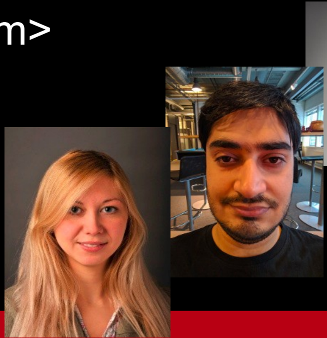
Low Latency Low Loss Scalable Throughput (L4S)

draft-ietf-tsvwg-l4s-arch-08
draft-ietf-tsvwg-ecn-l4s-id-12
draft-ietf-tsvwg-aqm-dualq-coupled-13

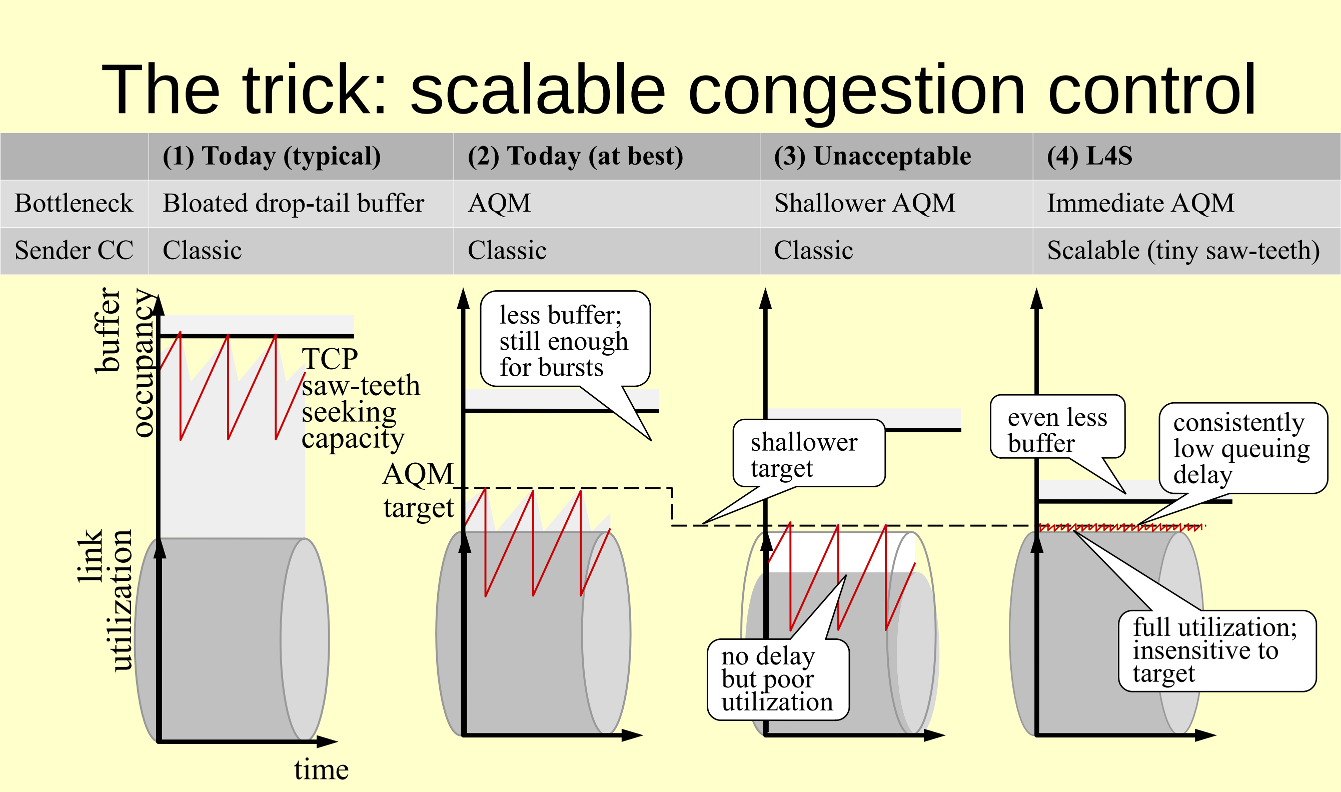
Bob Briscoe, Independent <ietf@bobbriscoe.net>
Koen De Schepper, NOKIA Bell Labs <koen.de_schepper@nokia.com>
Olivier Tilmans, NOKIA Bell Labs <olivier.tilmans@nokia-bell-labs.com>
Greg White, CableLabs <g.white@CableLabs.com>
Asad Sajjad Ahmed, Independent <me@asadsa.com>
Olga Albisser, Simula Research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



