L4S Status Update

draft-ietf-tsvwg-l4s-arch-10
draft-ietf-tsvwg-ecn-l4s-id-18++
draft-ietf-tsvwg-aqm-dualq-coupled-16

Bob Briscoe, Independent
Koen De Schepper, NOKIA Bell Labs
Greg White, CableLabs

<ietf@bobbriscoe.net> <koen.de_schepper@nokia.com> <g.white@CableLabs.com>

tsvwg, IETF-111, Jul 2021
Recap – L4S Motivation

• Very low queuing delay for all Internet applications

• including for capacity-seeking & capacity-adaptive

<table>
<thead>
<tr>
<th></th>
<th>(1) Today (typical)</th>
<th>(2) Today (at best)</th>
<th>(3) Unacceptable</th>
<th>(4) L4S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bottleneck</td>
<td>Bloated drop-tail buffer</td>
<td>AQM</td>
<td>Shallower AQM</td>
<td>Immediate AQM</td>
</tr>
<tr>
<td>Sender CC</td>
<td>Classic</td>
<td>Classic</td>
<td>Classic</td>
<td>Scalable (tiny saw-teeth)</td>
</tr>
</tbody>
</table>

The trick: scalable congestion control

- Less buffer; still enough for bursts
- Shallower target
- Even less buffer
- Consistently low queuing delay
- Full utilization; insensitive to target
- No delay but poor utilization
- Scalable (tiny saw-teeth)
Recent L4S Progress

• 3GPP L4S activity:
  • Currently favour L4S 5QIs with feedback & forward rather than adding ECN bits to RLC header
  • New KTH/Ericsson paper in (VTC'21) with simulated 5G performance evaluations of L4S: Low Latency Low Loss Scalable Throughput in 5G Networks
  • New joint Ericsson - DT white paper Enabling time-critical applications over 5G with rate adaptation

• One more cable modem certified for Low Latency DOCSIS (supports L4S)
• Nokia has more WiFi (+ soon Fixed-Wireless Access) devices supporting L4S
  • initiated several trials with operators and application providers
• Pete Heist's continuing testing & evaluation – see Greg's l4sops talk (next)
• Linux patches for AccECN TCP feedback https://github.com/L4STeam/linux
  • now works with all CC modules (BBRv2, DCTCP, TCP Prague, Cubic, Reno, etc)
  • enables A/B testing without changing more than one thing at a time – see tcpm talk
Draft updates: L4S Architecture
l4s-arch-10 (1 Jul)

• To address Vidhi Goel's review
  • Explained how an FQ-* L4S node works
  • Fixed the numerical examples of Reno & Cubic scaling; including Cubic's Reno-Friendly mode
  • Extensive clarification, precision and improvements
  • See list discussion. Thank you Vidhi.
Draft updates: ECN Protocol for L4S  
ecn-l4s-id-18++ (1 Jul)

• Interaction with VPN anti-replay protection:
  • Solely VPN configuration solutions
  • Rather than VPN implementation changes - out of scope for L4S

• RFC4774 citation placed in better context

• Editorial changes, esp. consistency betw. drafts

• Non-changes:
  • Disable tunnel resequencing of IP data for L4S (e.g. L2TP)
    - No need to mention, after consulting with int-area & pals
    - No evidence that sequencing is enabled for IP data
  • Prague requirements text (§4): pretty stable since developer survey
Draft updates: DualQ Coupled AQMs for L4S aqm-dualq-coupled-16 (6 Jul)

- in-depth justification for the default values of the parameters of the PI2 AQM for Classic traffic
  - To address Sebastian Möller's request
  - Summary of new PI2 Parameters paper (9pp of rationale, data and maths on default 'target')
  - Fuller justification for other parameters, Tupdate, RTT$_{\text{max}}$
  - All in the DualPI2 appendix (A.1)
Interesting data point (and relevant to the 'S' in L4S)

Reno Resurgence

- Where CDNs & AQMs pull down RTT
  - Cubic stays in Reno mode
  - nearly everywhere

CDN RTT under load [ms]
Base RTT + PI2 15ms target

Cubic switchover [ms]
$R > (3.5/r)^{0.4}$
L4S Status Update

Thank you to all those still contributing to list discussion

Q&A → after l4sops talk
Next Steps

• Expected work in parallel to approval process:
  • Scalable CC algorithm improvements,
    - esp. flow start and integrating delay with ecn & loss metrics
  • Progressing l4sops
  • Reporting performance results from L4S experiments
  • Tracking deployment status of L4S