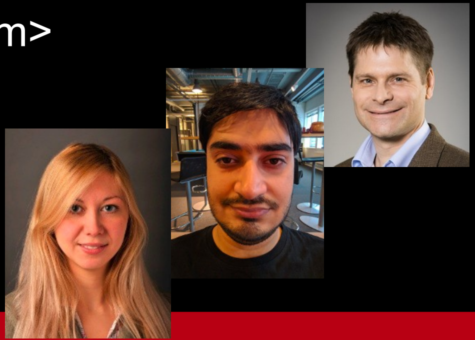
Low Latency Low Loss Scalable Throughput (L4S)

draft-ietf-tsvwg-l4s-arch-06
draft-ietf-tsvwg-ecn-l4s-id-10 {ToDo: 11}
draft-ietf-tsvwg-aqm-dualq-coupled-11

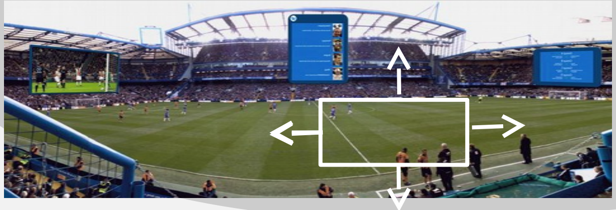
Bob Briscoe, Independent <ietf@bobbriscoe.net>
Koen De Schepper, NOKIA Bell Labs <koen.de_schepper@nokia.com>
Olivier Tilmans, NOKIA Bell Labs <olivier.tilmans@nokia-bell-labs.com>
Greg White, CableLabs <g.white@CableLabs.com>
Asad Sajjad Ahmed, Independent <me@asadsa.com>
Olga Albisser, Simula Research <olga@albisser.org>



TSVWG, IETF-107, Mar 2020

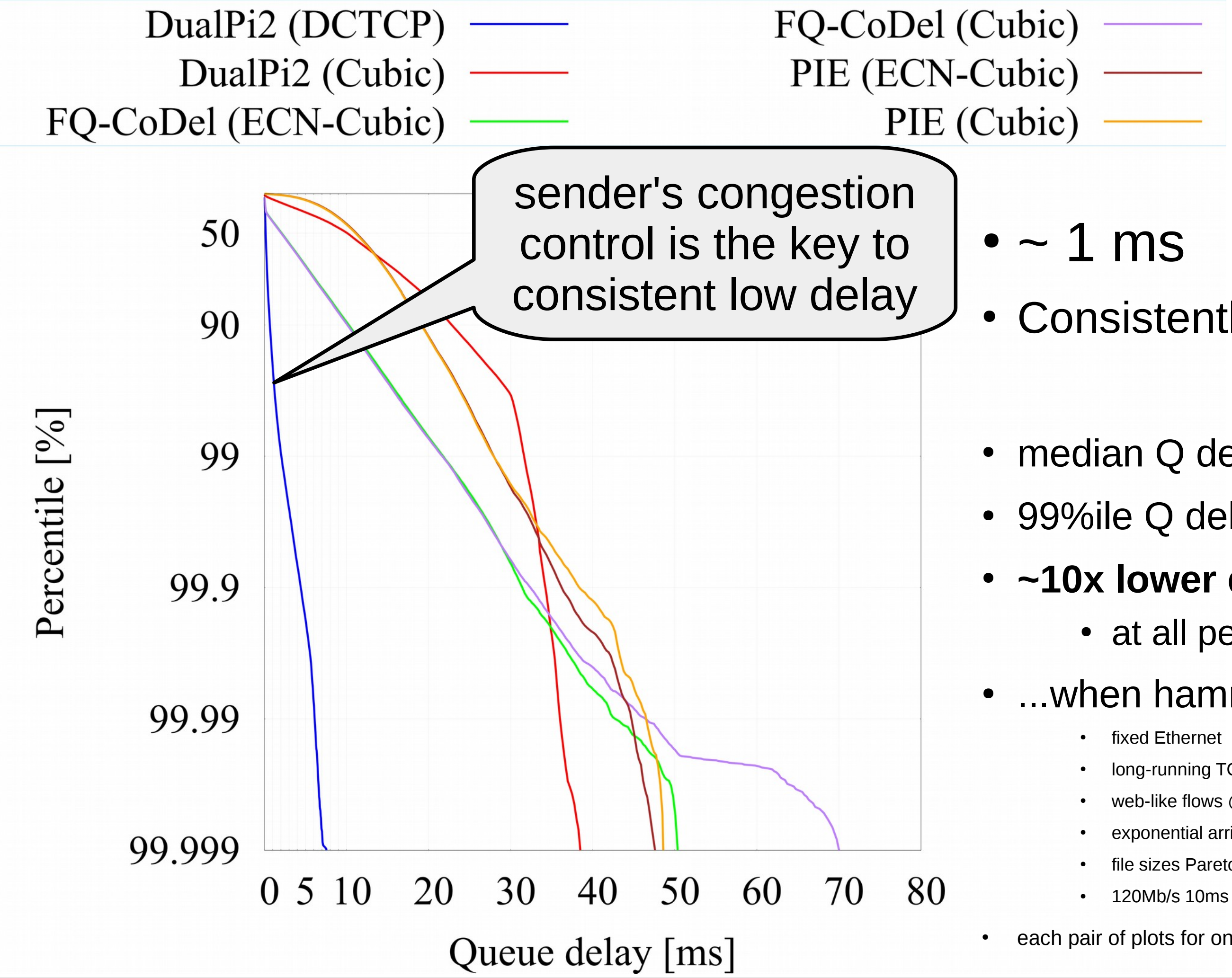
Ultra-low latency even with high throughput for *all* applications

