

Prague Congestion Control Results and Insights

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>

Olga Albisser, Simula

<olga@albisser.org>



ICCRG, IETF-114, Jul 2022

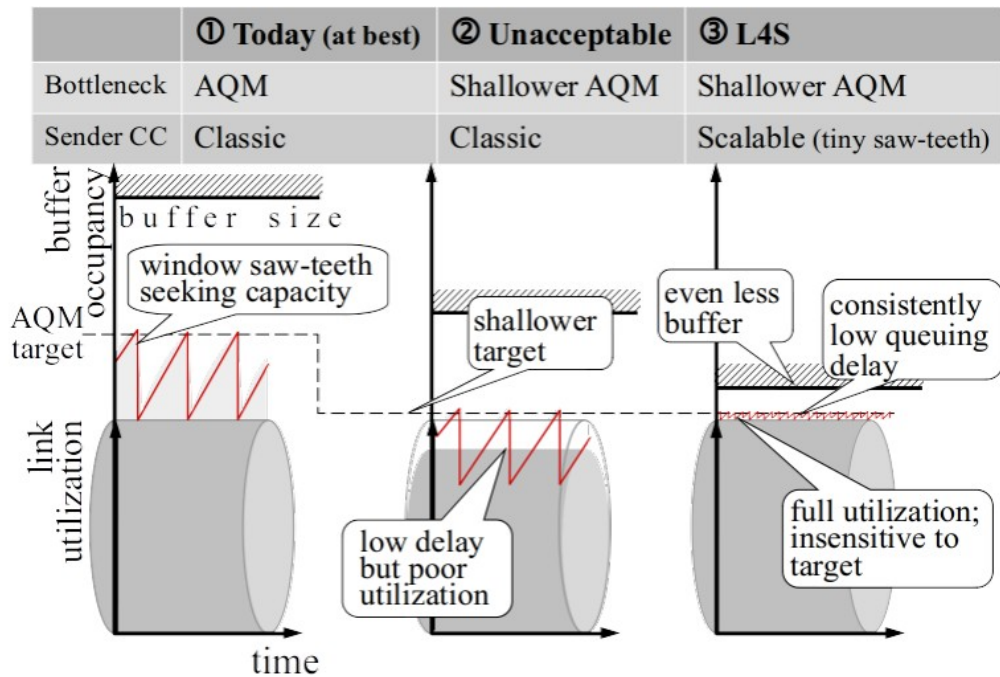
This talk

- Selection of comparative testbed results
 - to share insights
- Test traffic
 - designed to explain effects
 - not to be realistic

