

# Low Extra Delay Background Transport

Stanislav Shalunov <shalunov@bittorrent.com>  
IETF 73, Minneapolis, LEDBAT WG, Nov 20, 2008

**chair.hat → off()**

# Problem

- TCP fills buffer
- Buffer can be 1 second or more
- Interactive applications fail
- “The Internet doesn’t work”

# How large should the buffer be, anyway?

- Home connection
- 1.5 Mb/s down
- 128 kb/s up
- RTT = 50 ms (ping somewhere.interesting)
- Single bulk uploading TCP connection
- MSS = 1460 B

# Optimal buffer size

A. 50 ms

B. 100 ms

C.  $< 50$  ms

D.  $> 100$  ms

# No right answer

- 50 ms is 1/2 of a packet
- 100 ms is 1 packet
- Large buffer (say, 1 second) leads to standing queue

# Different congestion control

- Measure one-way delay
- Estimate queueing component
- Drive to a target
- Converge and stay there

# Status

- Implemented
- Tested
- Deployed in BitTorrent DNA (7M active users)
- Soon in  $\mu$ Torrent
- Plan to propose as LEDBAT solution candidate
- Didn't yet publish the draft

# One-way delay

- Clock offset doesn't matter
- Clock skew matters to some extent
- RTT doesn't cut it because of reaction to reverse-path congestion
- Not a big deal for loss-based protocols because ACKs are cumulative

# Estimate queueing delay

- current - base
- this is where offset cancels
- works because delays are non-negative
- both measurements may need filtering

# Drive queue to target

- Low delay on saturated pipe doesn't happen by accident
  - Need to keep track of delay and react
- Target in units of time autoscales
- Don't look at differences
  - Scales poorly to high speeds
  - Too fragile

# Converge

- TCP oscillation artifact of having to cause loss
- The closer to target, the slower the change

# Details

- Noise filtering
- Forget very old base delay
  - Route changes
  - Machine suspended
  - Clock skew
- Use smaller packets on slow links
- Controller location: Sender vs receiver
- ...

# Deteriorate to TCP

- Increase no faster than 1 packet/RTT
  - Decreasing faster is OK
- Halve window on loss
  - No more than once per RTT
- Safety net if base delay completely wrong
- Take same input from AQM as TCP

# Yield to TCP

- Queue builds
- Queuing delay increases before loss
- Delay-based congestion control gets the signal before loss-based

# Misc niceties

- Don't need RTT worth of buffer to saturate pipe with single connection
- As others' delay decreases, so does yours
- Suitable for potentially faster increase to tolerate some non-congestive loss (out of scope for LEDBAT)