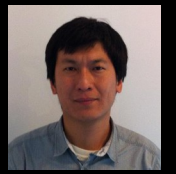
Low Latency Low Loss Scalable Throughput (L4S)

Bob Briscoe, CableLabs <ietf@bobbriscoe.net>
Koen De Schepper, NOKIA Bell Labs <koen.de_schepper@nokia.com>
Olga Albisser (née Bondarenko), simula <olgabnd@gmail.com>
Ing-jyh (Inton) Tsang, NOKIA <ing-jyh.tsang@nokia.com>

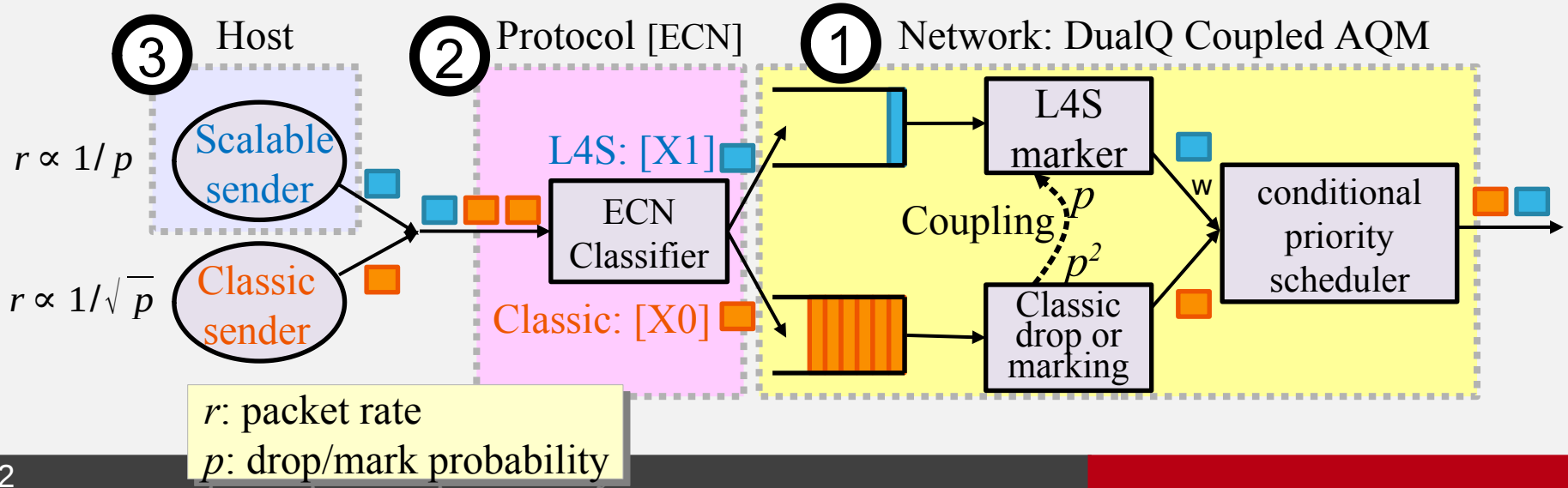


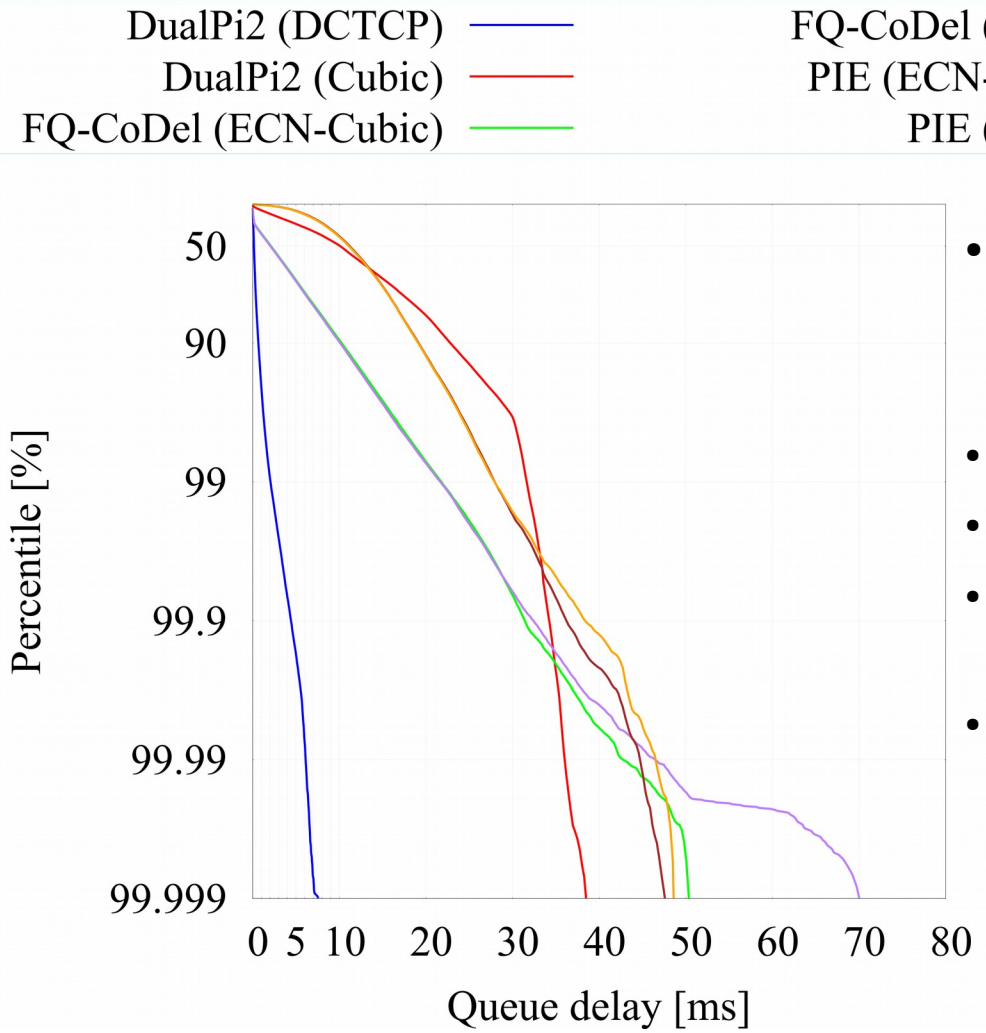
TSVWG, IETF-104, Mar 2019

L4S Recap

- Motivation
  - Extremely low queuing delay for *all* Internet traffic, including link saturating (TCP-like)
  - already 1-2 orders better than state of the art
  - 100-200 µs vs 5-15 ms (fq-CoDel or PIE)
- Architecture



