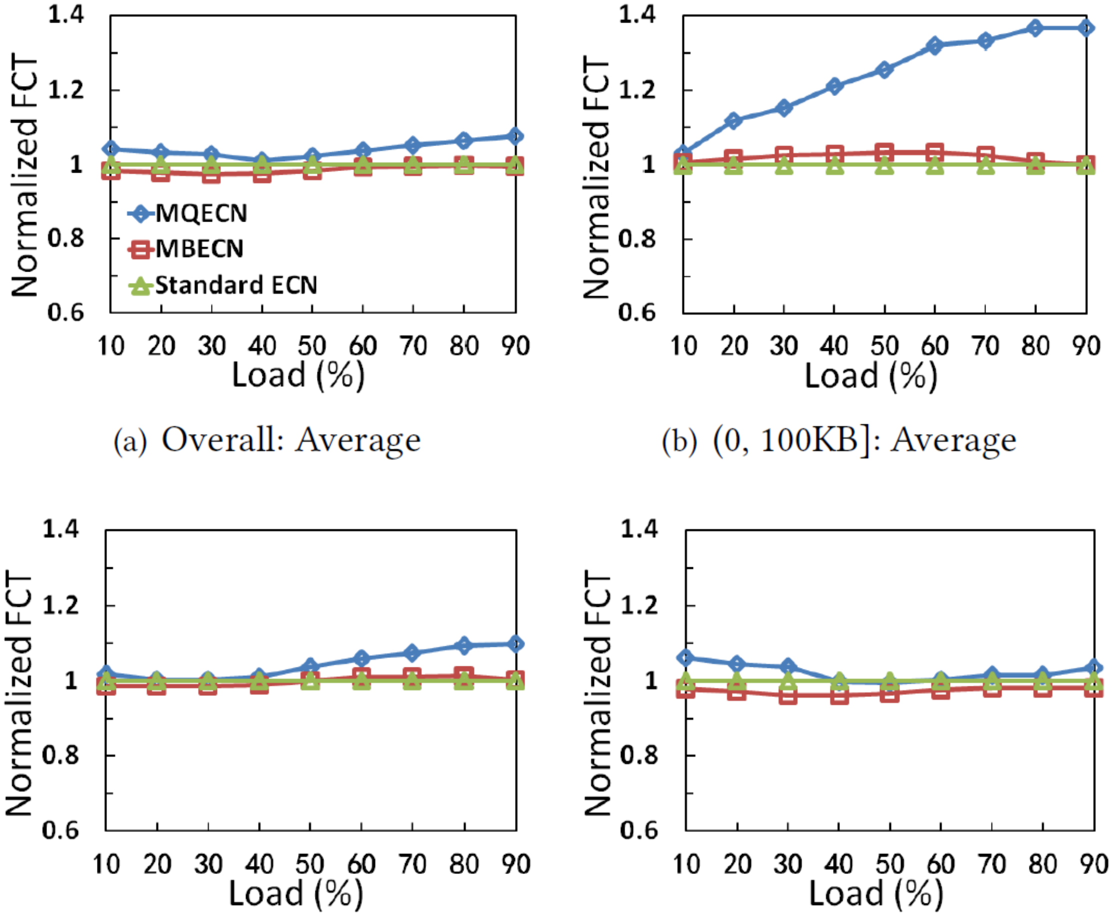
Simulations-Throughput



(a) Throughput with DCTCP and different flow number

(b) Throughput with ECN* and different flow number

Simulation-FCT with DCTCP



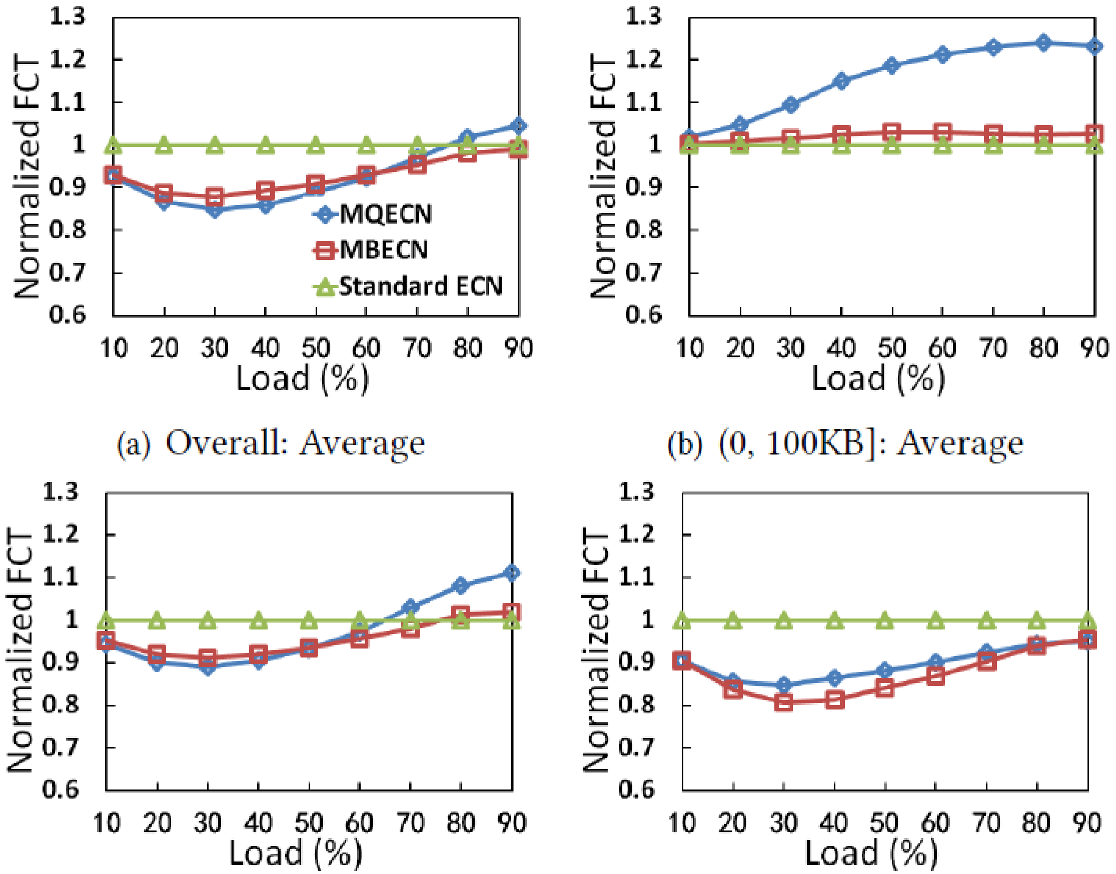
(a) Overall: Average