- Not only non-queue-building traffic
  - DNS, gaming, voice, SSH, ACKs, HTTP requests, etc
- Capacity-seeking and adaptive real-time as well
  - TCP, QUIC, RMCAT for WebRTC
  - web, HD video conferencing, interactive video, cloud-rendered virtual reality, augmented reality, remote presence, remote control, interactive light-field experiences,...



[L4S-MMSYS]

"Ultra-low" Q delay?

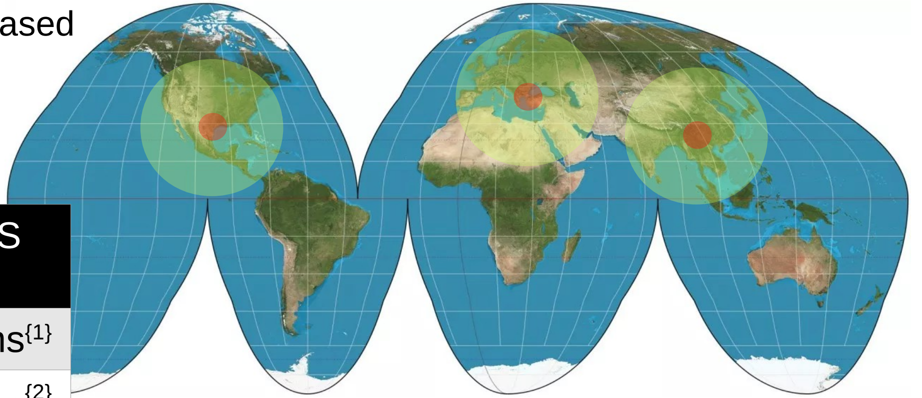


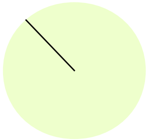
- ~ 1 ms
- Consistently – for real-time apps

- median Q delay: 100-200μs
- 99%ile Q delay: 1-2ms
- **~10x lower delay than best 2nd gen. AQM**
  - at all percentiles
- ...when hammering each AQM
  - fixed Ethernet
  - long-running TCPs: 1 ECN 1 non-ECN
  - web-like flows @ 300/s ECN, 300/s non-ECN
  - exponential arrival process
  - file sizes Pareto distr. α=0.9 1KB min 1MB max
  - 120Mb/s 10ms base RTT
- each pair of plots for one AQM is one experiment run

Is such consistently low delay needed?