Performance



- Low delay important at higher %-iles
  - for low latency real-time delivery
- 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

Implementation status

pasted from https://riteproject.eu/dctth/#code

Source Code

- Dual Queue Coupled AQM
  - with PI2: Linux repo
  - With Curvy RED (TBA)
- TCP Prague
  - Linux repo
- QUIC Prague
  - General repo (should work for Linux, FreeBSD, Windows)
- SCReAM (Self-Clocked Rate Adaptation for Multimedia) a mobile optimised congestion control algorithm for real-time interactive media, with support for L4S
  - General repo

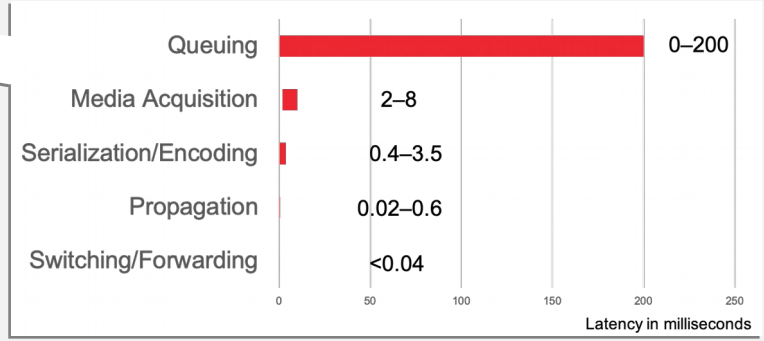
+DOCSIS 3.1 (next slide)

