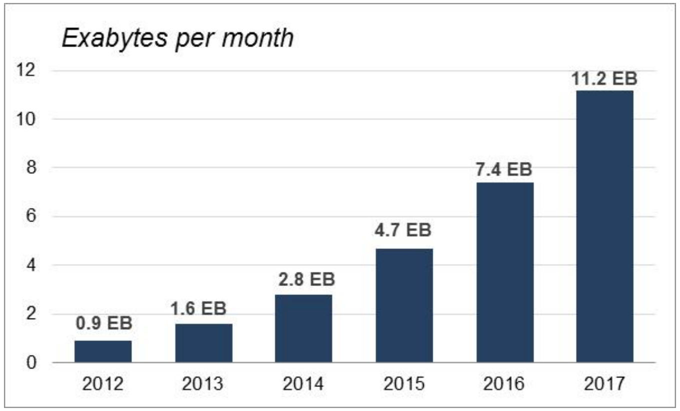
Edge versus Host Pacing of TCP Traffic in Small Buffer Networks

Hassan Habibi Gharakheili[1], Arun Vishwanath[2], Vijay Sivaraman[1]

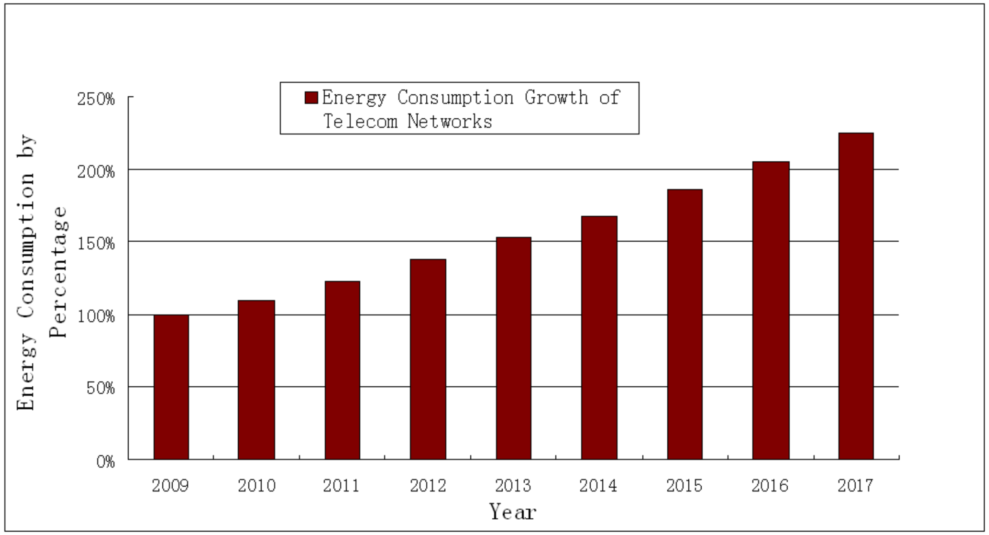
[1] University of New South Wales (UNSW), Australia
[2] University of Melbourne, Australia

Overview

- Increasing growth of data traffic
- Core network capacity growth
  - Energy concerns
- Use optical packet switching
  - Sacrifice buffering function



Source: Cisco VNI Forecast, 2013



Problem

- Small buffer network
  - Reduced buffer size (GB → MB/KB)
  - Increase congestion and contention
  - Performance loss

- TCP traffic
  - Bursty

Existing solutions

- Alleviate contentions
  - Wavelength conversion in the core
- Loss recovery
  - Packet-level forward-error-correction (FEC) at edge nodes
- Traffic pacing