- For responsive feel{1}, as interaction becomes more video-based
- L4S gives 5x more reach than previous AQMs{3}
  - from each user, or from each data centre
  - Los Angeles to Atlanta, not just to Phoenix

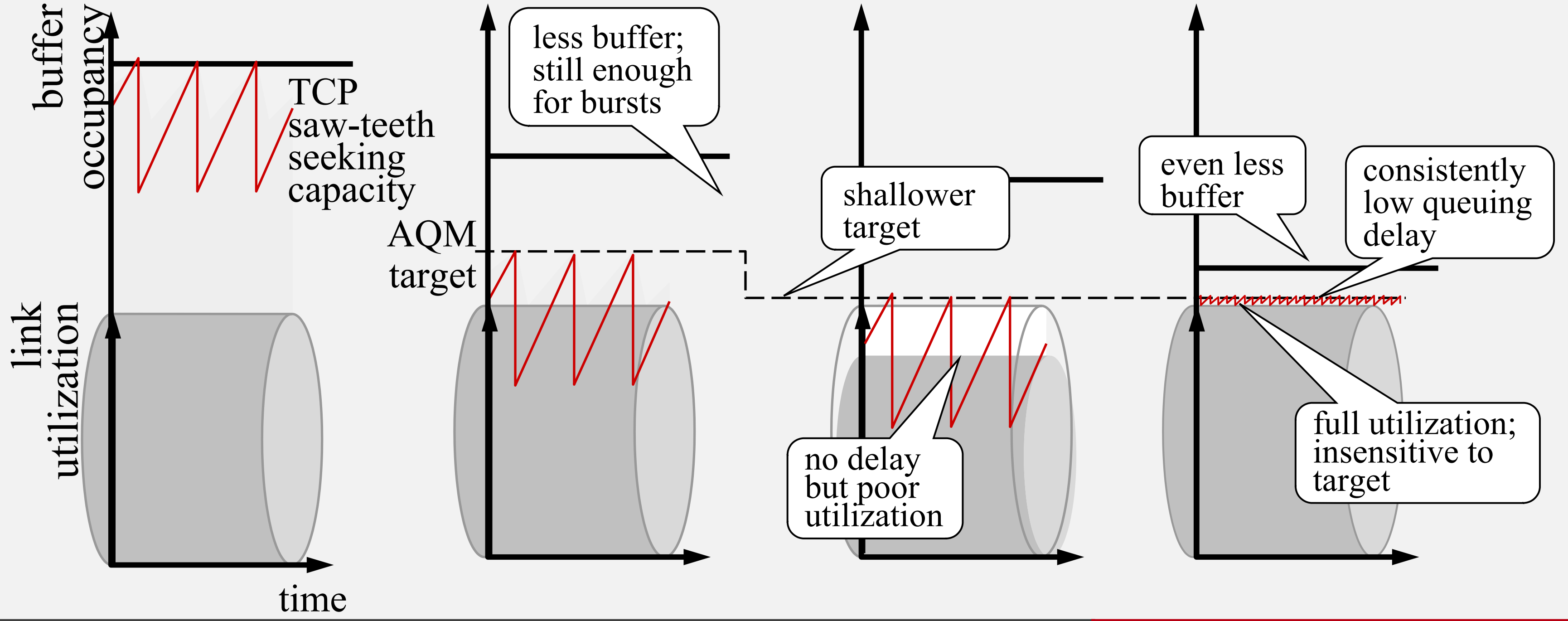


	PIE or FQ_CoDel	L4S
Delay budget for 'responsive feel'	50ms{1}	50ms{1}
min non-network delay	– 13ms{2}	– 13ms{2}
99.9th %ile queuing delay	– 32ms	– 4.5ms
Left for propagation round trip	5ms	32ms
Equivalent reach in fibre	500km (310 miles)	3200km (2000 miles)
Visualized radius on the map		