(b) (0, 100KB]: Average

(c) (100KB, 10MB]: Average

(d) (10MB, ∝): Average

Simulation-FCT with ECN*



(a) Overall: Average

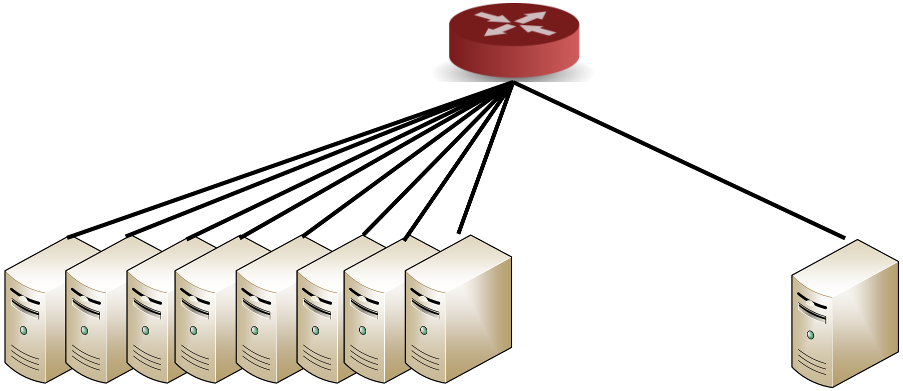
(b) (0, 100KB]: Average

(c) (100KB, 10MB]: Average

(d) (10MB, ∝): Average

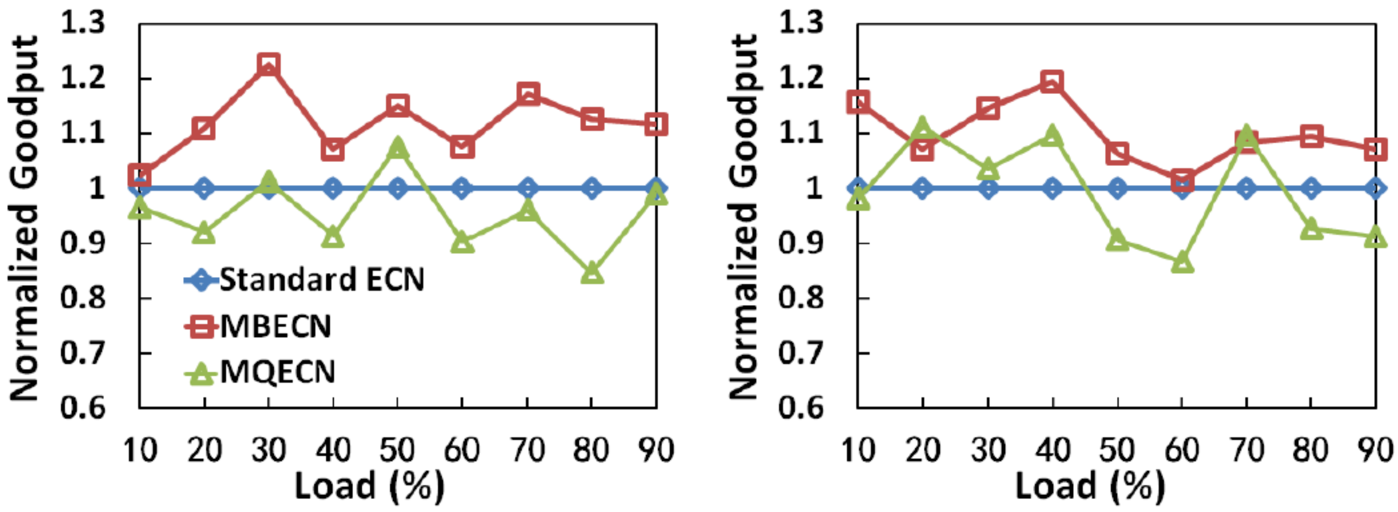
