QoS and AQM

AQM BoF IETF 87

It's about constraints

QoS and diffserv are about constraints that give good application behaviour

- minimum latency (interactive)
- minimum jitter (voice)
- maximum throughput (best effort)
- throughput scavenging

Most of these are matters of degree

Any guarantees are statistical

Not all packets are equal (e.g. DNS, SYN etc)

What behaviours make sense?

- Explicit signalling is broken by inconsistent legacy of standards and deployment
  - IETF never gave minimum set of code points
  - Linux pfifo_fast is the deployed base, but does something inconsistent with standards
  - Not this group's problem, but a recommendation needs made somewhere in the IETF
- Deduction is often possible according to application and/or transport behaviour
- Too many classes make the problem's statistics fail to reach significance

fq_codel

- Has implicit flow behaviour deduction
- Deficit round robin with two lists of flows
    - 'old flows' list has one quantum drawn for each pass through the 'new flows' list
    - move a flow to the 'old flows' list when the per-flow CoDel builds queue
    - flows return to 'new flows' when the queue drains
- Has some interesting corner cases
- 'Old flows' are really throughput maximising
- Scavengers tend to flip between old and new

Limitations on deduction

- Statistics and training time
- Lower layer behaviour
- Measurement precision
- Interaction between congestion control, packet scheduling, and queue behaviour

But we ought to try, in the absence of other information

Layers below IP

IEEE 802.11 and 802.1 have MAC-layer QoS

- 802.11e is frequently badly implemented
  - Strict-ish priority
  - Wastes air time (txops) easily
  - Experiment shows prioritised aggregation works better

- Four basic priorities
  - VO (voice)
  - VI (video/interactive)
  - BE (best effort)
  - BK (background)

Layers below IP

- Aggregating MACs group packets together
  - This messes with delay-based congestion control and sojourn-time based AQM unless the measurement is done carefully
  - 802.11n, LTE and DOCSIS all do this
  - There will be more, any MAC where acquiring the medium is expensive benefits
  - Cross-layer interaction between device drivers and IP stack
- AQM working group will need to take this into account

Shallow buffering

- Inverse of bufferbloat, some network paths have extremely small buffers
- SDN makes network capacity much cheaper, but typical switches have very small buffers, so this situation is likely to increase
- Causes linear rise in log drop probability, no real 'knee' in the drop curve
- Such switches can't cope with highly bursty traffic workloads in the absence of AQM
- AQM can help here by smoothing the traffic
- But smooth traffic hitting big buffers is worse

Criteria for success

Can we meet the desired constraints robustly, and define the 'flight envelope' of the algorithm?

Does that flight envelope make operational sense?

Can we say that we can recommend any set of behaviours for deployment on by default?

Can we recommend application behaviours?