TSVWG, IETF-109, Nov 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)

buffer occupancy
link utilization
time
TCP saw-teeth seeking capacity
AQM target
less buffer; still enough for bursts
shallower target
no delay but poor utilization
even less buffer
consistently low queuing delay
full utilization; insensitive to target

L4S Implementation News

- Low Latency DOCSIS
  - Interop testing of 3 independent implementations (2 cable modems, 1 CMTS)
  - Two implementations completely passing all functional tests
- Data Plane Development Kit (DPDK)
  - Open source libraries to accelerate packet processing on a variety of CPU architectures
  - DualQ Coupled AQM implementation work planned Q1-21
- ns-3
  - Low Latency DOCSIS® simulation model published in ns-3 app store. Includes:
    - Low Latency DOCSIS® DualQ Coupled AQM
    - Low Latency DOCSIS® Queue Protection
    - see also CableLabs press release (10 Sep ‘20)
  - L4S support added to ns-3 CoDel and FQ/CoDel models (L4S in FQ/PIE and FQ/Cobalt pending release)
  - Adds to IETF DualQ Coupled AQM model available earlier – could use further testing now

Prague Congestion Control

Heads-up: iccrg session, Fri 05:00 UTC

- Prague gives really low latency over a range of conditions [DCttH]
  - But pieces are still missing
    - so it’s easy to think up conditions where it doesn’t work well
  - We admit progress has been slow
- So we want to generate interest in the problem and the potential
  - explain the interesting changes we’ve had to make to DCTCP and what we’ve learned
- At the end of the talk, we also want to start a conversation
  - After the initial buzz, we imagined CC built in a new way
  - a loose collaboration rewriting components of DCTCP
  - Well, that didn’t happen – core team’s fault, nearly certainly
- Worse, a toxic codepoint war
  now risks overshadowing the potential of any CC using high fidelity signals
- What would a relaunch need to look like, for you to want to get involved?
  CONSTRUCTIVE VIEWS PLEASE

L4S Drafts – Status Summary

- All 3 main L4S drafts are *really* complete (IOHO)
    - other than ecn-l4s-id describing and referencing l4sops (→ previous talk) if nec.
- draft-ietf-tsvwg-l4s-arch-06 to -08 (INF):
    - Re-written throughout to put FQ in its proper place in the architecture
- draft-ietf-tsvwg-ecn-l4s-id-10 to -12 (EXP):
    - The L4S identifier and requirements for using it (host & net)
    - Rejigged SHOULDs / MUSTs in some req's, and added req's on pacing & smoothing
- draft-ietf-tsvwg-aqm-dualq-coupled-12 to -13 (EXP):
    - Already finished – just minor editorial fixes

L4S Architecture: Broadening (I)

draft-ietf-tsvwg-l4s-arch-08 (since -06)

- FQ and DualQ consistently treated as alternatives
  - DualQ no longer *the* architecture
  - cites Sec.5.2.7 of FQ-CoDel [RFC8290] for scalable congestion control support
- Hybrid case added (§4)
  - dual queue with per-flow marking

L4S Architecture: Broadening (II)

draft-ietf-tsvwg-l4s-arch-08 (since -06)

§5.2 “Why Not Alternative Approaches?” →
§5.2 “What L4S adds to Existing Approaches”

