Enhancing TCP to support Rate-Limited Traffic

ICCRG/TCPM

draft-fairhurst-tcpm-newcwv-05.txt

G. Fairhurst, A. Sathiaseelan

University of Aberdeen

Scotland, UK

http://trac.tools.ietf.org/group/irtf/trac/wiki/ICCRG_newcwv

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

Congestion Window Validation

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.

IETF diff between -03 and -04

ICCRG feedback

Used term rate-limited in all places.

Added justification and minor changes suggested on the list.

Added text to tie-in with more accurate ECN marking.

Added ref to Hug01 (but did not specify pacing)

IETF diff between -04 and -05

Fixed issue for infrequent large bursts:

- Non-Validated Period (NVP)
- Introduced pipeACK, to replace FlightSize
- This reflects actual acknowledged usage

Changed NVP entry to pipeACK < ½*cwnd

Changed NVP exit conditions:

- pipeACK leaves NVP after pipe was acknowledged.
- Removed need for hysteresis.

Overview : new-cwv

new-cwv:

record pipeACK

if pipeACK < ½*cwnd

enter non-validated period (NVP)

/* freeze cwnd */

else

behave as standard TCP

new-cwv exits NVP:

after 5 mins

if cwnd > ½*pipeACK

if the RTO expires, resets cwnd to RW

if sender receives congestion feedback ,

sender must rapidly reduce cwnd

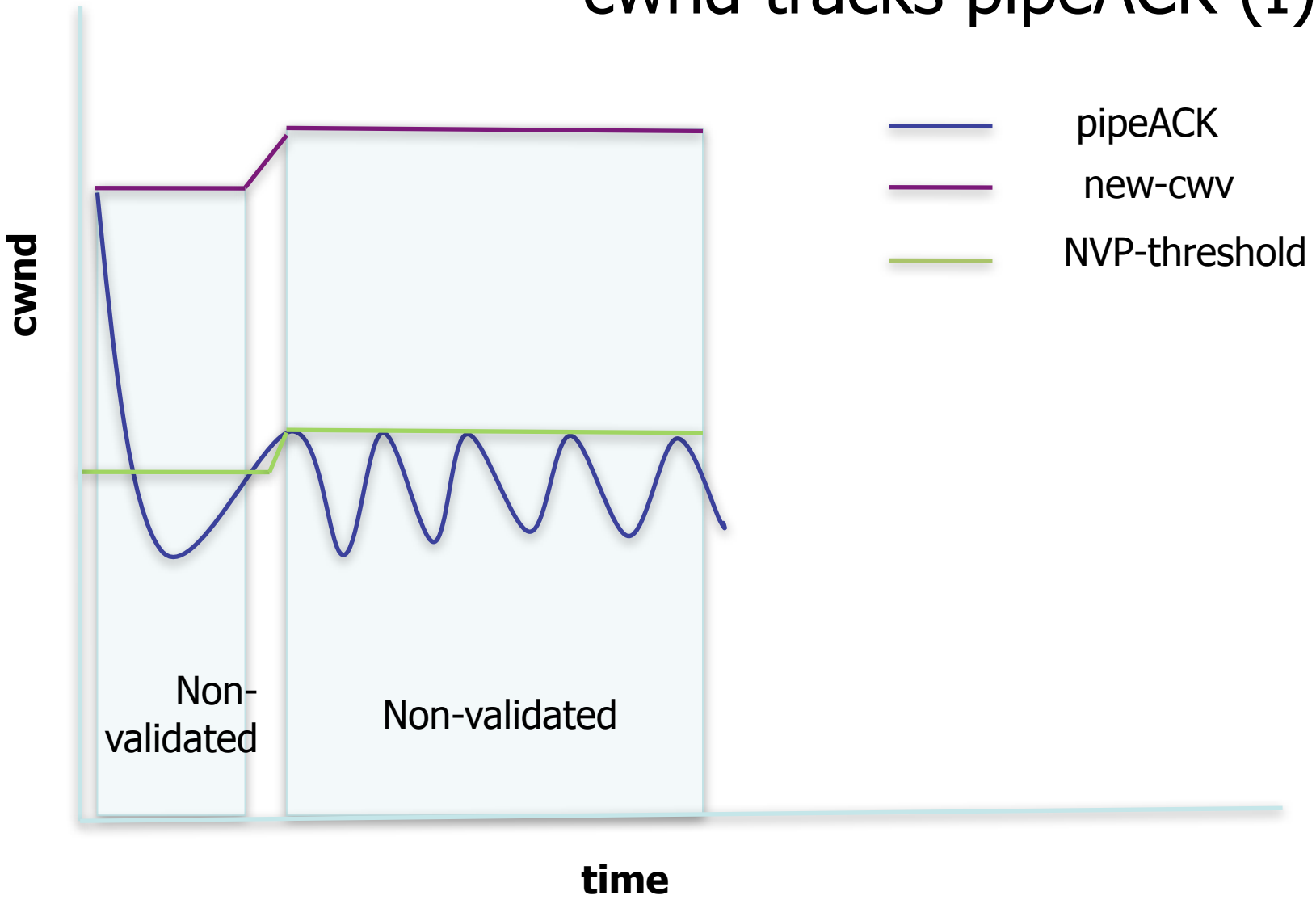
At the end of NVP:

ssthresh = max(ssthresh, 3*cwnd/4) /* reset path characteristic */

/* avoid excessive overshoot, as in RFC 2861 */

cwnd = max(1/2*cwnd, IW)

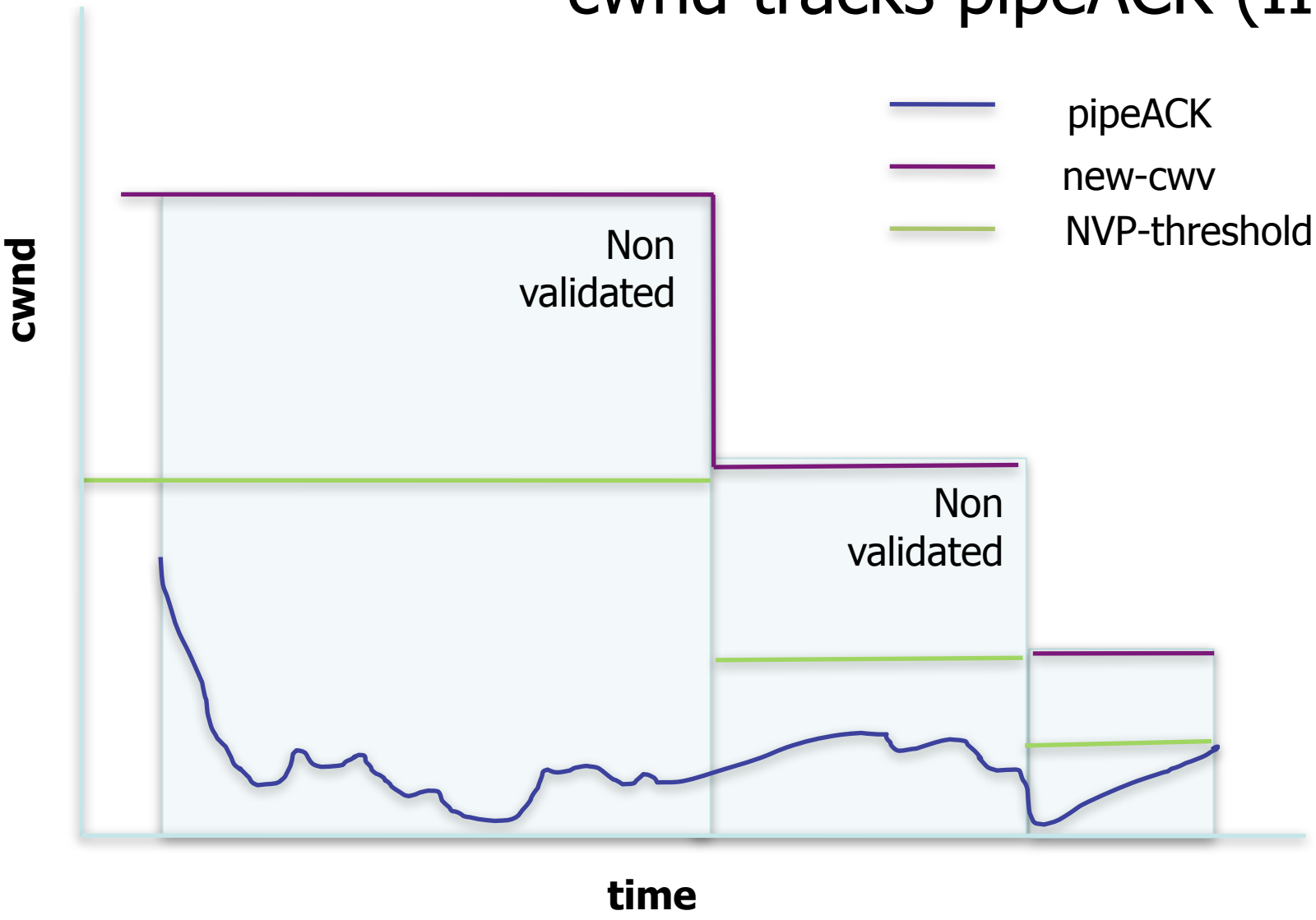
cwnd tracks pipeACK (I)



