L4S TCP-Prague

ICCRG IETF98 Chicago

March 27th, 2017

Koen De Schepper

Bob Briscoe

Olga Bondarenko

Inton Tsang

DualQ for DCTCP

DualQ AQM was main focus up to now

- Classic and DCTCP window compatibility
- PI2 as the classic AQM
- Overload handling
- Large number of experiments: flow numbers, RTTs, dynamic flows, overload

L4S - DualQ concept proven, usable with DCTCP

- Adoption of 3 drafts in TSVWG
- Linux open source released, mainline release ongoing

Recent focus on TCP-Prague

Internet-safety:

- 4.1: Fall back to Reno/Cubic congestion control on packet loss
- 4.2: Fall back to Reno/Cubic congestion control on classic ECN bottlenecks
- 4.3: Reduce RTT dependence
- 4.4: Scaling down the congestion window
- tcpm: Accurate ECN and negotiation draft-ietf-tcpm-accurate-ecn

Performance improvements:

- 5.1: Setting ECT in SYN, SYN/ACK and pure ACK packets
- 5.2: Faster than additive increase
- 5.3: Faster convergence to fairness

implemented work in progress

Prevent marking probability saturation

4.4: Scaling down the congestion window

Range p = [0 .. 1]

Range 1/p = [1 .. infinite]

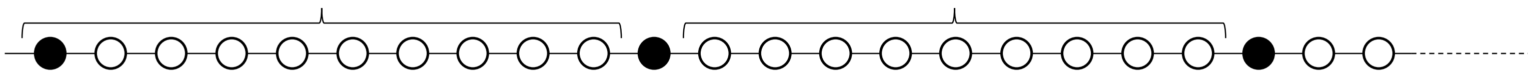
Rate should range from [0 .. infinite] → 1/p - 1

Solution: Average unmarked packets between marks

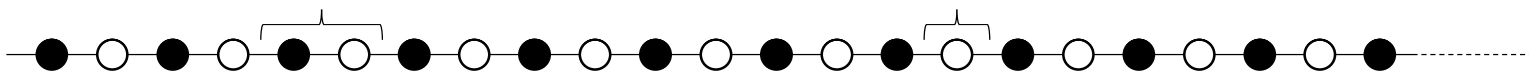
$$u = 1/p - 1 = (1-p) / p$$

Average unmarked:

p = 10%   1/p = 10   u = 9   = 1/p – 1

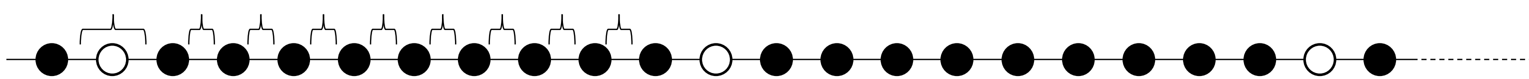


p = 50%   1/p = 2   u = 1

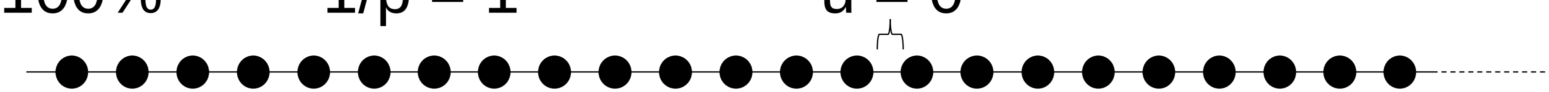


p = 90%   1/p = 1,111   u = 0,111

avg(1 0 0 0 0 0 0 0 0) = 0,111



p = 100%   1/p = 1   u = 0



Marking probability saturation

Drop based rate is also reduced by the dropped packets:

$$r_{drop} = (1 - p) / p.RTT$$

$$\text{unified:} \qquad r \sim= u / RTT$$

Helps for

- scaling the congestion window down
- better drop compatibility
- solving RTT independence

4.3: Reduce RTT dependence

In Classic TCP, big queues → less RTT dependent:

RTT1 = 100 ms + 20 ms queue delay = 120 ms

RTT2 = 1 ms + 20 ms queue delay = 21 ms

Rate ratio = 120/20 = **6x less throughput** for flow with 100ms RTT

L4S has small or no queues at all → high RTT dependence

RTT1 = 100 ms + 1 ms queue delay = 101 ms

RTT2 = 1 ms + 1 ms queue delay = 2 ms

Rate ratio = 101/2 = **50x less throughput** for flow with 100ms RTT

Marking rate & probability

Marking probability p

- Equal for all flows
- Used to converge to equal window or rate

Marking rate m = p.rate

- Depends on the rate too
- Is the signal frequency, which is indication for level of delay control

Question for ICCRG

Compromise between:

- RTT independence with RTT_{ref} = 2ms:

  $r = 2 / p.RTT_{ref} = 1000 / p \rightarrow p.r = 1000$

  ✓ always 1000 marks per second

  ✗ not scalable to small RTTs

- RTT scalability:

  $r = 2 / p.RTT \rightarrow p.r = 2 / RTT$

  ✓ always 2 marks per RTT

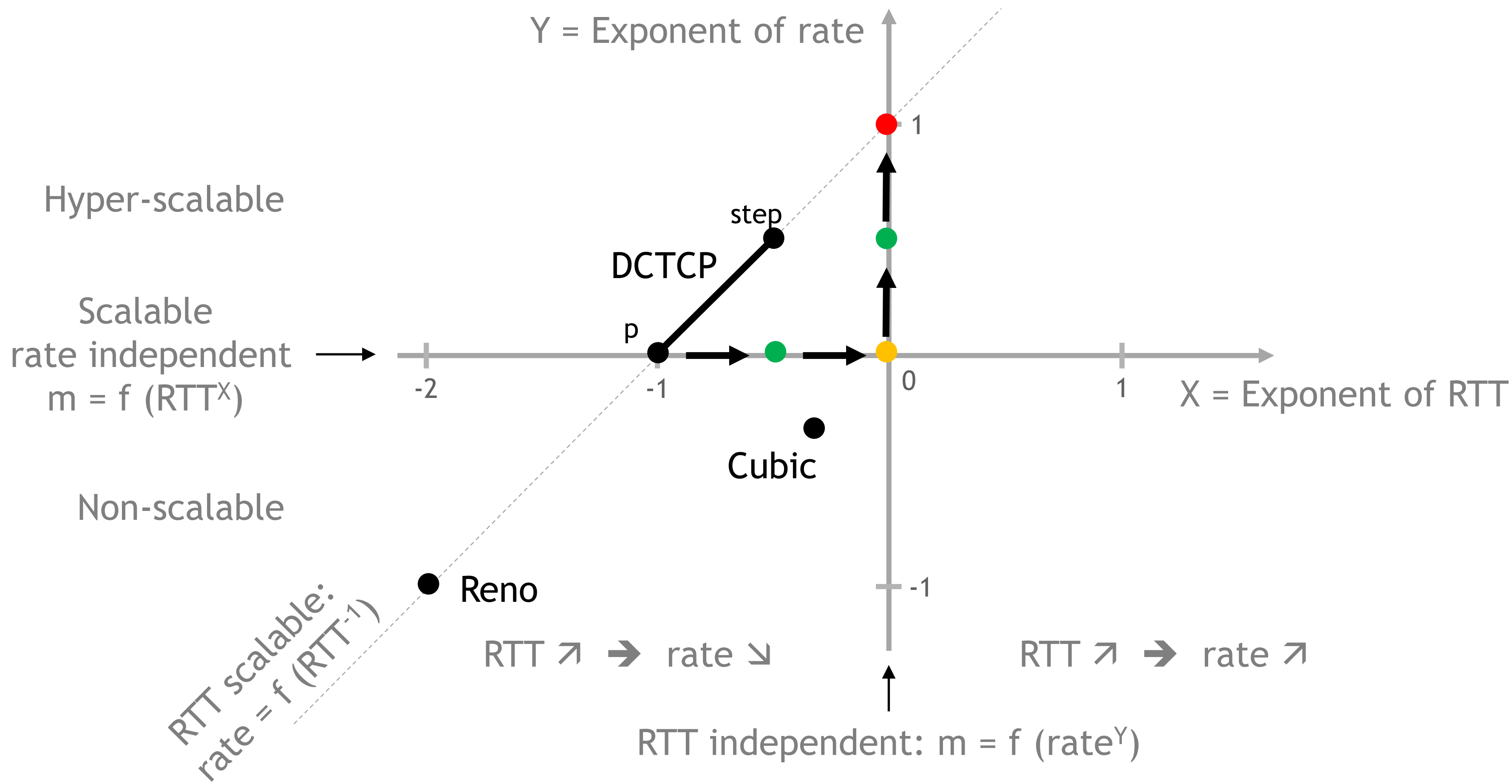
  ✗ rate is very RTT dependent

Where is the right compromise?