1: <20ms latency 'imperceptible', 50ms feels responsive according to the VR "Oracle" [Carmack13]
2: draft-han-iccrg-arvr-transport-problem-01#appendix-A.1.2

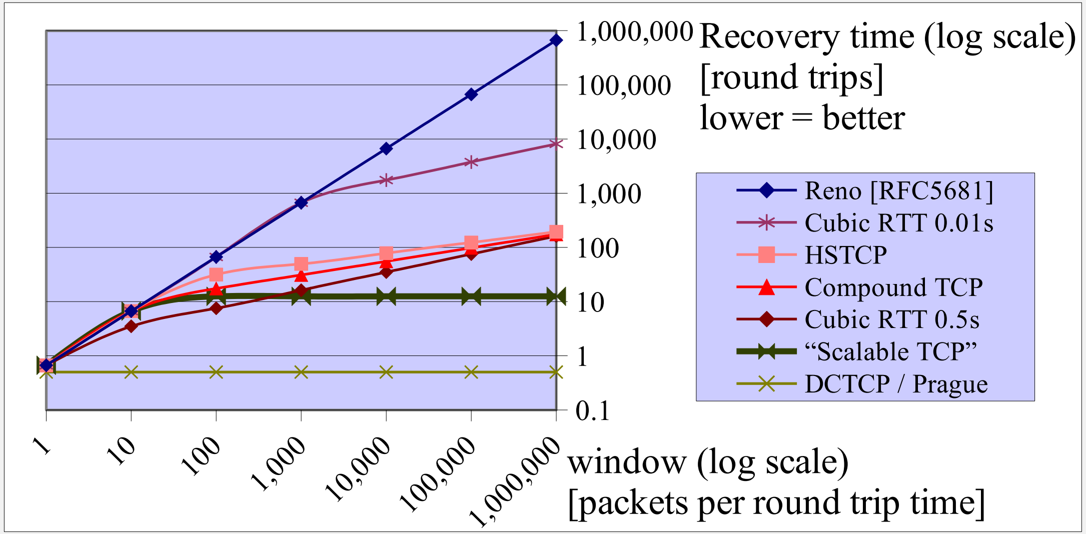
The trick: scalable congestion control

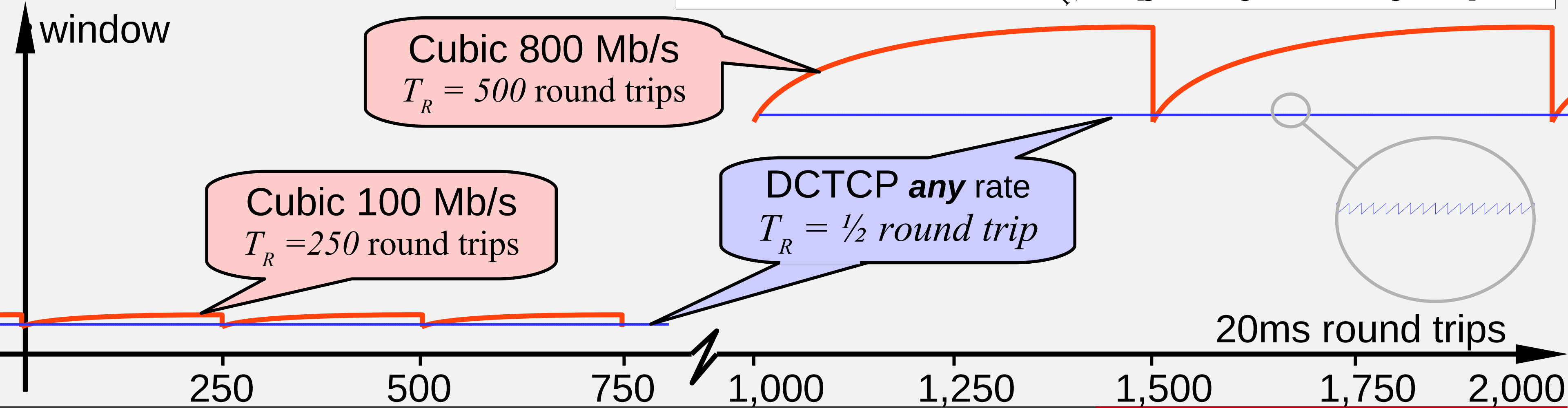
	(1) Today (typical)	(2) Today (at best)	(3) Unacceptable	(4) L4S
Bottleneck	Bloated drop-tail buffer	AQM	Shallower AQM	Immediate AQM
Sender CC	Classic	Classic	Classic	Scalable (tiny saw-teeth)