Testbed Experiment

- Topology
  - 8 senders, 1 receiver, 1 server-emulated switch



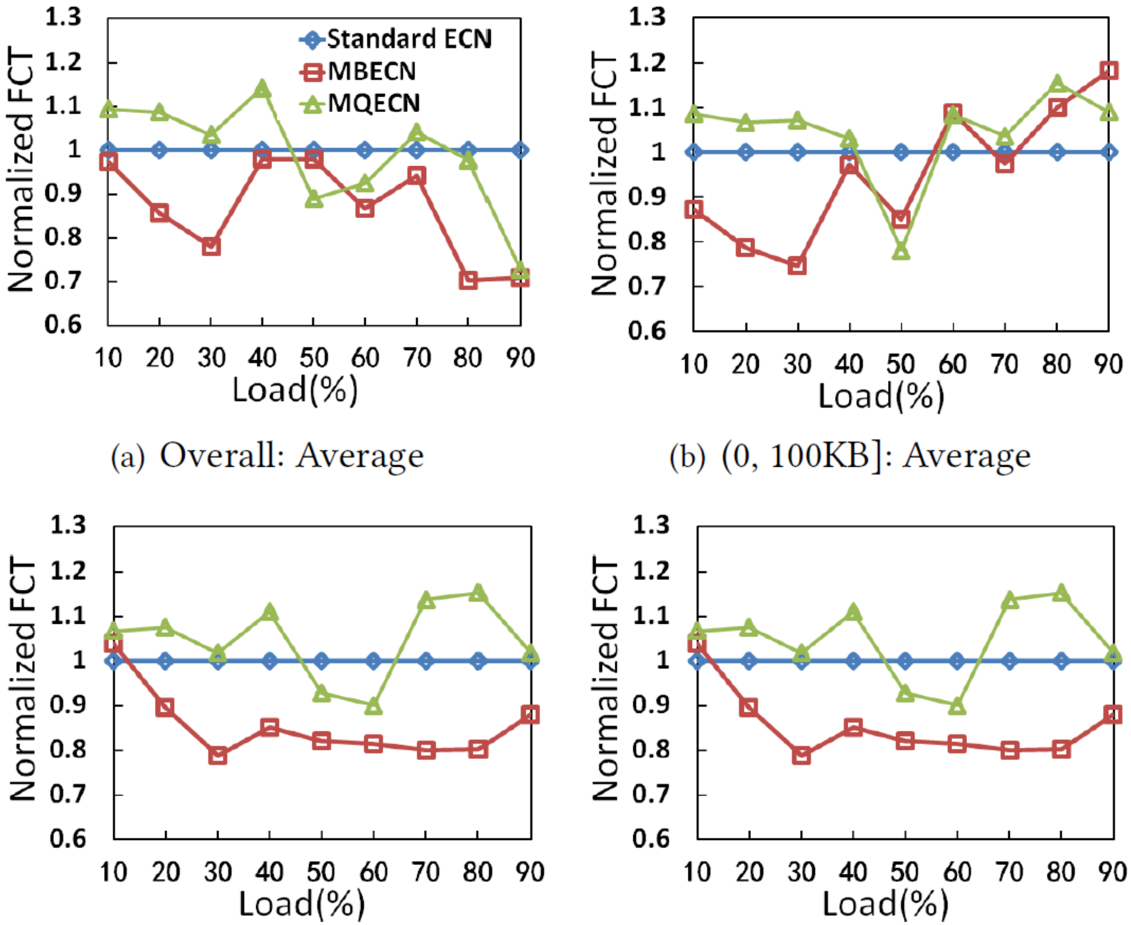
senders **receiver**

Testbed-Throughput



(a) Throughput in balanced traffic (b) Throughput in unbalanced traffic

Testbed-FCT with DCTCP