Varying pipeACK

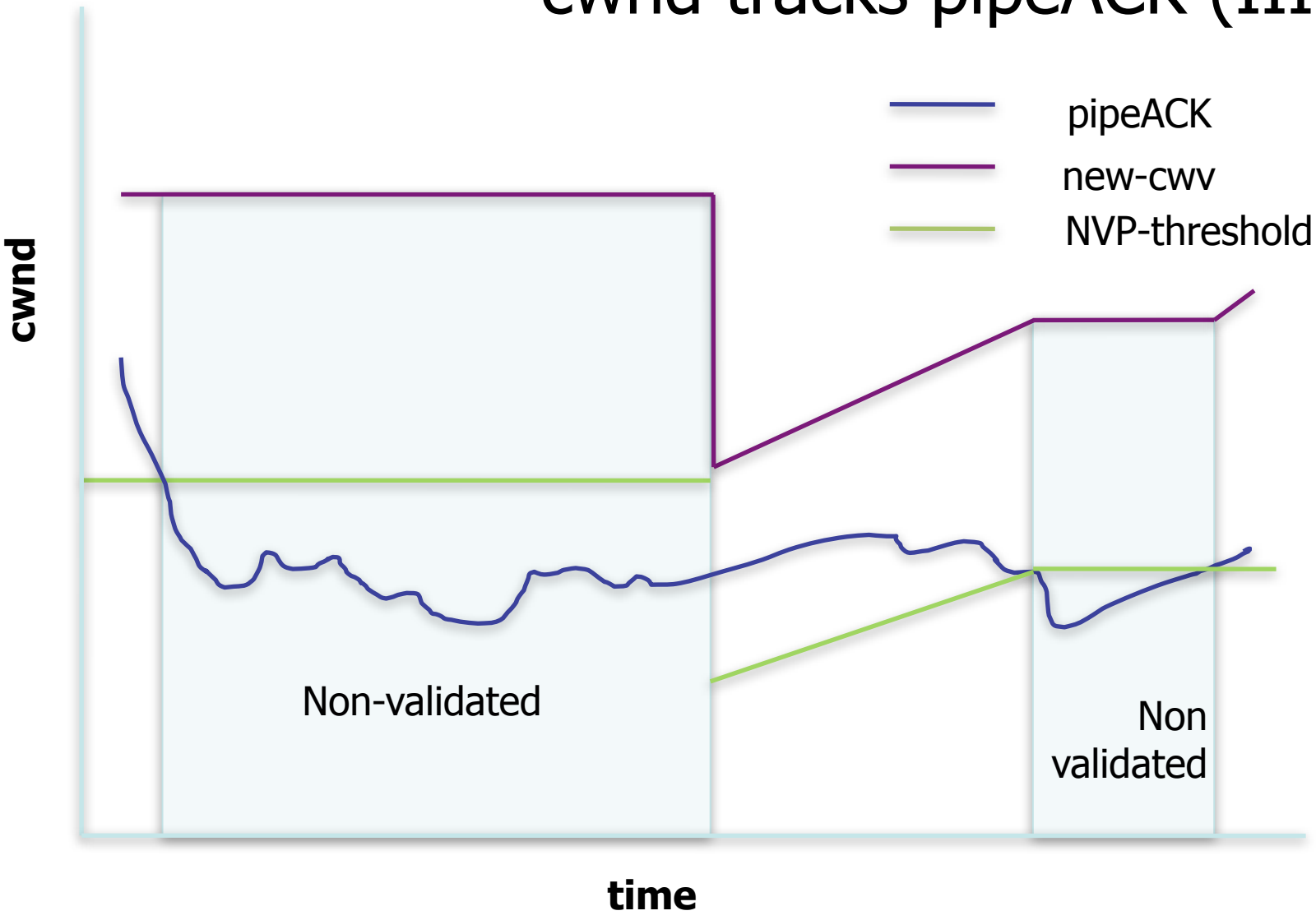
With new-cwv, the cwnd does not grow beyond 2*pipeACK