'Scalable'?

- Duration of sawteeth (recovery time) is invariant as flow rate scales [RFC3649]
  - otherwise problems return in a few years:
    - more queue delay or underutilization
    - more sensitive to disturbance
    - more sluggish at tracking dynamics





L4S ECN

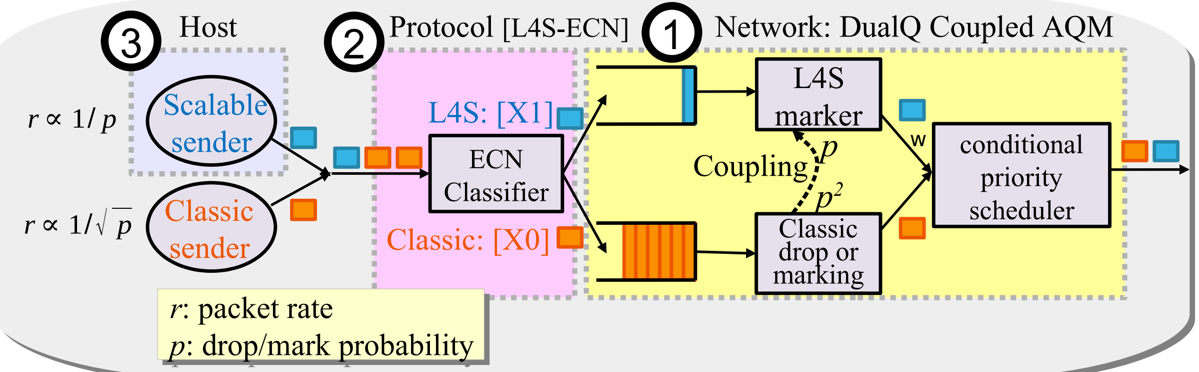
- Scalable congestion controls use ECN
  - but no longer equivalent to loss
  - scalable congestion signals would be too frequent to use loss
- Fine-grained ECN feedback required
  - Standards track update to TCP wire protocol
  - Supported by IETF QUIC from the start
- Smoothing of congestion signals – shifted from network to sender
  - sender knows its own RTT, network otherwise has to smooth over worst-case RTT
  - network just marks ECN based on simple instantaneous queue

The Coexistence Problem

- If Scalable & Classic traffic share a queue:
  - No Latency Isolation: Classic congestion controls need large queue to utilize link (~1 base RTT)
  - Capacity Sharing: Scalable flows induce high ECN marking, which makes Classic flows yield

Coexistence

between Scalable & Classic traffic



1) network:

- DualQ Coupled AQM:
  - Dual Queues: Isolate Scalable traffic from Queuing Delay of Classic Traffic
  - Coupled AQMs: counterbalances more aggressive CC with equally aggressive marking
    Equalizes flow rates across queues without flow inspection
- or per-flow Qs with shallow ECT(1) threshold