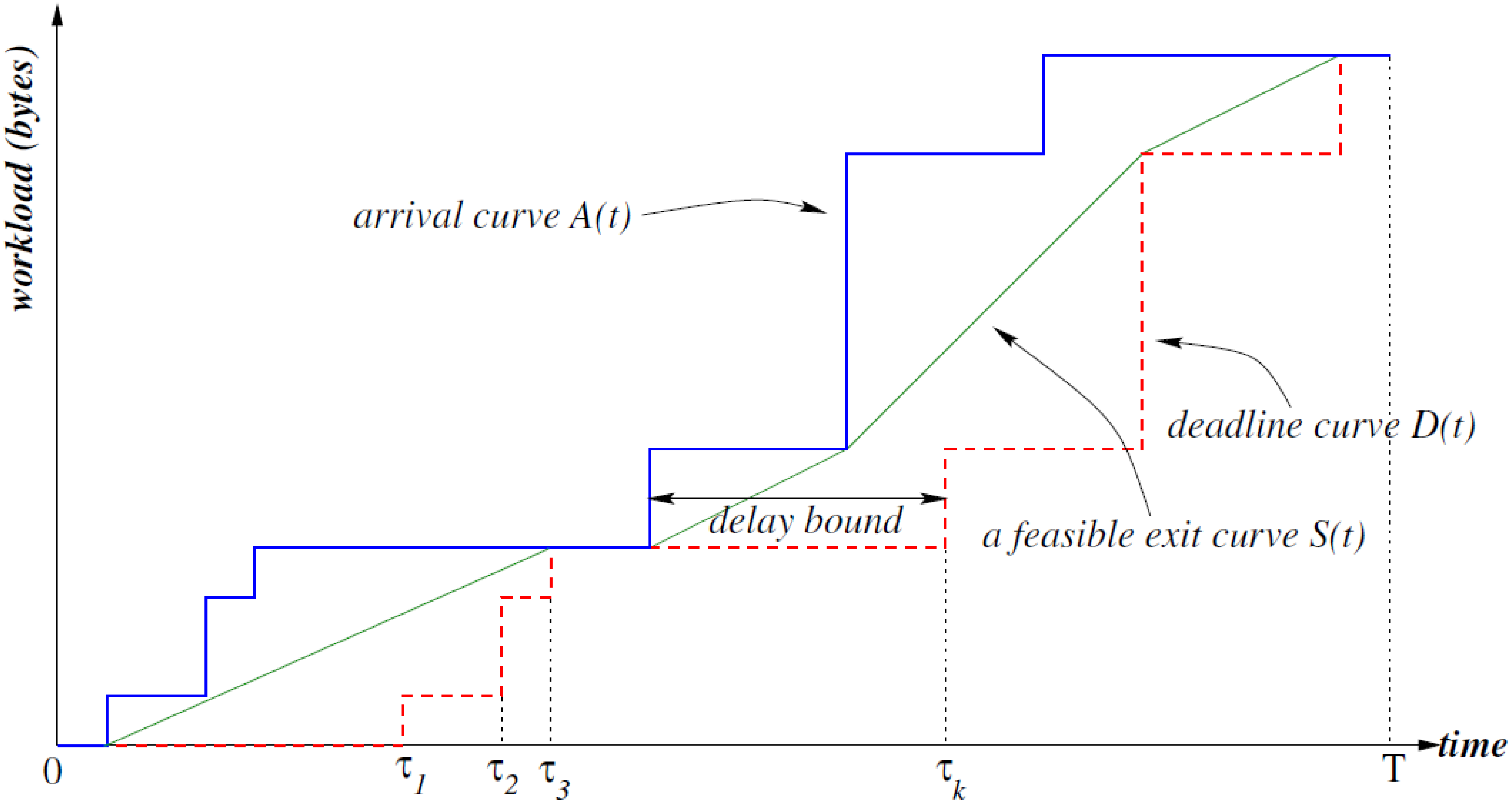
Traffic Pacing

- Host pacing (TCP pacing)
  - Requires TCP stack modification
  - Spreading packet transmission
    - Impractical (Out of operator control)
    - Costly (too many devices involved)
    - Paced hosts that are penalised over unpaced hosts [A. Aggarwal et al, IEEE INFOCOM]
- Edge pacing
  - Smoothing traffic prior injection into the core
    - by edge nodes

Edge-Pacer

- Input: traffic with given delay constraint
- Output: smoothest traffic s.t. time-constraint
- Adjusts traffic release rate to maximise smoothness, subject to a given upper bound on packet delay

Edge-Pacing mechanism



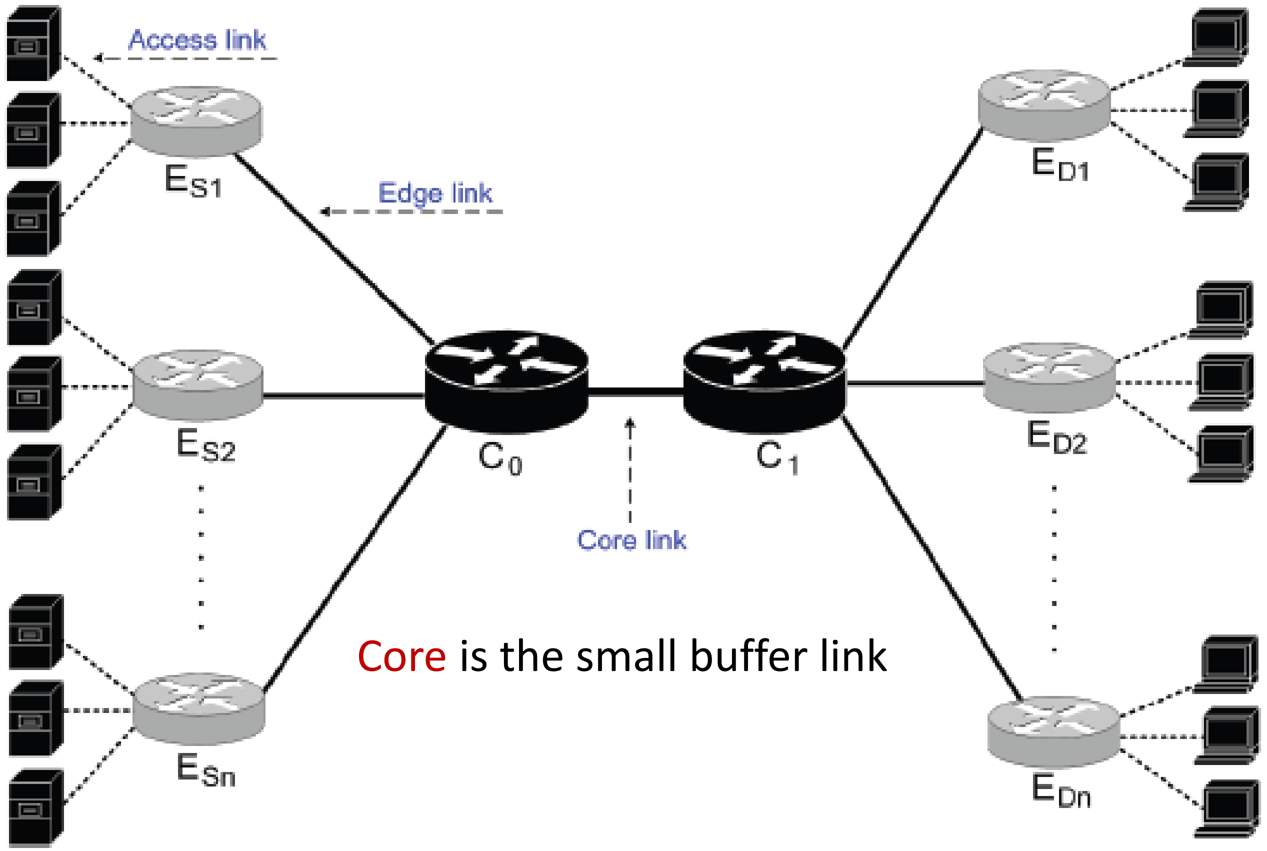
Contributions

- Simulations of small-buffer core network
  - Bottleneck vs. Non-bottleneck
  - Low-speed vs. high-speed access links
  - Short-lived vs. long-lived flows
  - Different number of flows
  - Different variants of TCP

- Selection model for edge pacing delay

- Benefits of incremental deployment

ns-2 Simulation



- Simulation is run for 400 sec
- Data in the interval [100, 400]sec is used (capture the steady state)

Small buffer link as the bottleneck

- 10 edge links (ES_1-ES_{10}) and (ED_1-ED_{10})
  - 100Mbps
  - uniformly distributed [5, 15]ms propagation delay

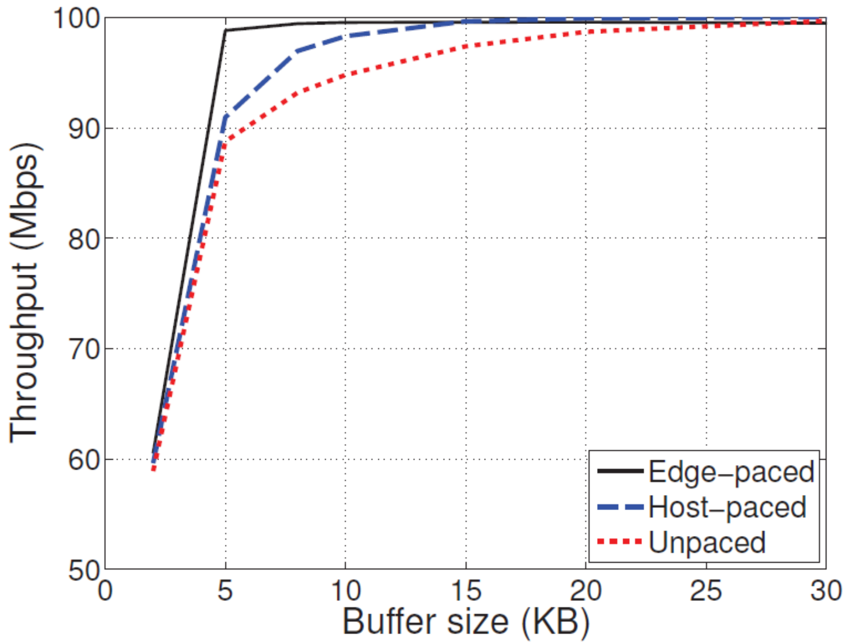
- Each edge link is fed by 100 access links
  - uniformly distributed [5, 15]ms propagation delay