cwnd tracks pipeACK (II)



Varying pipeACK (around ½ cwnd)
new-cwv behaviour reduces cwnd after 5 minutes by ½

cwnd tracks pipeACK (III)



Varying pipeACK (around ½ cwnd)
new-cwv behaviour tracks pipe

Why is NVP 5 mins?

There is no “magic number”

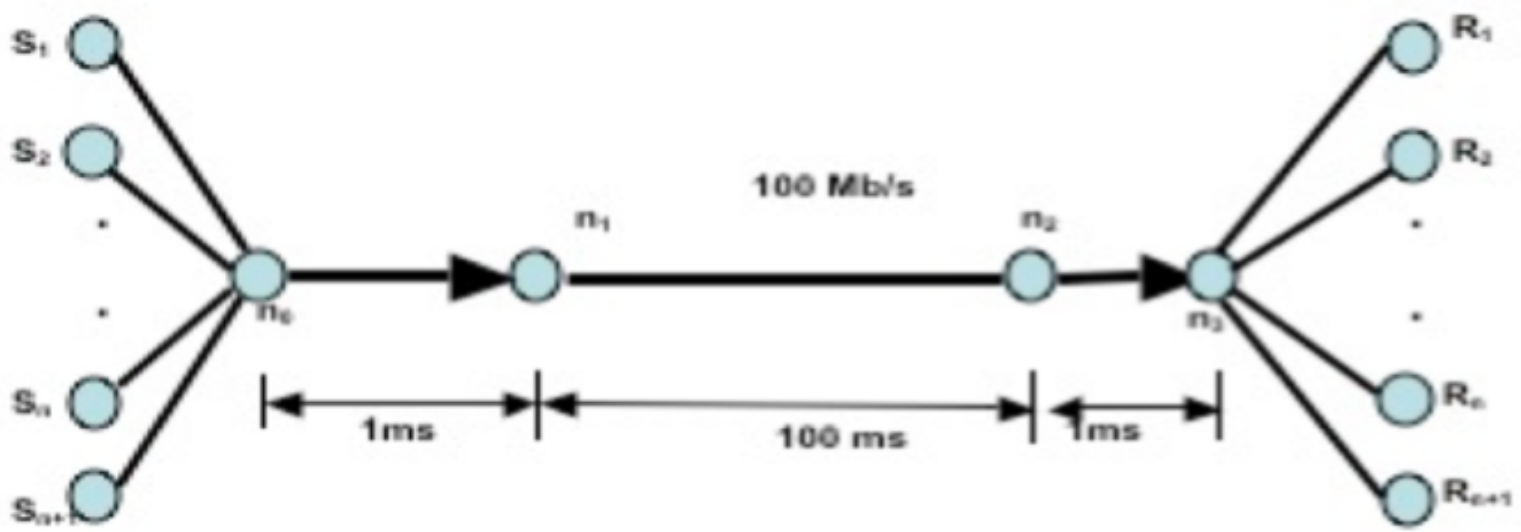
Characteristic idle periods ~ few secs to few minutes