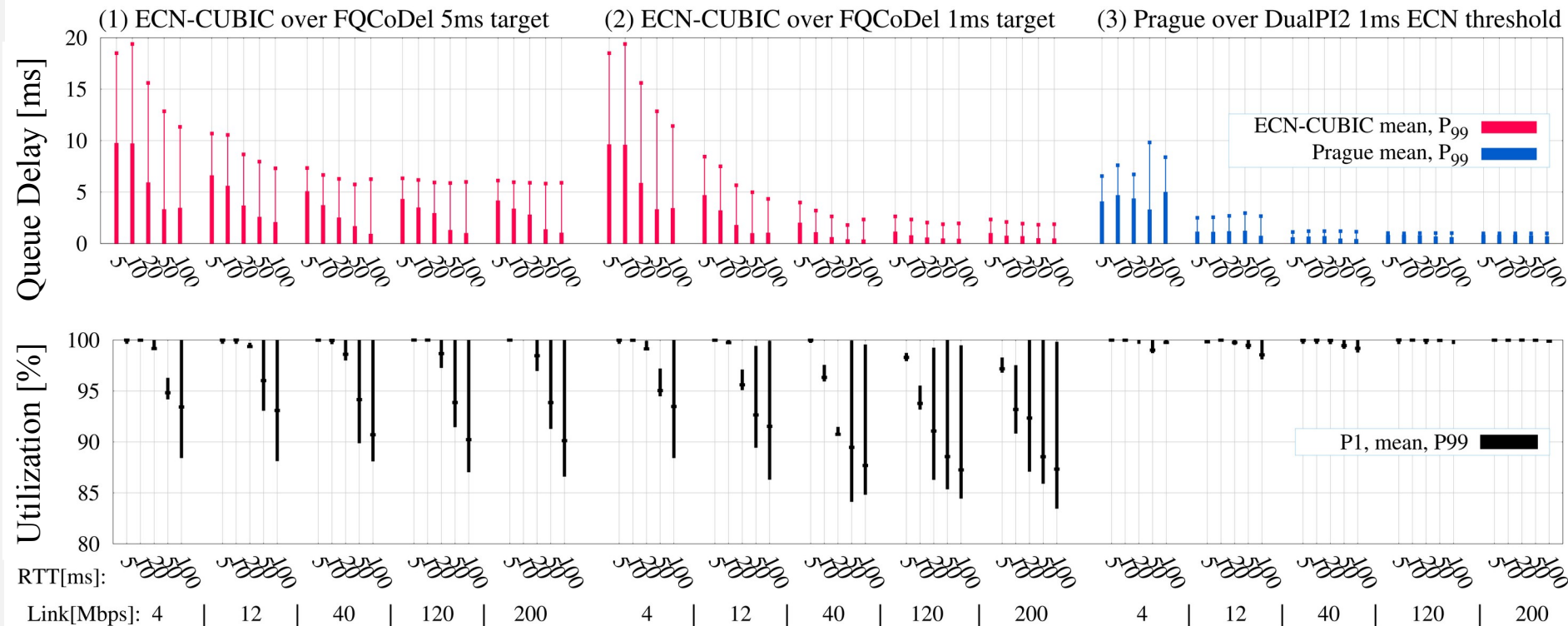
Prague CC: brief recap

draft-briscoe-iccrp-prague-congestion-control-01

- Based on DCTCP
- Functional differences:
 - ECT(1)
 - Accurate feedback,
- Algorithmic differences:
 - AI: No suspension of Additive Increase, but only on ACKs, not NACKs
 - MD: No dead-zone in EWMA
 - Paced
 - Max TSO burst 250 μ s (DCTCP: 1ms)
 - Reduced RTT-dependence