2) packet identifier:

- ECT(1) in IP/ECN field

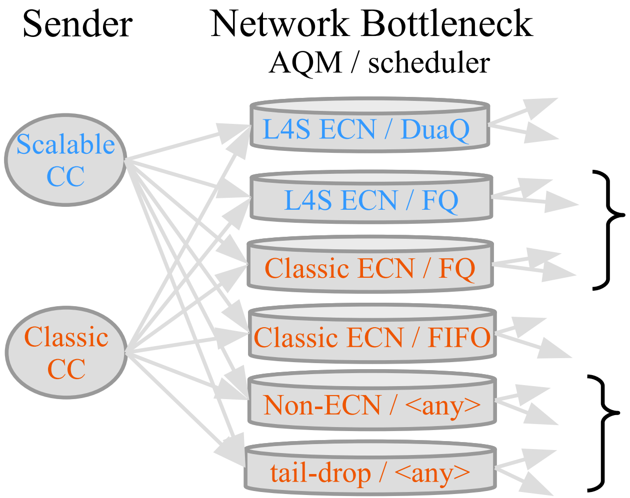
Codepoint	IP-ECN bits	Meaning
Not-ECT	00	Not ECN-Capable Transport
ECT(0)	10	Classic ECN-Capable Transport
ECT(1)	01	L4S ECN-Capable Transport
CE	11	Congestion Experienced

3) host:

- For non-L4S bottlenecks falls back to Reno-friendly
  - on loss: always
  - on classic ECN: only necessary during transition (later slide)

The Full Coexistence Problem

- Across all combinations of congestion control, AQM & scheduler

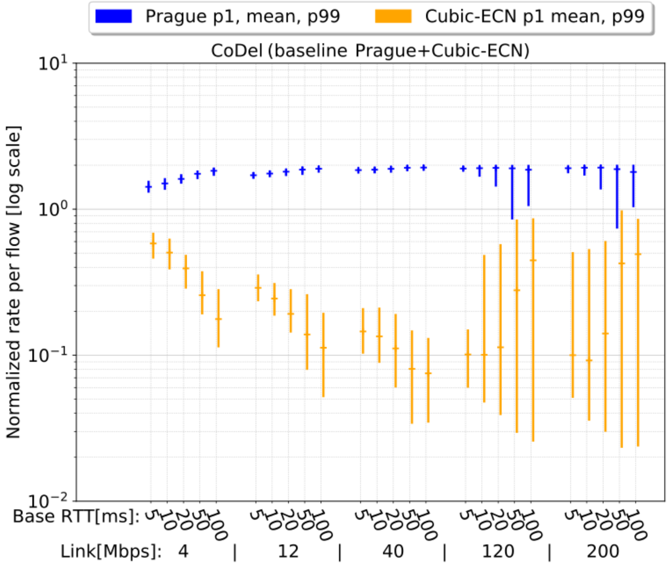


- Network-based solution: Dual Q Coupled AQM (previous slide)
- Non-problem:
  per-flow scheduler enforces capacity shares
- Problem: Scalable CCs induce frequent ECN marking;
  Classic CCs yield to apparent high congestion (next slide)
- Non-problem: Scalable CCs apply
  Reno-friendly response to drop