- 1000 end-host (= 10 * 100) having one TCP Reno flow each
  - 1000 TCP flows start randomly distributed [0, 10]sec

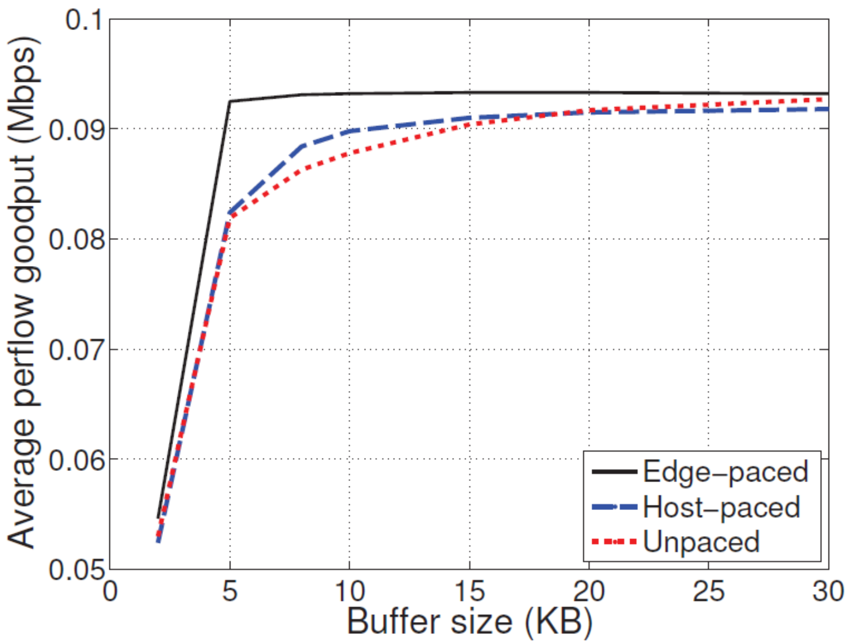
- Core Link
  - 100Mbps (bottleneck)
  - 100ms delay
  - FIFO queue with droptail queue management
  - queue size is varied in terms of KB
  - RTT : [224, 280]ms

Small buffer link as the bottleneck

Aggregate TCP throughput



Average per-flow goodput

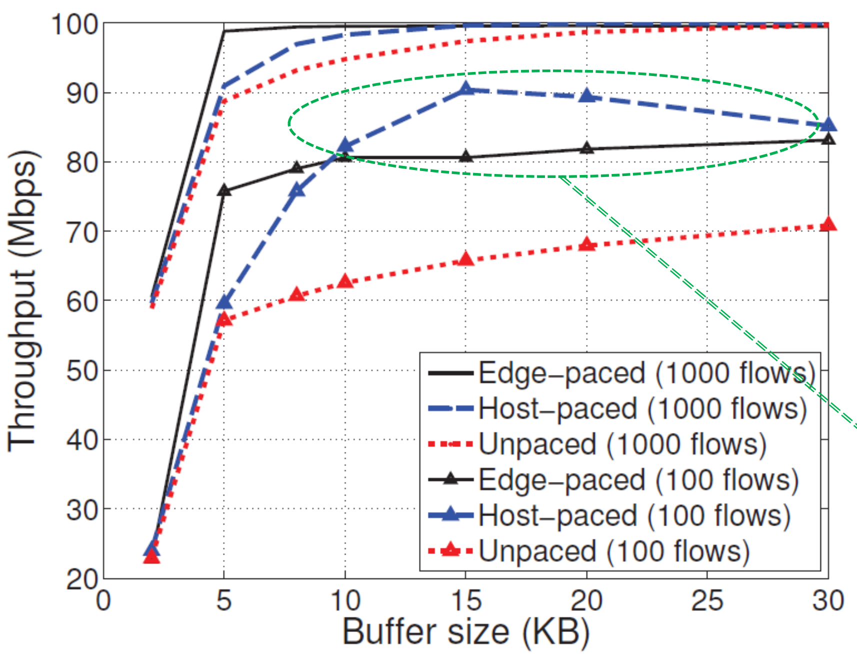


Outperformance of edge pacing

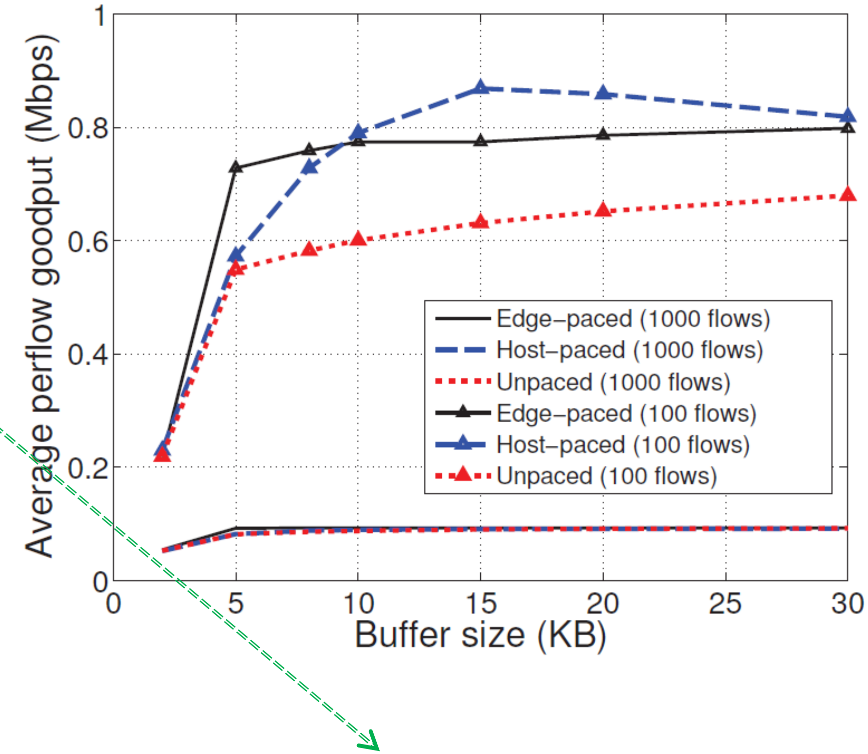
(especially in the region of 5-15KB buffers)