Low Queue Delay *and* High Utilization

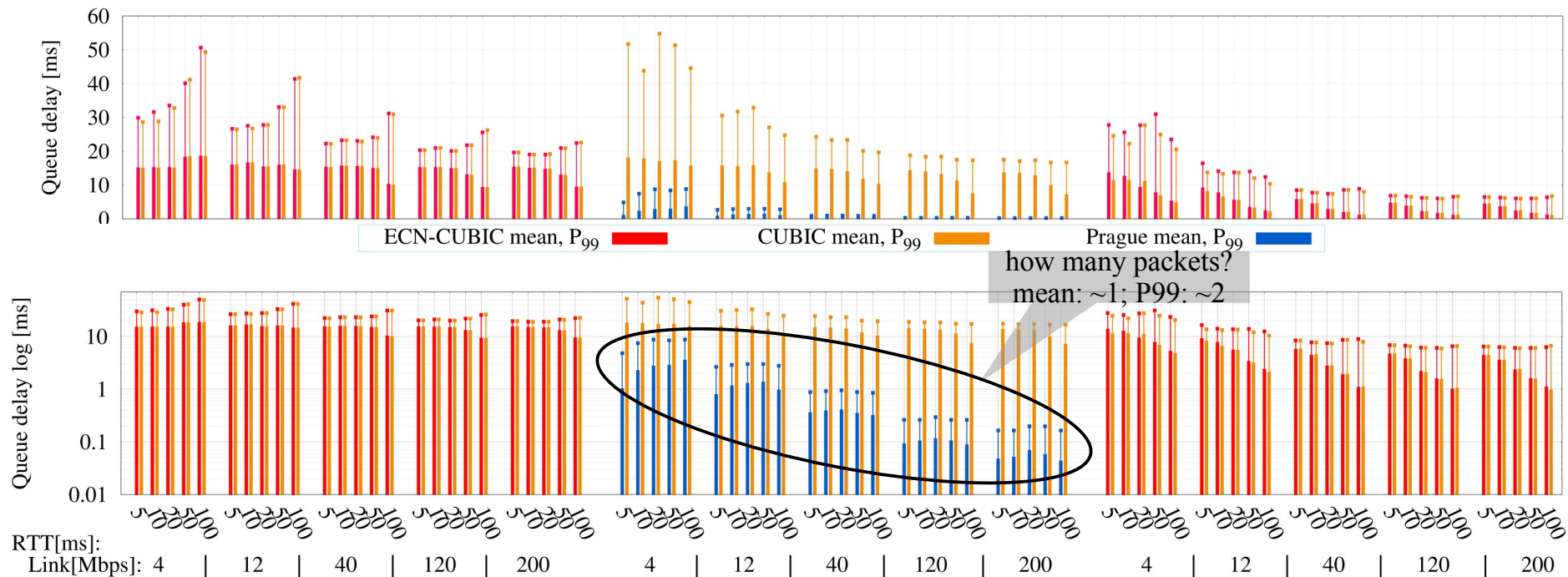


steady-state queue delay; 1:1 flows

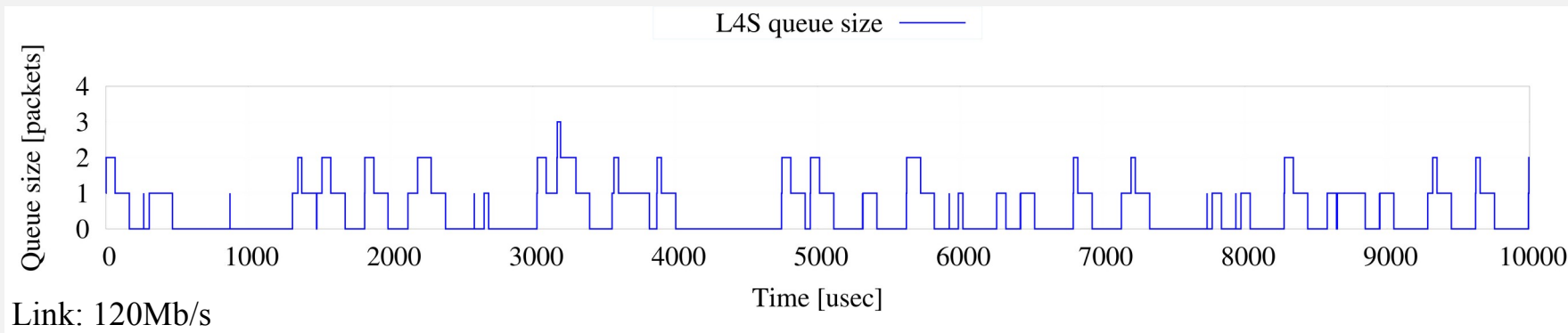
PIE (ECN-CUBIC/CUBIC)

DualPI2 (Prague/CUBIC)

FQ-CoDel (ECN-CUBIC/CUBIC)