- Still covers Diffserv, Classic AQMs, per-flow queuing/marking, and BBR
  - but overhauled to emphasize improvement, rather than critique
- Confined “DualQ vs. per-flow” to two *architectural* comparisons
  - low delay without sacrificing full encryption (e.g. IPSec), if use DualQ
  - L4S can provide low delay whichever way flow rate is controlled
    - FQ: network control
    - DualQ: e2e control
    - DualQ + rate policer: e2e control, but network constrained

L4S Architecture: Additions

draft-ietf-tsvwg-l4s-arch-08 (since -06)

- §6.3 Applicability with Specific Link Technologies (new)
  - L4S addresses queue for e2e congestion control not queue for medium acquisition (etc)
  - Nonetheless, removing longest pole in the tent, focuses attention on the second-longest
- §8.2 Latency Friendliness
  - enumerates 7 types of latency protection (was 1)
- §8.5 Privacy Considerations (new)
  - low delay and full encryption (IPSec, encrypted VPNs) no longer mutually exclusive
  - with only two visible categories little scope for correlation betw. traffic types and users (no need for visible ports, classes, etc.) [RFC8404]

L4S Architecture: Editorial

draft-ietf-tsvwg-l4s-arch-08 (since -06)

- Main improvements
  - Explained saw-tooth scaling rationale of L4S
    - previously relied on ref to ecn-l4s-id
  - Repeatedly emphasize L4S is for all transport protocols and CCs
    - not just for TCP, not just for capacity-seeking
  - Considerable clarification added throughout

L4S ID; Transport Req’s: Updates

draft-ietf-tsvwg-ecn-l4s-id-12 (since -10)

§4.3 Congestion Control Requirements (paraphrased)

- A scalable congestion control MUST implement monitoring to detect a likely non-L4S but ECN-capable AQM

  On detection ... it SHOULD be capable (dependent on configuration) of automatically adapting its congestion response to coexist with Reno

- A scalable congestion control MUST eliminate RTT bias as much as possible in the range between the minimum likely RTT and typical RTTs expected in the intended deployment scenario

L4S ID; Transport Req’s: Additions

draft-ietf-tsvwg-ecn-l4s-id-12 (since -10)

- §4.3 Scalable CC MUST be replaceable by Classic CC
- §4.3 Burst limiting (new)
  - RFC defining specific CC MUST define burst limiting (e.g. pacing)
- §4.4 Sender smoothing of ECN Feedback (new)
  - responsibility shifts from network to sender (e.g. EWMA in DCTCP)
  - Previously only covered in l4s-arch
  - RFCs that define specific CC SHOULD define feedback smoothing

L4S ID; Network Req's: Updates & Additions

draft-ietf-tsvwg-ecn-l4s-id-12 (since -10)

- §5.1 Overload protection
  - Under persistent overload L4S AQM ~~SHOULD~~ MUST use Classic drop
- §5.2 Immediate AQM
  - An L4S AQM SHOULD NOT smooth or filter queue measurements when signalling congestion
  - responsibility for smoothing congestion feedback shifts to the sender
- §5.5 Limiting bursts (new)
  - When implementing L4S AQM, review opportunities to reduce link-layer burstiness (informative)
  - general advice on reducing link-layer bursts in non-AQM nodes too

L4S Identifier Draft: Other Additions

draft-ietf-tsvwg-ecn-l4s-id-12 (since -10)

- L4S Experiments
  - greatly expanded with 3 new subsections

    1) open questions
    2) open issues
    3) future potential
- IANA Considerations
  - Allocation of ECT(1)
- Rationale for choice of identifier (Appx B)
  - Added ECN-DualQ-SCE1 & 0 to alternatives

Status

- IOHO all 3 drafts are ready for WGLC

- More comments on latest normative text changes pls?
- How high should the bar be set for proof-of-concept CC before L4S docs progress to IESG?
- How mature does l4sops have to be before L4S drafts progress to IESG?

L4S

Q&A