Network paths are relatively stable for several minutes

TCP default user timeout of 5 minutes - how long transmitted data may be unacknowledged before closed.

Expected to be sufficient for common apps

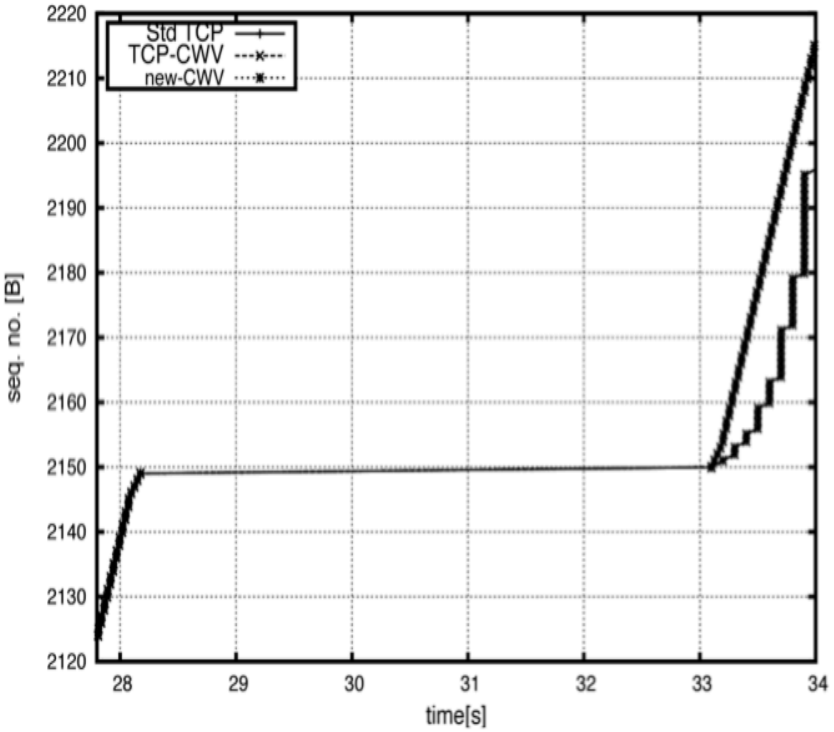
Analysing Impact - Simulation Topology



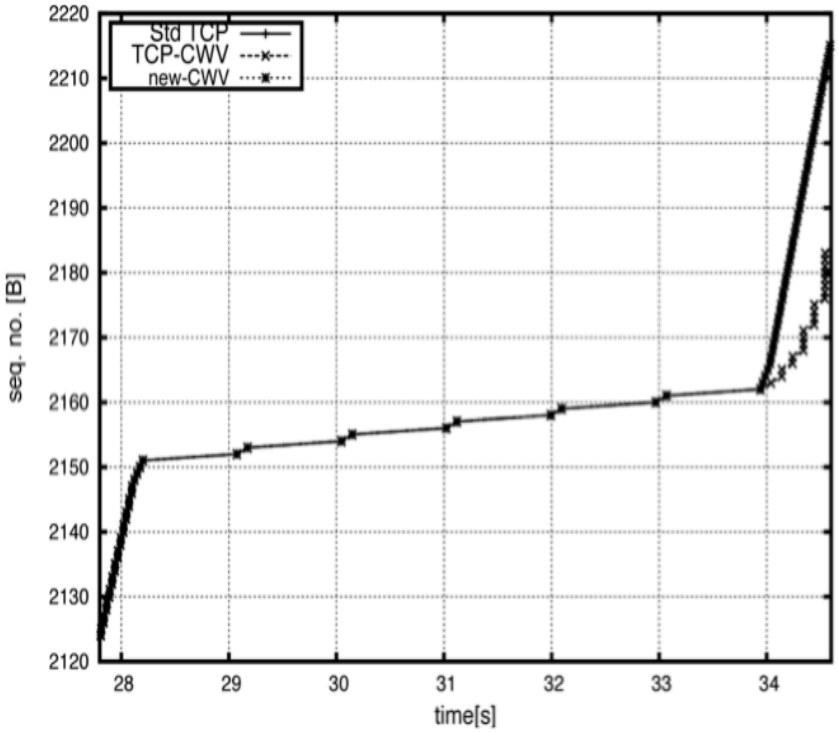
Rate-limited traffic sources (512 kb/s)