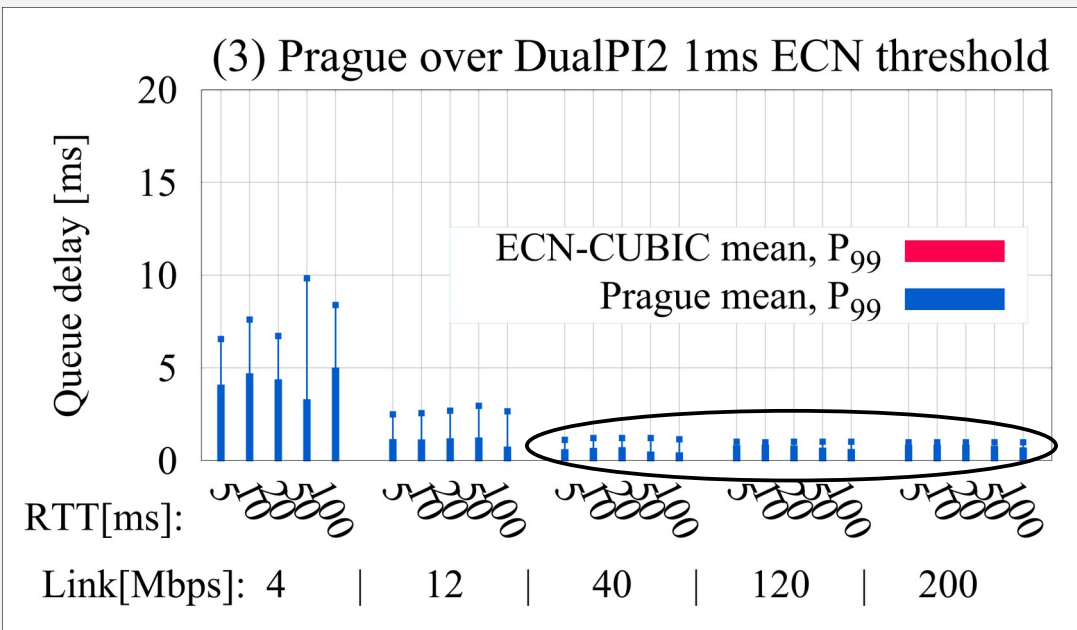


Thanks to: Coupled Marking *and* Pacing



- Coupled marking: increases until Prague leaves enough gaps
 - no standing Q
- Pacing: small bumps with lots of small gaps (not large and few)

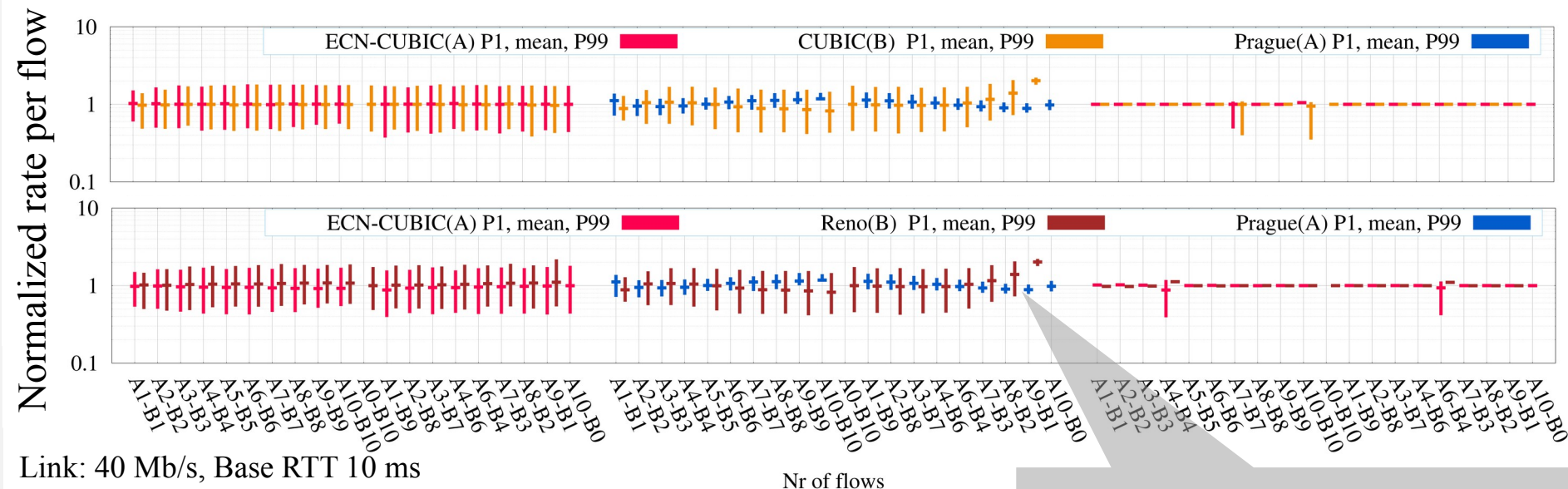
But,... with no Classic flows



- Standing queue at L4S AQM target (1ms)
 - not a problem, just interesting...
- **Insight:**
to prevent a standing queue, control marking from *another* queue
- Examples:
 - coupled marking
 - virtual queue*

* a virtual queue is a number representing what the queue would be, if it were drained slightly more slowly