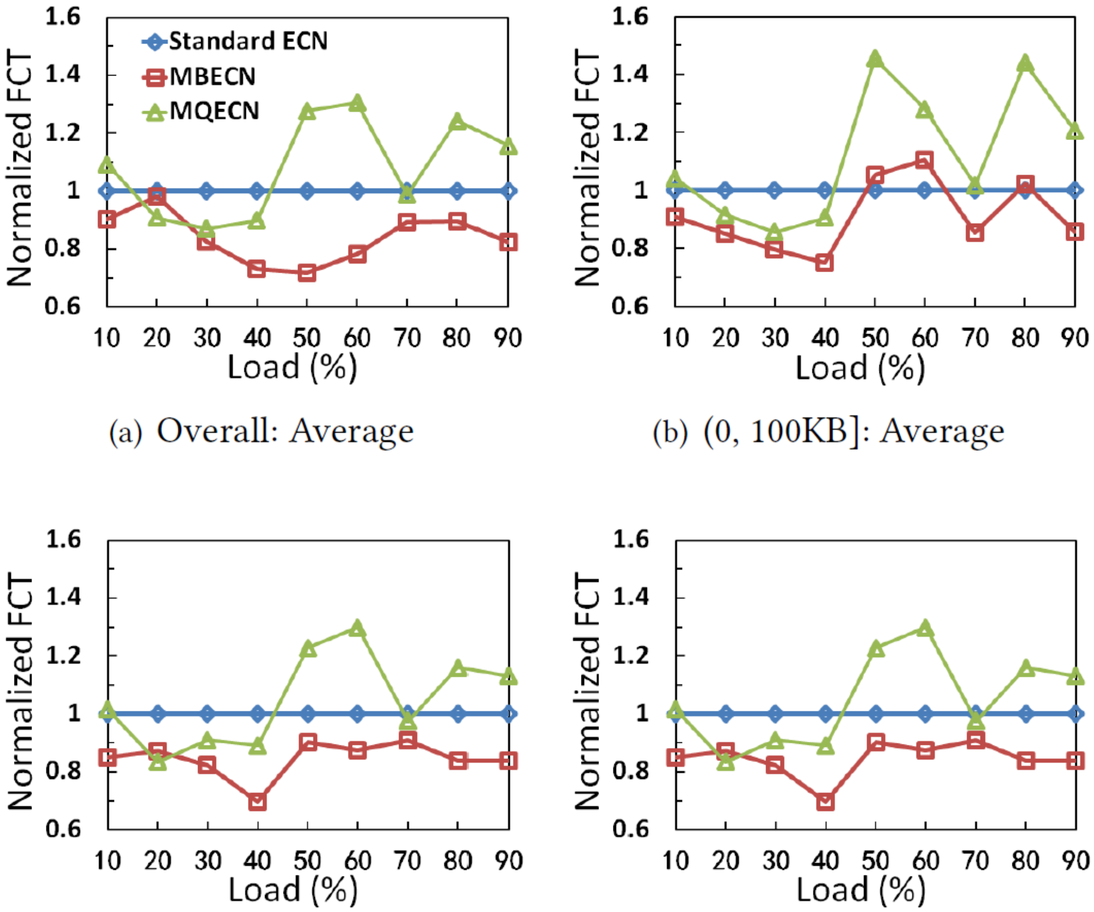
(a) Overall: Average

(b) (0, 100KB]: Average

(c) (100KB, 10MB]: Average

(d) (10MB, ∝): Average

Testbed-FCT with ECN*



(a) Overall: Average

(b) (0, 100KB]: Average

(c) (100KB, 10MB]: Average

(d) (10MB, ∝): Average