	RTT indep.	RTT scalable	rate converg.
Current DCTCP: 2 marks per RTT	✗	✓	✓
Less dependent: f(RTT) marks per RTT • The higher the RTT the more marks per RTT	✓	✓	✓
Full RTT independence: • Constant marks per second (eg: 1 mark per ms)	✓	✗	✓
• The higher the rate the more marks per ms	✓	✓	✓
Full RTT scalability and RTT independent: • Constant marking probability at all rates	✓	✓	✗

Where is the right compromise?

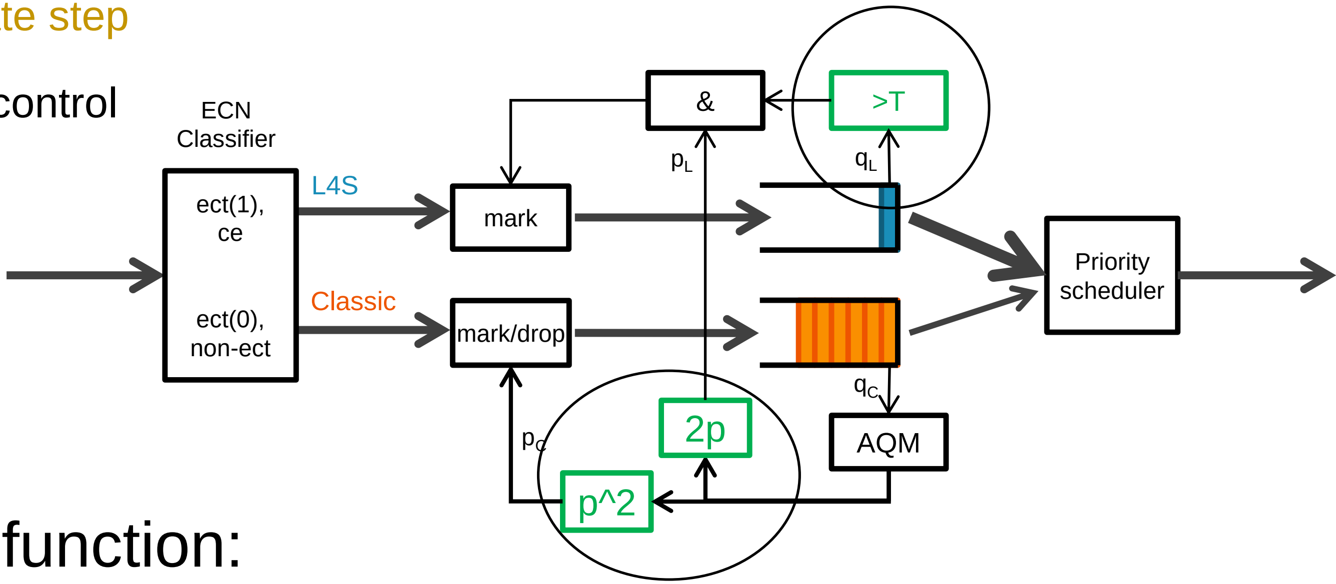
m = steady state marking rate = p * rate = $f(RTT^X, rate^Y)$



Related DualQ discussion topics

L4S-only AQM:

- DCTCP-like immediate step
- AQM with gradual p control



DualQ Coupling function:

- Classic TCP-fairness is well known: 1/sqrt(p)   but future?
- Also coupling is determined by how DCTCP / TCP-Prague behaves
- RTT-independent related coupling

work in progress

Conclusion

L4S - DualQ concept proven and usable with DCTCP

- Low latency and low loss with window-fairness to classic Reno, Cubic, …

L4S: opportunity for new/existing improvements

- What other improvements can we bring to the Internet together with L4S - DualQ?
- Limited opportunity if tsvwg drafts go for last call

Think and discuss about RTT fairness: supporting paper, design team?

Next meeting in Prague: TCP-Prague implementations?

Questions

koen.de_schepper@nokia.com