- Component parts
  - Accurate ECN TCP Feedback (included in TCP Prague above)
    - Linux repo and Linux repo without AccECN TCP Option
  - Paced Chirping
    - for Linux (initial proof-of-concept research code)
  - Data Centre TCP (DCTCP) for
    - Linux (in the mainline kernel)
    - FreeBSD (in the mainline kernel)
    - ns2 patch.

particular thanks to Olivier Tilmans
for pulling together TCP Prague and the Hackathon team

Low Latency DOCSIS 3.1

- Low Latency measures mandatory from Jan'19
  - upstream (Cable Modem) & downstream (CMTS)
    - DOCSIS 3.1 MAC and Upper Layer Protocols i/f (MULPI) Spec (i17+)
    - Cable Modem Operations Support System Interface Spec (i14+)
    - CCAP Operations Support System Interface Specification (i14+)
- Cuts 2 main sources of delay
  - MAC: Request-grant loop
  - Queuing: **Mandatory L4S support**
- White paper: Low Latency DOCSIS: Technology Overview
  - Also translated into ASCII: draft-white-tsvwg-lld (Informational)
- Certification test plans nearing completion
- Implementation in progress



Reviews this IETF cycle

ecn-l4s-id (full)

- Nicolas Kuhn
- Gorry Fairhurst x2
- Richard Scheffenegger

ecn-l4s-id (focused)

- Michael Abrahamsson
- Ingemar Johansson
- Praveen Balasubramanian
- David Black

aqm-dualq-coupled (full)

- David Pullen
- Greg White

aqm-dualq-coupled (focused)

- Gabi Bracha

non-supportive

- Jonathan Morton
- Dave Täht

questioning codepoints

- Roland Bless
- Jake Holland

(many apologies if you've contributed a review and I've omitted you)
no implication that reviews not categorised as non-supportive are supportive

Technical issues #1:
Classic bottleneck

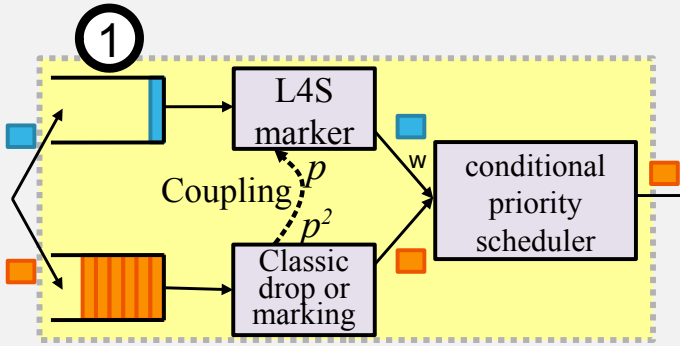
Drop detection

- fixed Linux DCTCP bug for TCP Prague 2yrs ago
  - no response to fast re-xmt, only RTO
  - compound reduction of ECN and loss: halves
  - returns to ECN EWMA after loss episode
- submitted DCTCP patch
  - now picked up by Yuchung Cheng & Larry Brackmo
- loss detection in time units
  - clarified: links will only relax ordering up to most ssensitive transport

Classic CE detection

- raised implementation/test priority – was lowest
  - all studies except Apple's, no evidence of appreciable CE on Internet
  - Apple 2017 data
    - Large numbers of Apple devices (e.g. 30% Argentina) at least 1 CE in 12 hrs
    - helping dig into their ongoing stats
  - Jan 2019, Trammel: still little sign (~13 CE)
- if CE is solely from FQ, no problem
  - designed test to distinguish FQ v. FIFO CE

Technical issues #2:
Scheduler misconceptions



- WRR scheduler?
  - coupling negates bandwidth priority (but not latency priority)
- FQ as alternative to DualQ
  - emphasis on DualQ has eclipsed this L4S solution
- Technical non-issues
  - text needed in l4s-arch to explain both

