Enhancing TCP to support Rate-Limited Traffic

CSWS
draft-fairhurst-tcpm-newcww-05.txt

G. Fairhurst, A. Sathiaseelan, R. Secchi, I. Biswas
University of Aberdeen
Scotland, UK

This work has been partially funded by the FP7 RITE Project, 3177700
Rate-limited Traffic

Rate-limited apps are prevalent:

- CBR/VBR motion compensated video
- RTC-Web
- Applications that switch content between streams
- HTTP 1.1 persistent connections
- Google SPDY (persistent TCP connections)
- HTTP Adaptive Streaming (HAS)

TCP was not designed to support rate-limited apps!

TCP reduces to RW and slow starts after idle
TCP increases cwnd during app-limited periods
RFC2861 had a good motivation (protect the network)

However, too conservative for apps to benefit.

Not widely implemented or used.

Propose to obsolete RFC 2861, and define something else.
Differentiate between Validated & Non-Validated Phases

Validated: Standard behaviour

Non-Validated: Updated behaviour
  - ssthresh adjusted
  - Different rate reduction for loss \((D-R)/2\)
  - cwnd does not increase
  - cwnd decreases after NVP
Varying pipeACK
With new-cwv, the cwnd does not grow beyond 2*pipeACK
Varying pipeACK (around $\frac{1}{2}$ cwnd)
new-cwv behaviour reduces cwnd after 5 minutes by $\frac{1}{2}$
Varying pipeACK (around ½ cwnd) 
new-cwv behaviour tracks pipe
Radical behaviour is restart from one packet

Standard TCP recovery mechanism

Alternatively we can estimate successfully transmitted packets \((D-R)\) that provides an indication of path capacity.

reset the cwnd after recovery by \((D-R)/2\)

* \(D\) is the flight size \(R\) the number of packets detected as lost
Rate-limited traffic sources (512 kb/s)

Idle (no data sent) or app-limited (reduce to 12kb/s)
App benefit

5 sec Idle period

5 sec app-limited period

ew-cwv promptly resumes without reducing cwnd
app benefits
Pathology: Path capacity while idle

200 ms path RTT, BDP router buffer, 100 Mbps capacity, app rate 512kbps, 5 sec idle period,
Capacity changes to 2 Mbps, Flow monitor duration 10RTT

new-cwv flows only ~3% higher than TCP fair share during heavy congestion (from 16 flows)
Average receive rate of all new-cwv flows <= TCP Fair share (less than 0.1% difference).
Pathology: Path capacity change while idle

200 ms path RTT, BDP router buffer, 100 Mbps capacity, app rate 512kbps with a 5 sec idle period,
Capacity changes to 2 Mbps, Flow monitor duration 10RTT

new-cwv quickly reduces cwnd after first RTT

Reduced drop rate at bottleneck router compared to padding
We think this is a problem we should address.
Capacity sharing with rate-limited apps is important
  • Benefits the rate-limited apps
  • Protects other apps from collateral damage

Outstanding issues:
  • Tail loss can also be an issue for bursty apps
  • Need to implement the method!
  • Pacing (at least coarse-pacing) may help?

[Link to more information](http://trac.tools.ietf.org/group/irtf/trac/wiki/ICCRG_newcwv)