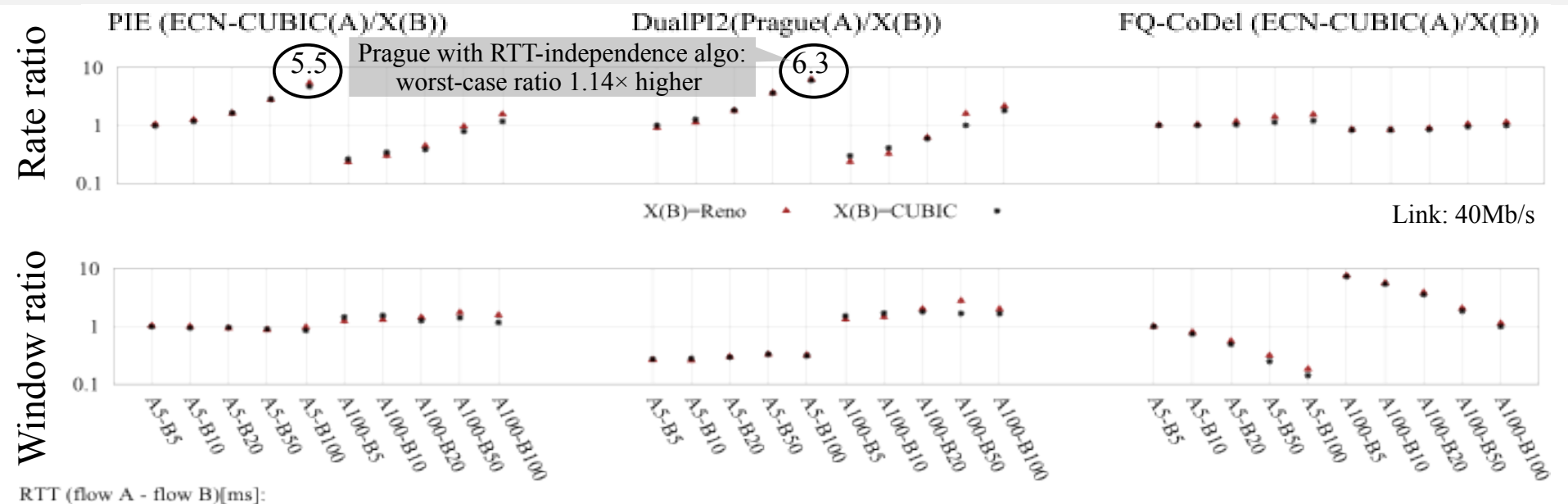
Different no's of steady-state flows for each CC



- PIE: averages are near-perfect but with variance
- FQ: v. low variance, except occasional hash collisions
- DualPI2: Classic variance is like PIE's, Prague averages waiver a little, but less variance
- Reno near-identical to CUBIC

'Normalized rate per flow' := flow rate relative to link/N (for N flows)
 Example: 'A2-B8' means 2 A-type flows and 8 B-type flows