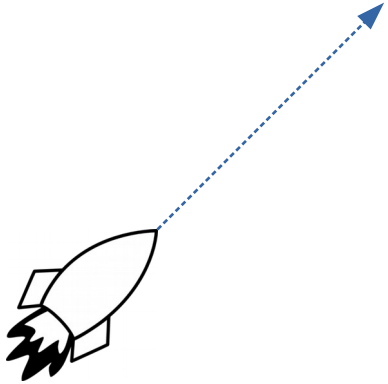
Intellectual Property

- Nokia IPR declaration on DualQ Coupled AQM
  - default FRAND terms
  - GPLv2 licence on Linux implementation

- IPR being addressed through WG chairs
  - availability of work-rounds (e.g. alternative DualQ AQMs, FQ)
  - and/or WG liaison with Nokia

Next Steps for 3 core L4S drafts

- IPR
- Classic ECN bottleneck
- Minor text updates to all three



- Once satisfactorily resolved (ASAP), WGLC all three
- L4S experiment can start

Low Latency Low Loss Scalable Throughput (L4S)

Q&A

L4S status update: IETF specs (2/2)

Deltas since last IETF in Red

tsvwg

- L4S Internet Service: Architecture <draft-ietf-tsvwg-l4s-arch-03> [stable]
- Identifying Modified ECN Semantics for Ultra-Low Queuing Delay (L4S) <draft-ietf-tsvwg-ecn-l4s-id-05> [2 UPDATES]
- DualQ Coupled AQMs for L4S: : <draft-ietf-tsvwg-aqm-dualq-coupled-08> [2 UPDATES]
- Interactions of L4S with Diffserv <draft-briscoe-tsvwg-l4s-diffserv-02> [UPDATE]
- Identifying and Handling Non-Queue-Building Flows in a bottleneck link draft-white-tsvwg-nqb-00 [NEW]
- enabled by <RFC8311> [RFC published]

tcpm

- scalable TCP algorithms, e.g. Data Centre TCP (DCTCP) <RFC8257>, TCP Prague
- Accurate ECN: <draft-ietf-tcpm-accurate-ecn-07>
- ECN++ Adding ECN to TCP control packets: <draft-ietf-tcpm-generalized-ecn-03> [UPDATE]

Other

- ECN support in trill <draft-ietf-trill-ecn-support-07>, motivated by L4S [RFC Ed Q]
- ECN in QUIC <draft-ietf-quic-transport-16>, [motivated by L4S – 3 Updates, but not ECN part]
- ECN and Congestion Feedback Using the Network Service Header (NSH) <draft-eastlake-sfc-nsh-ecn-support-02> [UPDATE] [supports L4S-ECN]

ECN transitions

- RFC3168 & RFC8311
  - ECT(0) → CE
  - ECT(1) → CE
- RFC6040 added support for RFC6660
  - ECT(0) → ECT(1)
- Many encapsulations will still be pre-RFC6040
  - decap will revert ECT(1)
- Ambiguity of CE
  - ECT(0) → CE early on path
    CE → L4S queue later on path
  - 5 unlikely scenarios have to coincide
    to cause an occasional spurious re-xmt

incoming inner	incoming outer			
	Not-ECT	ECT(0)	ECT(1)	CE
Not-ECT	Not-ECT	Not-ECT	Not-ECT	drop Not-ECT
ECT(0)	ECT(0)	ECT(0)	ECT(0)	CE
ECT(1)	ECT(1)	ECT(1)	ECT(1)	CE
CE	CE	CE	CE	CE

Outgoing header (RFC4301 \ RFC3168)

incoming inner	incoming outer			
	Not-ECT	ECT(0)	ECT(1)	CE
Not-ECT	Not-ECT	Not-ECT	Not-ECT	**drop**
ECT(0)	ECT(0)	ECT(0)	**ECT(1)**	CE
ECT(1)	ECT(1)	ECT(1)	ECT(1)	CE
CE	CE	CE	CE	CE

Outgoing header (RFC6040)
(bold = change for all IP in IP)