L4S Issues Related to CE Ambiguity

• #16, #17, #20, #21, #22
Issue #16
L4S - Interaction w/ 3168-only ECN [FIFO] AQMs

- Should still remain open, but making good progress on 3168-only AQM detection (see previous presentations)
- Prevalence moot if solution works;
  Solution moot if no prevalence
- Detailed solution design posted Nov'19 timeframe. Implemented and being evaluated
- Working well distinguishing DualPI2 and CoDel
  - CoDel is our most stringent test, given Qdelay is lowest
Issue #21
CE codepoint semantics

• General concerns about CE ambiguity
• Two specific concerns have been raised
  • #16 L4S - Interaction w/ 3168-only ECN [FIFO] AQMs
    • being actively addressed
  • #22 Deployment feasibility, including incremental (which is about a case where re-ordering can occur)
    • been addressed
• No need for this generic placeholder issue as well?
Issue #20
Objection to ECT(1) codepoint usage

● “If ECT(1) is used for L4S ID, there should be a clear understanding of to what extent this precludes experimenting with SCE”
  ● The question here should not be whether the L4S precludes SCE, but whether there's anything the community might want from SCE that it can't get from L4S
  ● Any discussion of arrangements for parallel experiments depends on that
Issue #24—Evaluation & testing results

• Issue summary:
  • Questions about testing results and scenario’s
  • Problems with availability of code to test and reproduce results

• Prague and DualPI2 code upgraded to latest kernel versions (full kernel tree available too)
• [1] Shows excerpt of the test suite run on the dualIQ/TCP Prague:
  • Per packet measurement to witness actual tail latencies instead of using coarse-grained/smoothed estimates
  • Scenarios mixing:
    • Various bottleneck bandwidth (4-200M) and base RTTs (5-100ms)
    • Variable number of long-running flows, with Prague/Cubic/Reno/BBR(v2)
    • Dynamic load—from 10 to 500 web request per sec, downloading objects from 1kb to 1MB
    • Unresponsive UDP flow in either queue (overload experiments)
    • Mixed RTTs experiments
• [2] Tests reusing P. Heist’s scenarios

• Proposal:
  • Close

- Experiment made over a bottleneck of 120Mbps
- Per packet measurements during 5 mins
- Mixture of two long running flows and 200 web request/s
Issue #28—DualQ suitability

• Issue summary:
  • Claim that “DualPI2 needs to make sure that RTT-based unfairness is removed”

• Questions:
  • Is the goal of an AQM to police/verify RTT fairness between flows? Have these requirements also been imposed on PIE, CoDel, RED, ...?
  • Shouldn’t the end-systems address the issues their behavior creates?
  • Is the end-system principle not applicable here? (Good design is to prefer end-system functionality above network functionality)
  • Possible AQM solutions are undesirable
    • provide per packet RTT information (no headers available)
    • Set Classic PI2 target to 1ms also (Classic underutilization and high drop probability)
    • Increase coupling factor to compensate worse case RTT ratio (L4S gets lower throughput in all other cases)

• Proposal:
  • Not an issue of the AQM (additional policers will be added on if-needed basis)
  • Prague CC is updated with definable target RTT mapping function (code released soon)
  • RTT-independence solves many issues on the internet as smallest seen base RTTs and queue latencies become smaller and smaller
SCE-L4S Bakeoff Scenario 1 (SCE)
Sender → SCE middlebox 1q (bottleneck) → SCE Receiver
b:sce-s1-1 cc:cubic q:cake(50Mbit 1q) bw:50Mbit rtt:80ms

L4S fluent tests (Scenario 1)
Sender → L4S middlebox (bottleneck) → L4S Receiver
b:l4s-s1-1 cc:cubic q:htb(50Mbit)+dualpi2 bw:50Mbit rtt:80ms

[2] https://l4s.cablelabs.com/l4s-testing/key_plots/batch-l4s-s1-1-cubic-50Mbit-80ms.png