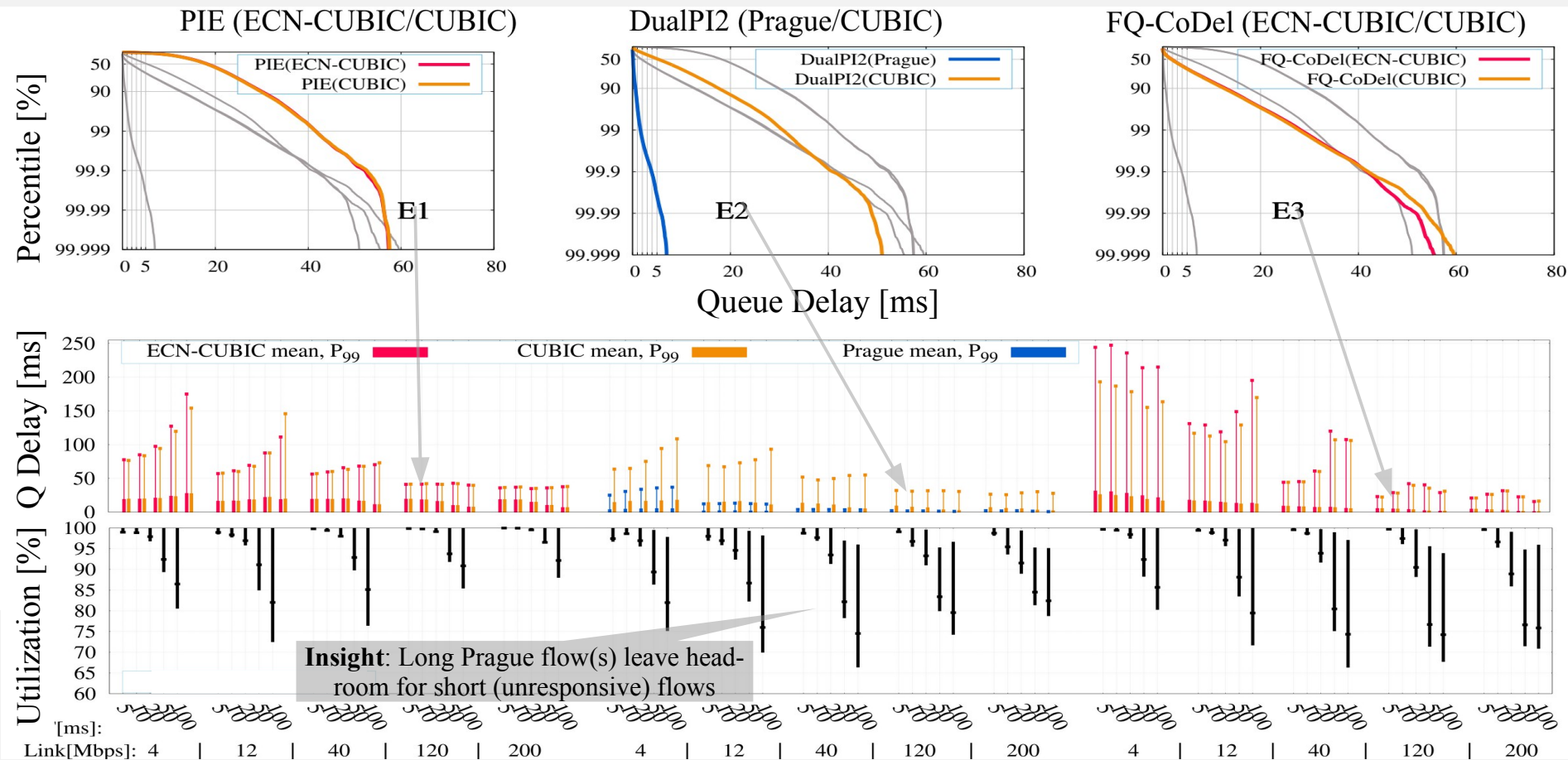
1:1 flows, but mixed RTTs



- CUBIC over FQ: 'rate-fair'
- CUBIC over PIE: 'window fair' so RTT-dependent, but $(R1 + Q)/(R2 + Q)$ is cushioned by $Q=15\text{ms}$; $(100+15)/(5+15)=6$
- RTT-dependence would become problematic as we reduce Q . E.g. $(100+1)/(5+1)=17$
 - So Prague algorithm becomes RTT-independent for $\text{RTT} \leq 25\text{ms}$
- **Insight:** Proper (and sufficient) place to address RTT-dependence in shallow Q s: new low latency senders (e.g. Prague)

Example: 'A5-B100' means RTT of A flow: 5ms; RTT of B flow: 100ms

Heavy web-like load + 1 long-running flow; from both CCs



Take-Home Insights

- To prevent a standing queue in your buffer
 - control marking from *another* queue
- Proper place to address RTT-dependence:
 - newly deployed low latency CCs (like Prague)
- Long Prague flows leave head-room for short ones
- These results have been stable since Jul'19

Where to Get Started

- L4S landing page
 - l4s.net
- TCP Prague mailing list
 - www.ietf.org/mailman/listinfo/tcpprague
- Open source code from L4S team
 - Linux kernel code, testbed scripts and GUI visualizer, Prague virtual machine, ...
 - github.com/L4Steam
 - l4steam.github.io
- ns-3 simulation models (some in mainline, some out-of-tree)
 - Prague, AccECN, DualQ, FQ/CoDel/Cobalt/PIE with L4S support, scenario scripting
 - www.nsnam.org/wiki/L4S-support

Prague Congestion Control

Q&A
and spare slide

Heavy web-like load + 1 long-running flow; from both CCs