Number of TCP flows

Aggregate TCP throughput



Average per-flow goodput

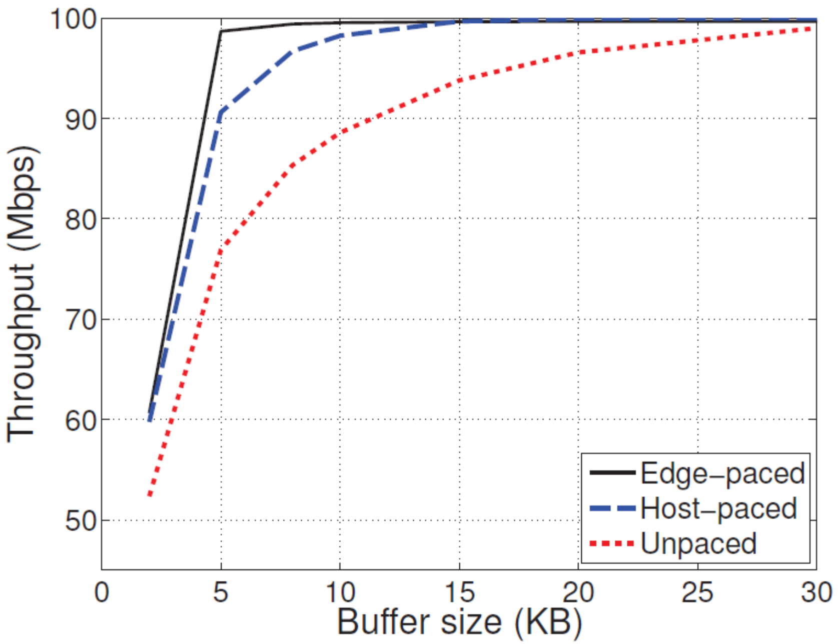


Same setup as before, but alter the number of access links (10, 50, 100) feeding into the edge ➔ thus 100, 500 and 1000 flows respectively

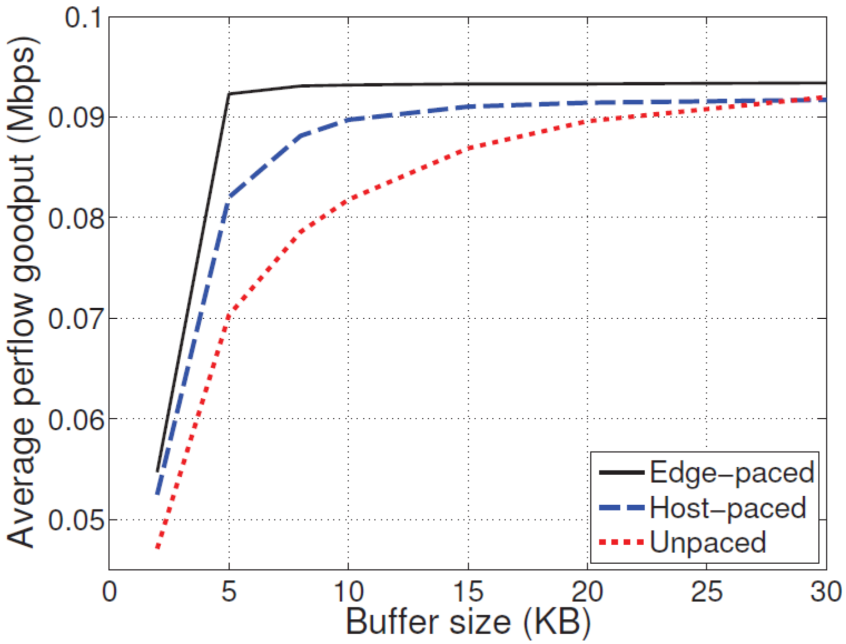
With small number of flows; individual flow burtiness contributes more to loss than simultaneous arrival ➔ host pacing effectively reduces the source burstiness

High-speed access links

Aggregate TCP throughput



Average per-flow goodput

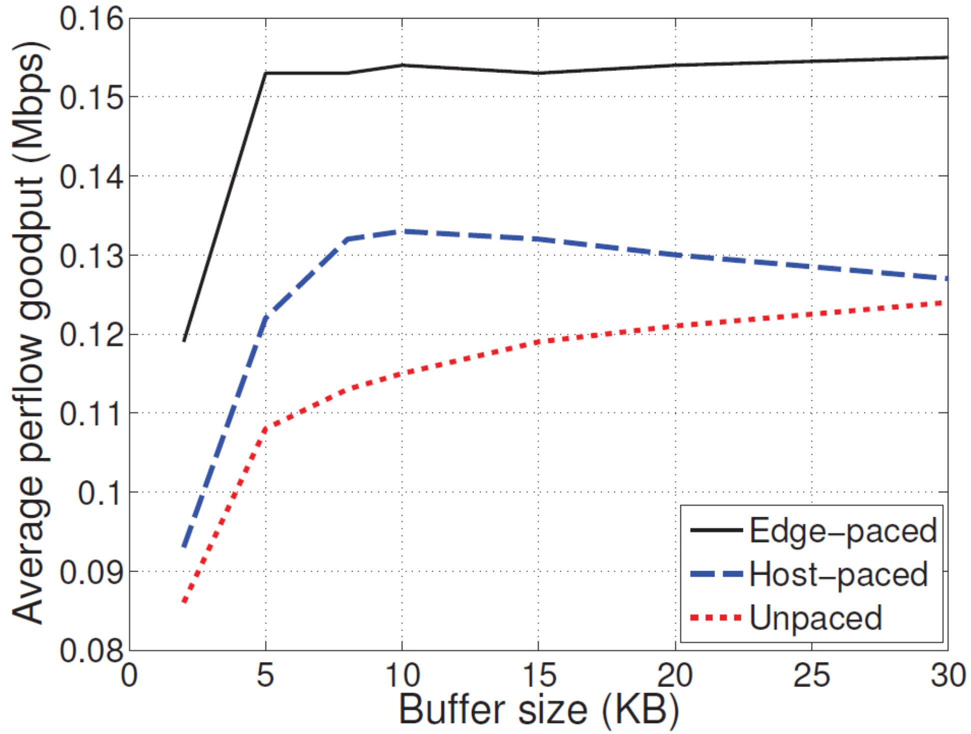


Setup for 1000 flows but access links operate at 100Mbps (enterprise, data-centre, ..)

