REDUCING INTERNET TRANSPORT LATENCY

AQM Evaluation Criteria and Scenarios

Naeem Khademi <naeemk@ifi.uio.no>
Amadou B. Bagayoko <Amadou.Bagayoko@telecom-bretagne.eu>
Gorry Fairhurst <gorry@erg.abdn.ac.uk>
Chamil Kulatunga <chamil@erg.abdn.ac.uk>
David Ros <David.Ros@telecom-bretagne.eu>
Michael Welzl <michawe@ifi.uio.no>

AQM WG – IETF 88
Vancouver, BC, Canada

Outline

AQM and Bufferbloat

- Two very recent proposals ((FQ_)CoDel (IETF 84) and PIE (IETF 85)) aim to mitigate latency
- First AQM algorithms were proposed in early 90's and 00's (*RED, REM, BLUE, CHOKe,...)
- RED's main goals (from abstract of original paper):
  - Low avg queue size, allow occasional bursts
  - Probability of notifying a flow roughly proportional to its rate
  - Break synchronization among TCP flows
- AQM charter contains all these things + "help sources control their rates without unnecessary losses, e.g. through ECN"

AQM Evaluation Criteria (Metrics)

- Latency vs. utilization trade-off
  - Link utilization
  - Queuing delay (ms) and queue length (packets or bytes)
    - mean, median, and upper/lower quantiles
- Packet loss
  - long-term rate/probability
  - pattern (loss inter-arrival time and distribution)
- Jain's fairness index
- Synchronization metrics
- ...

Discussion: do we need...?

- Flow completion time (application layer delay)
- MOS (or similar) for VoIP or other multimedia apps

AQM Parameter Sensitivity

- All AQMs keep a set of parameters
- Need to understand their impact
- Start with a simple “baseline scenario” (e.g. single TCP flow) and evaluate under different congestion levels

Examples of AQM Parameters

Parameter	PIE	CoDel	ARED
Target delay	20 ms	5 ms	$(th\_min + th\_max)/2$
Update interval	30 ms	100 ms	500 ms
(α, β)	(0.125,1.25)	N/A	$(min(0.01, p_{max}/4), 0.9)$

Note: Entirely different semantics for update interval and (α, β)

AQM Parameter Sensitivity – cont.

- Packet-mode vs. Byte-mode
- Head-drop vs. Tail-drop

Sub-RTT Burst Absorption

- Queue as a shock absorber but often inflated for maximizing utilization
- Impact of buffer size and burst allowance on AQM performance
- Micro bursts vs. Macro bursts
  - PHY rate mismatch
  - IW10
  - HTTP mice
  - bursty video frames (H.264/AVC)
  - Financial data traffic?
- To what extent bursts cause TCP loss synchronization?

RTT Sensitivity and Fairness

- TCP dynamics as a driving force for AQM design
- Worst-case RTT design
  - (FQ_)CoDel postpones marking/dropping for 100 ms when it enters *dropping mode*
- Important to evaluate against a set of RTTs (from data centers to satellite links)
  - {1 ms, 5 ms, 20 ms, 100 ms, 500 ms, 1000 ms}

Fluctuating Bandwidth

Better Metrics Used?

(est. or act.) queuing delay vs. (average) queue size

PHY/MAC Scenarios

- **ADSL2+ modems** (up to 24.0 / 1.4 Mbps DL/UL)
- **DOCSIS 3.0 CMs** (at least 171.52/122.88 Mbps DL/UL, 4 CHs)
- **802.11 APs** (different modulation and coding schemes)

Tests

- Downlink/Uplink asymmetry
  - 802.11-DCF's impact on AQM (w/ bulk uplink TCP)
  - 802.11 L2 RA (*SampleRate* in FBSD, *Minstrel* in Linux)
- ACK loss with AQM on the reverse path

Extremely Low Target Delays (Data-Centers)

- How do AQMs perform when target_delay$\leq$1 ms on 1$\sim$10 Gbps links and RTT_{base}=1$\sim$2 ms?
- Parameter tuning most likely required e.g. PIE/ARED's (α,β)

Limitations

Kernel clock granularity is a limiting factor

- Linux kernel Hz=1000
- Some device drivers simply assume Hz=1000
- NICs' Offload Engines mess with AQMs! (GSO, TSO, UFO)

Rural Broadband Networks (RNBs)

- Large RTTs, small and fluctuating BWs (120 ms packet transmission time for a 1500 B packet over 100 Kbps link)
- > 500 ms RTTs is not uncommon in RBNs
- Link utilization is paramount => careful setting of AQM thresholds
- Bust absorption is important in RBNs

AQM's Interaction with Scheduling

Benefits

- Flow protection/isolation w/ non-responsive traffic
- Flow-level fairness
- Straightforward AQM config (e.g. picking thresholds per single flow in VQ)

AQM's Interaction with Scheduling

Benefits

- Flow protection/isolation w/ non-responsive traffic
- Flow-level fairness
- Straightforward AQM config (e.g. picking thresholds per single flow in VQ)

(S)FQ_*AQM* Implementation Status

- SFQ_CoDel (ns-2, Linux/iproute)
- SFQ_RED (Linux/iproute)
- SFQ_ARED (TO-DO)
- SFQ_PIE (?)

AQM and ECN

- Use of ECN mandates AQM deployment
- Simplistic ECN implementation in AQMs (simply CE-marking instead of dropping)

AQM and ECN

- Use of ECN mandates AQM deployment
- Simplistic ECN implementation in AQMs (simply CE-marking instead of dropping)

Implementation Flaws and Mis-conceptions

- **RFC 3168:** *CE code-point SHOULD only be set if the router would otherwise have dropped the packet as an indication of congestion.*
- CE-marked packets contribute to delay/queue-size measurements => normally $p_{marking|ecn} > p_{drop|noecn}$ with constant backlog

AQM and ECN – cont.

TO-DO

- Update the code to (somehow) take into account CE-marked packets
- "Baseline" configuration of similar marking / dropping should be documented; this is not a configuration with equal thresholds
- Update the AQM thresholds for ECN traffic (lower)

$p_{marking|ecn} / p_{dropping|noecn}$ (real-life test)

TCP Flows	CoDel	PIE	ARED
4	1.256	1.156	6.621
16	1.356	1.106	3.465
32	1.719	1.591	4.303
64	6.117	6.569	3.873

Traffic

- Bulk TCP transfer as a starting point to verify TCP-based AQM assumptions
  - CoDel uses TCP-based relationship between p_{drop} and throughput
- Realistic HTTP web traffic (ON-OFF dist.)
  - Mostly in Slow-Start
  - TMIX?
- Many others (e.g. Video, Audio, Gaming, etc.)

Ongoing Work

- Common AQM evaluation suite I-D
- ns-2 simulation (and real-life test) code to be published

Q&A