Low Extra Delay Background Transport (LEDBAT)

draft-shalunov-ledbat-congestion-00.txt

Stanislav Shalunov ⟨shalunov@bittorrent.com⟩

IETF 74, March 23, 2009, San Francisco, LEDBAT WG

Problem reminder

- TCP fills buffer, maximizing delay
- Interactive apps need low delay
- Let's minimize delay in congestion control

Design goals

- saturate the bottleneck
- keep delay low with no other traffic
- yield to TCP
- add little extra delay
- work today: FIFO, drop-tail, NATs
- work tomorrow: ECN, AQM, DiffServ

Simplest topology

- single bottleneck
- single LEDBAT connection
- no other traffic

Extra delay

- bottleneck queuing delay
- keep low
- but not 0 (or not saturating)

Queuing delay target

- to deploy today, need external delay measurements
- min error at least best-case scheduling error
- no need to go higher

Need to measure delay

- control queuing delay → estimate it
- without delay, would have to go by loss
- loss would need delay spikes

Queuing delay estimate

- delay = propagation + serialization + processing + queuing
- delay = base + queuing
- queuing ≥ 0
- estimator: base = min(delay)
- queuing = current - base

Controller

- queuing > target → slower
- queuing < target → faster
- queuing ≈ target → steady (stability)
- simplest is proportional
- proportional works

Max rampup rate

- fastest increase ↔ queuing = 0
- calibrate this to be 1 MSS/RTT as in TCP to be on the safe side

Halve on loss

- loss is induced by TCP (or brokenness) and not part of standard regime
- for severe congestion, halve on loss

Saverio's alternate

- cwnd = min (cwnd/2, base_rtt * measured used bandwidth)
- need to play with
- should mostly matter in LANs

Yield to TCP

- delay needs to increase to buffer for loss
- queuing = 2*target → LEDBAT rampdown = TCP rampup

One-way delay

- why: reverse-path congestion
- how: add header timestamps

Route changes

- base over last few minutes
- fewer minutes → faster reaction
- more minutes → more stable

Timestamp errors

- error = offset + skew*t
- offset cancels in (current + offset) - (base + offset)
- 10PPM skew = 0.6 milliseconds/minute
- base over last few minutes mostly takes care

Noise filtering

- min of several recent current samples
- several = from 1 to a fraction of RTT

Safety and worst case

- estimator goes bonkers
- queuing = 0 always regardless of speed
- rampup as fast as TCP
- halve on loss
- in other words, fail into TCP

Rough pseudocode

```
on acknowledgement:
        current_delay = acknowledgement.delay
        base_delay = min(base_delay, current_delay)
        queuing_delay = current_delay - base_delay
        off_target = TARGET - queuing_delay
        cwnd += GAIN * off_target / cwnd

on loss:
        cwnd /= 2

        see draft for more precise version
```

QUESTIONS?

ledbat@ietf.org