Avg. goodput of 90Kbps, requires buffer;

- 20KB for unpaced
- 10KB for host-paced
- 5KB for edge-paced

Short-lived (mice) TCP flows



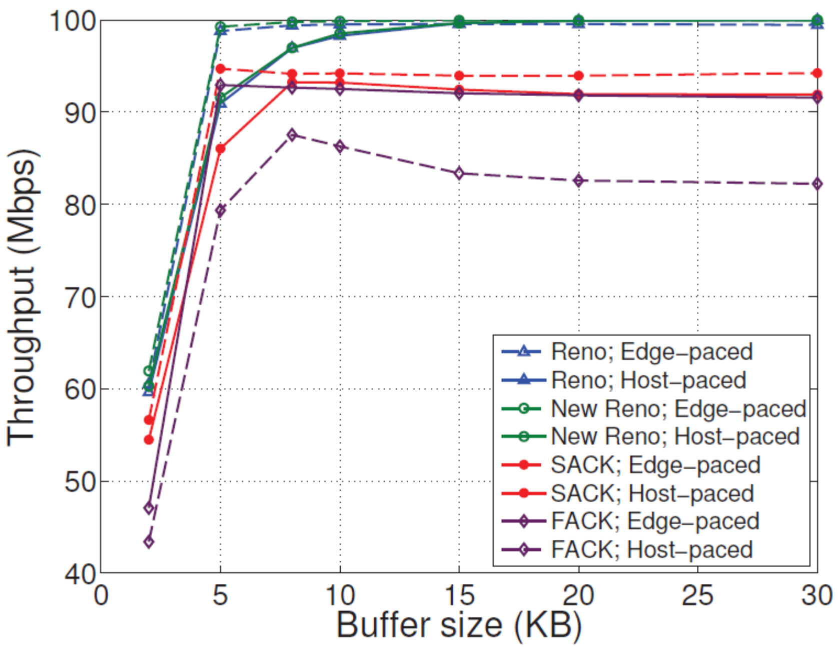
Efficacy of edge pacing in combating short time-scale burstiness

On-off traffic flow;

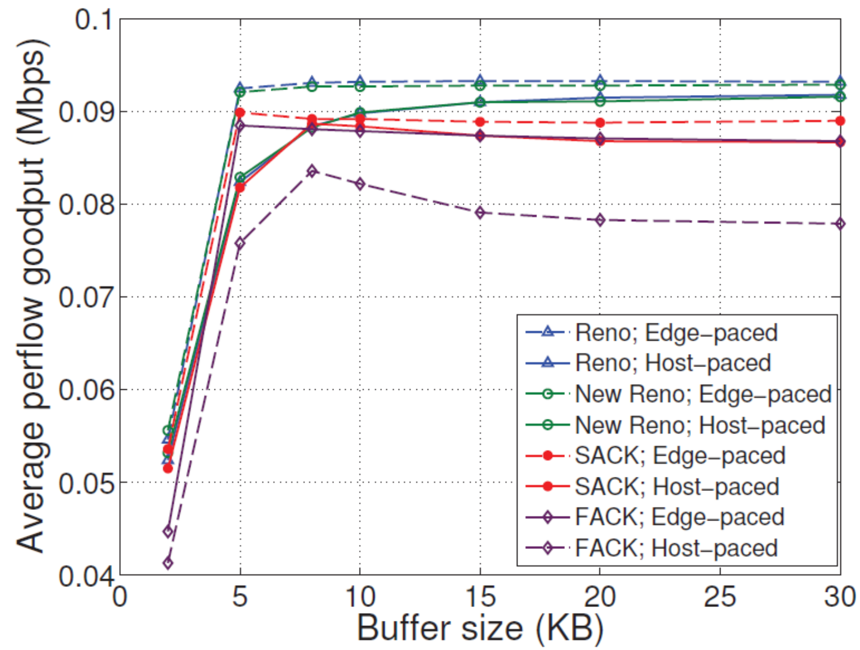
- On: size of transferred file follows Pareto distribution with mean 100 KB and shape parameter 1.2
- Off: duration of the “thinking period” is exponentially distributed with mean 1 sec

Different versions of TCP

Aggregate TCP throughput



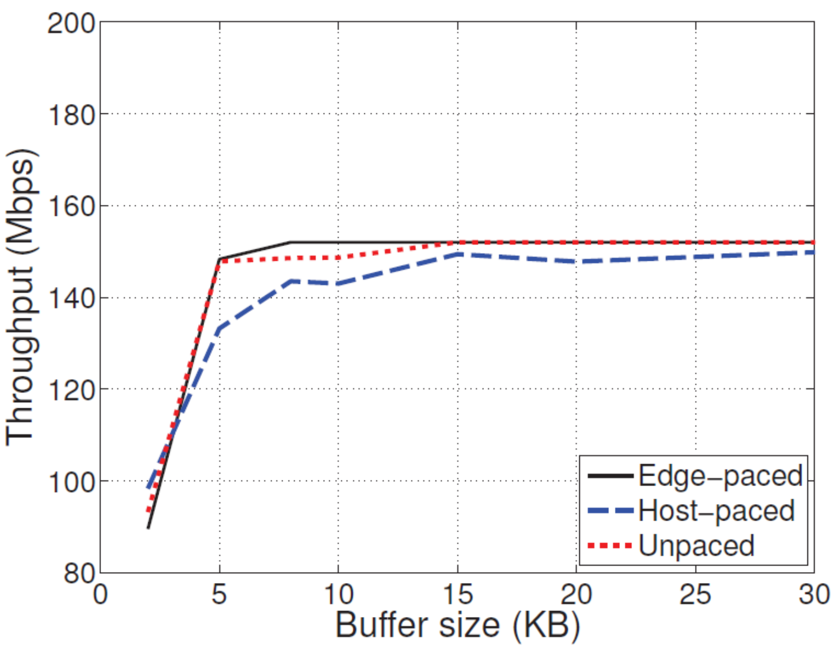
Average per-flow goodput



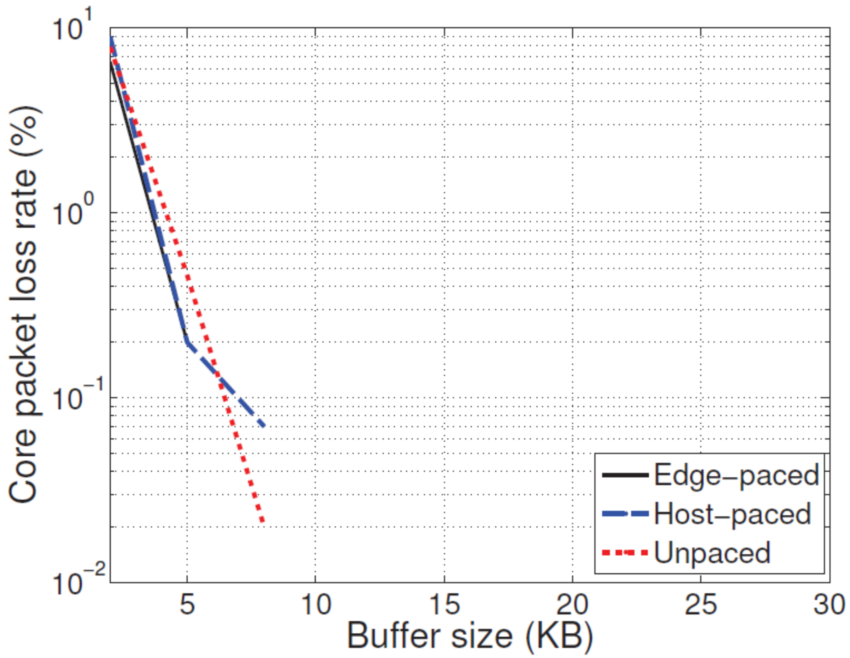
- Setup for 1000 flows, Low-speed access links, and Core bottleneck
- for all above variants of TCP, edge pacing offers better performance than host pacing