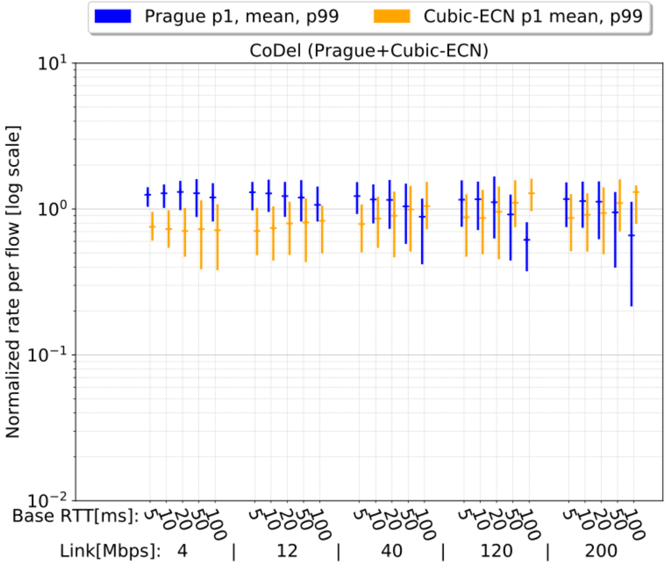
- Classic ECN: RFC3168 Explicit Congestion Notification
- CC: Congestion Control
- Scalable CC: 1/p response to congestion (p)
- Classic CC: Reno-Friendly CC
- AQM: Active Queue Management
- FIFO: First-In First-Out
- FQ: Per-Flow Queuing
- L4S: Low Latency Low Loss Scalable throughput

Sender-based Coexistence Solutions

- CC detects larger queue variability of Classic ECN bottleneck
  - and falls back to Reno-Friendly congestion response



Without fall-back algo



With fall-back algo

Normalized rate per flow = flow rate after convergence / (capacity / no. of flows)
CoDel AQM (not FQ), default config. See [Fallback20] for more scenarios

more info

All via L4S Landing page: https://riteproject.eu/dctth/

Systems Papers about L4S

- [DCttH19] Koen De Schepper (Nokia Bell Labs), Olga Albisser (Simula), Olivier Tilmans (Nokia Bell Labs) and Bob Briscoe (CableLabs), `Data Center to the Home': Deployable Ultra-Low Queuing Delay for All, Draft Paper (Jul 2019).
- [DualPI2_19] Albisser, O., De Schepper, K., Briscoe, B., Tilmans, O. & Steen, H., "DUALPI2 - Low Latency, Low Loss and Scalable (L4S) AQM," In: Proc. Netdev 0x13 (March 2019)
- [TCPPrague19] Briscoe, B., De Schepper, K., Albisser, O., Misund, J., Tilmans, O., Kühlewind, M. & Ahmed, A.S., "Implementing the `TCP Prague' Requirements for L4S," In: Proc. Netdev 0x13 (March 2019)

Papers on Detailed Aspects

- [Tensions17] Briscoe, B. & De Schepper, K., "Resolving Tensions between Congestion Control Scaling Requirements," Simula Technical Report TR-CS-2016-001; arXiv:1904.07605 (July 2017)
- [Fallback20] TCP Prague Fall-back on Detection of a Classic ECN AQM, Bob Briscoe (Independent) and Asad Sajjad Ahmed (Independent), bobbriscoe.net Technical Report TR-BB-2019-002; arXiv:1911.00710v2 [cs.NI] (Apr 2020) – see also large online visualization of evaluation

IETF Specifications of L4S

- [ecn-expt] Black, D. “Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation” RFC8311 (Jan 2018)
- [l4s-arch] Briscoe (Ed.), B., De Schepper, K. & Bagnulo, M., "Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture," Internet Engineering Task Force Internet Draft draft-ietf-tsvwg-l4s-arch-00 (May 2017) (Work in Progress)
- [l4s-id] De Schepper, K., Briscoe (Ed.), B. & Tsang, I.-J., "Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay," Internet Engineering Task Force Internet Draft draft-ietf-tsvwg-ecn-l4s-id-00 (May 2017) (Work in Progress)
- [dualq-aqm] De Schepper, K., Briscoe (Ed.), B., Bondarenko, O. & Tsang, I.-J., "DualQ Coupled AQM for Low Latency, Low Loss and Scalable Throughput," Internet Engineering Task Force Internet Draft draft-ietf-tsvwg-aqm-dualq-coupled-01 (July 2017) (Work in Progress)