

# Low Latency Low Loss Scalable Throughput (L4S)

draft-ietf-tsvwg-l4s-arch-06

draft-ietf-tsvwg-ecn-l4s-id-10

draft-ietf-tsvwg-aqm-dualq-coupled-12

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>



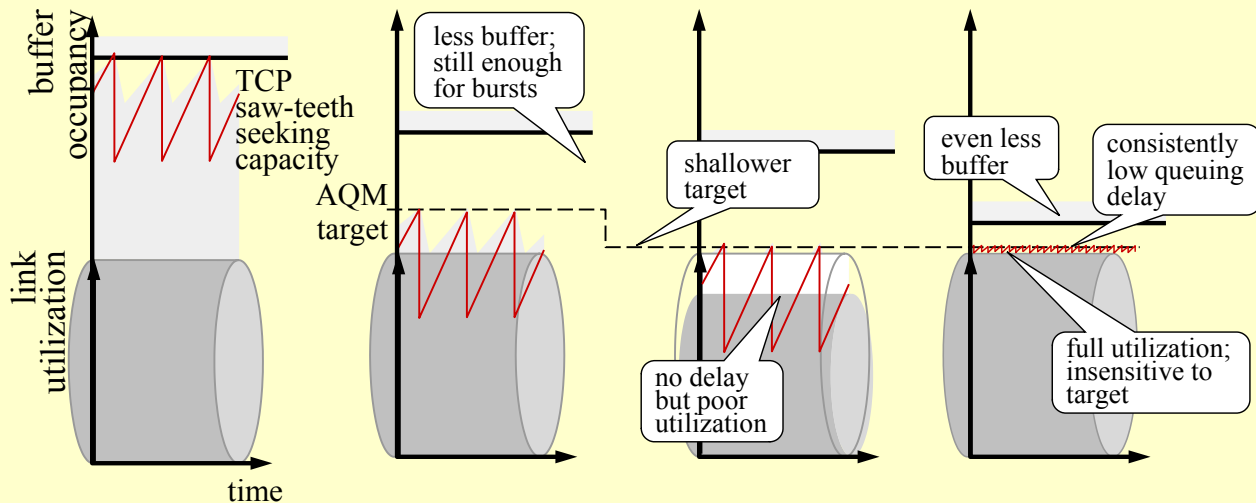
TSVWG, IETF-108, Jul 2020

# Recap – L4S Motivation

- Ultra-low queuing delay for *all* Internet applications
  - including capacity-seeking (TCP-like) and capacity-adaptive

## 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)



# L4S Implementation News

- Broadcom [BCM88800](#) integrated packet processor & traffic manager
  - released – 7<sup>th</sup> gen of Broadcom's StrataDNX™ switching product Line
  - with L4S DualQ Coupled AQM (Curvy RED) support
- Low Latency DOCSIS
  - Interop testing of 3 independent implementations (2 cable modems, 1 CMTS)
  - Not completely passing all functional tests yet (one implementation is)
- Data Plane Development Kit ([DPDK](#))
  - Open source libraries to accelerate packet processing on a variety of CPU architectures
  - DualQ Coupled AQM implementation planned
- ns3 – see next 2 slides (courtesy of Tom Henderson)

# L4S in ns3: status update

- ns-3 TCP Prague implementation now contains:
  - Linux-like pacing
  - paced chirping or Reno flow start
  - RTT independence support
- 3 summer students working on L4S-related native ns-3 models, specifically:
  - TCP Prague (above items, plus AccECN and validation & alignment with Linux ref implementation)
  - L4S-aware FQ/CoDel (completed)
  - FQ/Cobalt with L4S, with which existing ack filtering will be integrated
  - generation of Flent-compatible experiment traces to mirror the experiments written by Pete Heist

# IETF-108 Hackathon report back

- ns3 work
  - students worked on the above (RTT independence, FQ/Cobalt, Flent)
  - Tom Henderson worked on integrating and testing the combined code
  - On track to include most of these components and the dual queue coupled AQM model for ns-3.32 release (Sep'20)
- Ashutosh Srivastava and Fraida Fund (NYU)
  - documentation and scripts on how to experiment with L4S in CloudLab,
  - including reproducing most of the Pete Heist scenarios, and posted on GitHub

see [https://www.nsnam.org/wiki/Sprints#IETF\\_108\\_Hackathon.2C\\_July\\_20-23.2C\\_2020](https://www.nsnam.org/wiki/Sprints#IETF_108_Hackathon.2C_July_20-23.2C_2020)

# L4S Drafts - Status

- draft-ietf-tsvwg-l4s-arch-06:
  - complete AFAICT,
  - except for FQ\_CoDel with L4S support it currently only cites the code - would be great if someone would write a brief update to the FQ\_CoDel draft
- draft-ietf-tsvwg-aqm-dualq-coupled-12:
  - edits to Curvy RED appendix, from Broadcom experience (see earlier).
- draft-ietf-tsvwg-ecn-l4s-id-10:
  - ToDo: summarize Jake's proposed approach as a new subsection of Appendix B
  - Discuss SHOULDs vs MUSTs in Prague L4S requirements (see next slide)

# Prague L4S Requirements – Changes?

- Remain responsive at low RTT (i.e. remove window  $\geq 2$  constraint):
  - MUST → SHOULD (MAY?), because it's a tradeoff:
    - no min window keeps queue delay low
    - min window protects against unresponsive flows
- Fall-back on Classic ECN AQM detection: How about?:
  - For the L4S experiment, implementations MUST be capable of:
    - monitoring for a Classic ECN AQM and logging results
    - falling back to Reno-Friendly if detected
  - each being possible to enable independently
  - Discuss
  - known issue: need to wordsmith to allow for out-of-band monitoring instead

# Relaunched invitation to get involved

- Encourage more implementations, testing, repeatability
- Discussion fora:
  - [tcpPrague@ietf.org](mailto:tcpPrague@ietf.org) L4S CC & transports (not just TCP)
  - [Github.com/L4STeam](https://github.com/L4STeam) code-specific issues
  - [tsvwg@ietf.org](mailto:tsvwg@ietf.org) and transport-specific lists (tcpm, quic, etc.) for standardization discussion
  - [iccr@irtf.org](mailto:iccr@irtf.org) CC (pre-)standardization discussion