Small buffer link not the bottleneck

Aggregate TCP throughput



Packet loss rate at the core link



Setup:

- Core @ 200Mbps
- 10 edges @ 40Mbps
- 10 access / edge @ [1, 2] Mbps
- Buffer < 10KB ➔ Zero packet loss

TCP throughput is not sensitive to pacing when the small buffer link is not the bottleneck

Analysis

- Modeling TCP performance
  - difficult due to its control feedback loops

- TCP throughput

$$T \propto \frac{1}{RTT\sqrt{L}}$$

- Edge pacer increases the mean RTT (i.e. RTT_0)
  - Pacer with delay bound d , adds on average $d/2$ delay in each direction:
  - $RTT_0 \Rightarrow RTT_0 + d$

Analysis (cnt'd)

- Aggregate traffic of TCP flows sharing small buffer, is Poisson-like with a certain rate λ
- Traffic burstiness [5]:

$$\beta = 1/\sqrt{2\lambda d}$$

- Loss rate [5]:

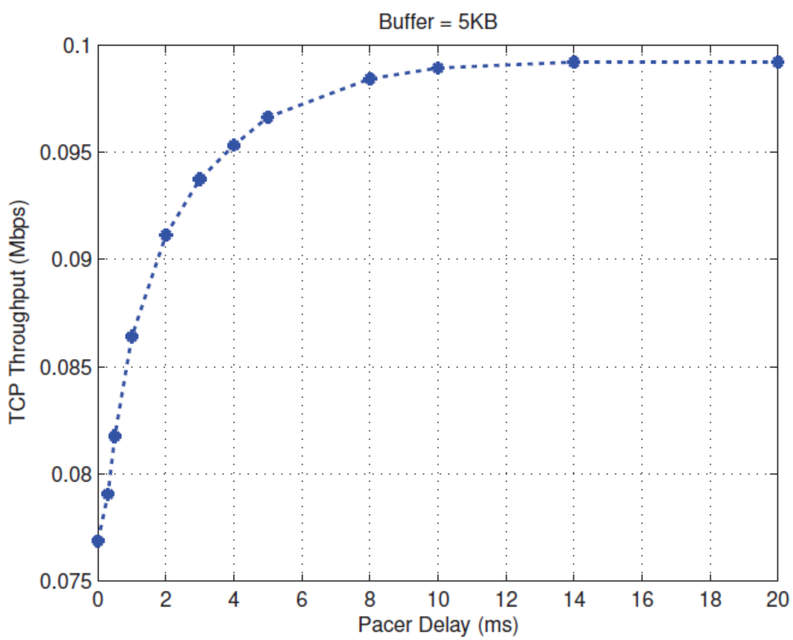
$$L \leq (\lambda e^{1-\lambda})^{2d}$$

Analysis (cnt'd)

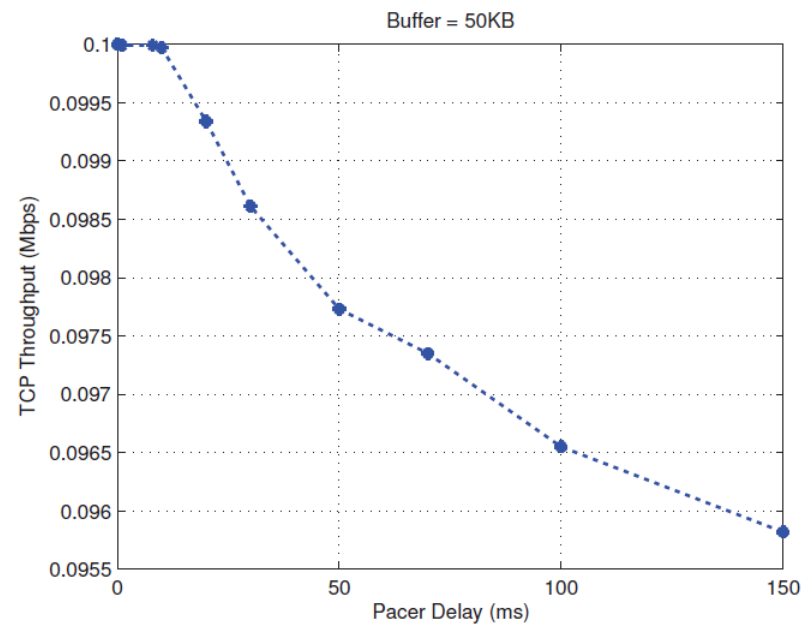
- Therefore;

$$T \propto \frac{1}{(RTT_0 + d)(\lambda e^{1-\lambda})^d}$$

Low load / Small buffer



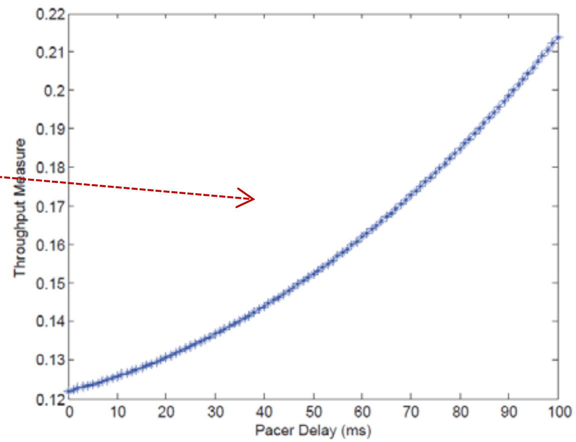
High load / Large buffer



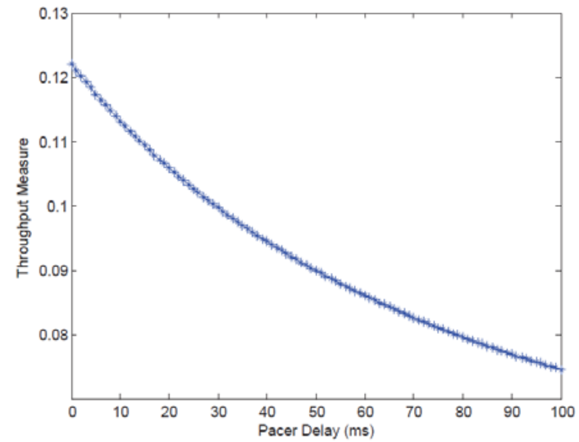
Analysis (cnt’d)

Analysis assumes a fixed load

Low load / Small buffer (analysis)

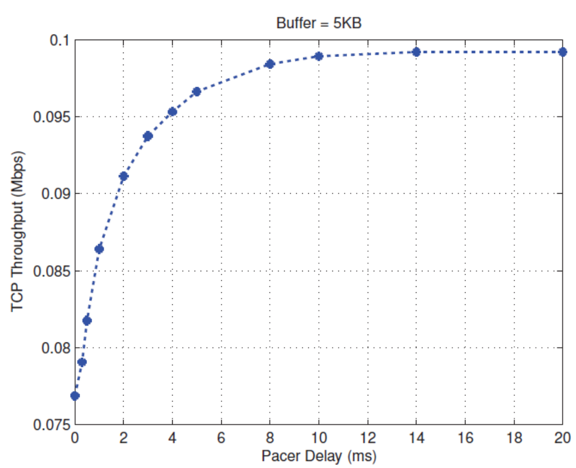


High load / Large buffer (analysis)

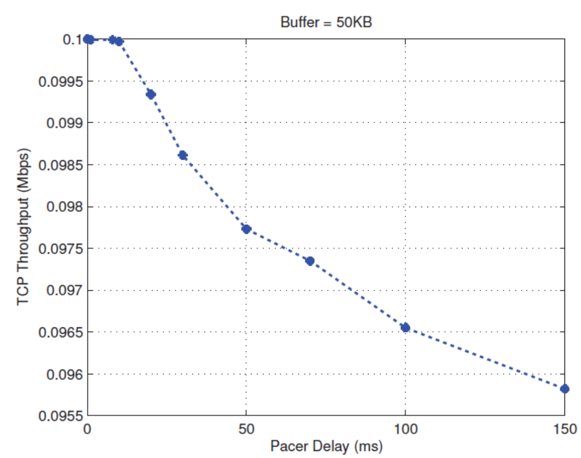


Increasing pacing delay results low loss reduces, then loss reduction is compensated by TCP reaction of increasing offered load ➔ TCP throughput curve saturates in simulation by pacing delay reaches 10 ms

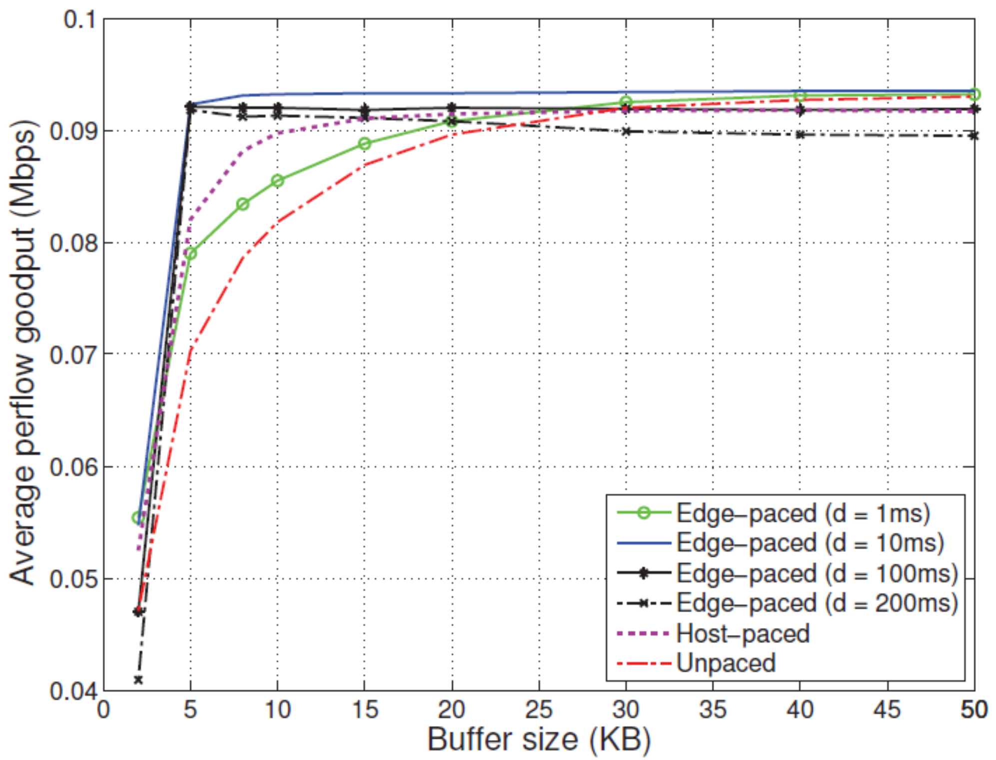
Low load / Small buffer (sim)



High load / Large buffer (sim)



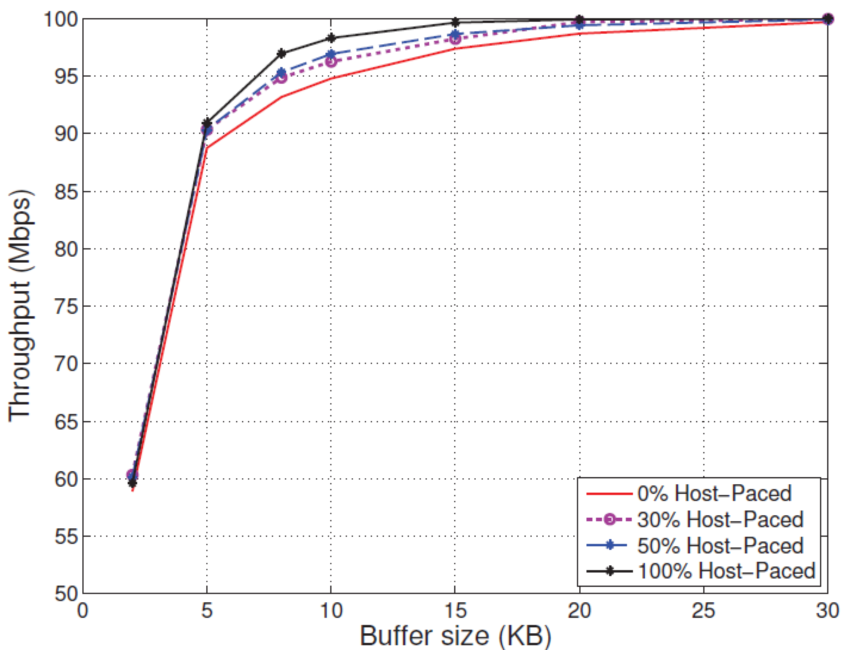
Variation of pacing delay



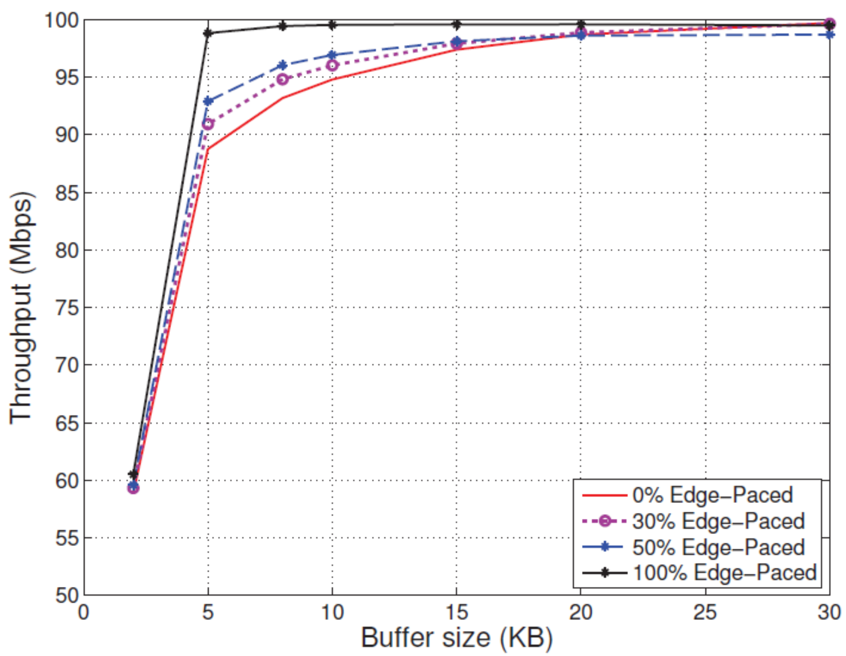
- Small pacing delay $d = 1$ ms: ineffective at small buffer sizes
- large pacing delay such as $d = 100$ or $200\ ms$: detrimental at large buffer sizes
- Throughout our simulations we found that $d = 10\ ms$ performs well across entire range of buffer sizes

Practical deployment of Pacing

Host pacing



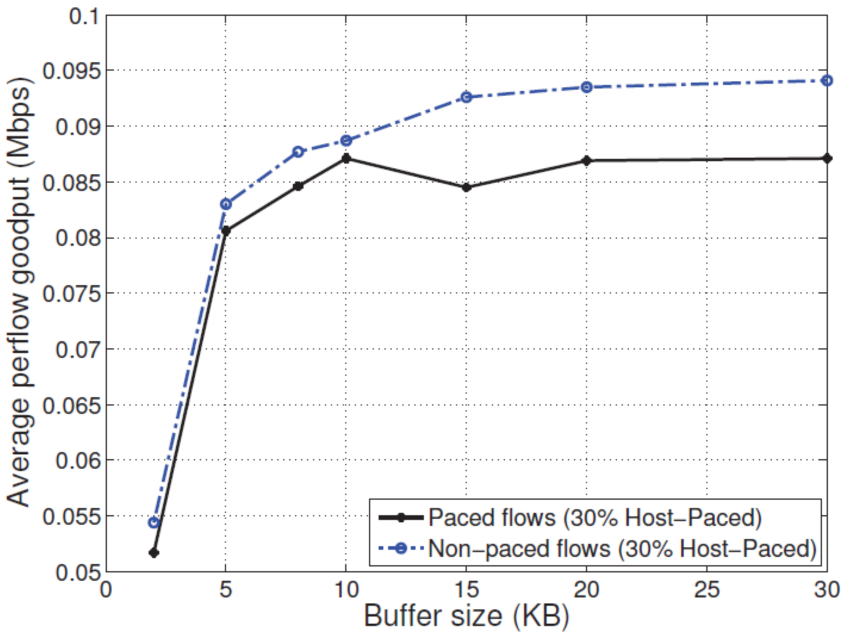
Edge pacing



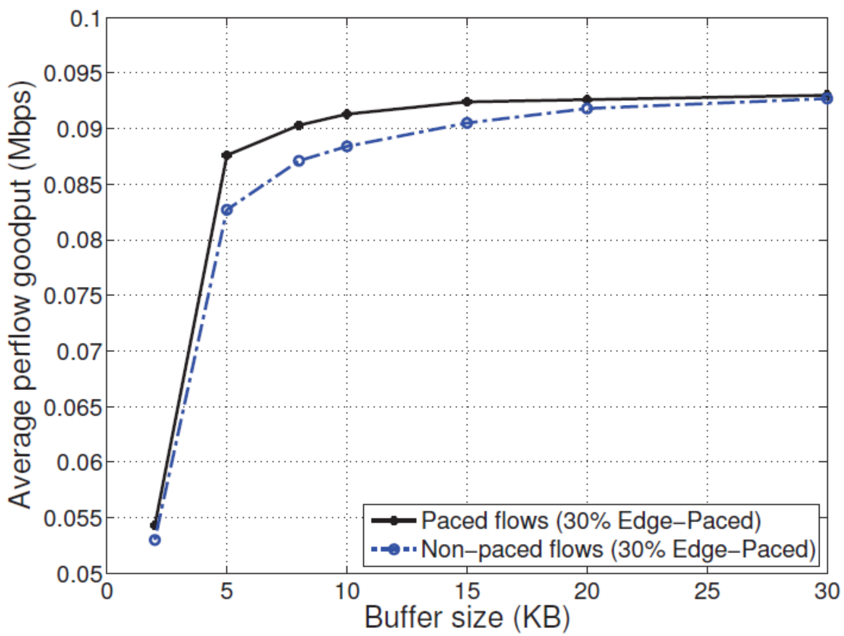
Throughput rises gradually as the fraction of hosts/edges that perform pacing increases, and therefore it would seem the benefits of pacing can be realised incrementally with progressive deployment

Practical deployment of Pacing

Host pacing



Edge pacing



- 30% pacing deployed (i.e. 300 out of 1000 flows perform TCP pacing in the case of host pacing and 3 out of 10 edge nodes perform pacing in the edge pacing case)
- Early adopters of host pacing can obtain worse performance than their non-pacing peers ➔ substantial disincentive for users to deploy host pacing
- However, for edge pacing; paced flows experience better performance than unpaced ones

Conclusion

- Energy concern of high-speed routers
  - Optical switching ➔ reduced buffering
- Two different pacing technique to address TCP performance
- Edge-pacing performs as good as Host-pacing or better
- Clear incentive for incremental deployment of edge-pacing in operational network