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

App benefit



5 sec Idle period

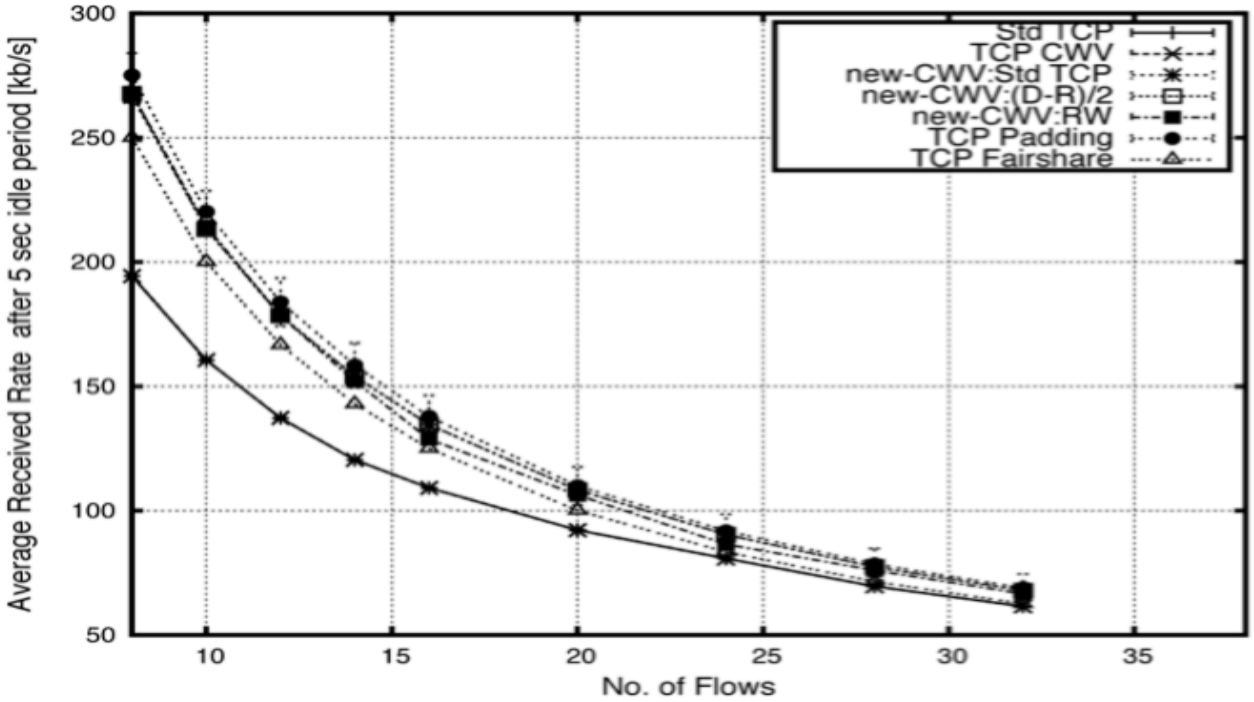


5 sec app-limited period

new-cwv promptly resumes without reducing cwnd
app benefits

Pathology: Path capacity while idle

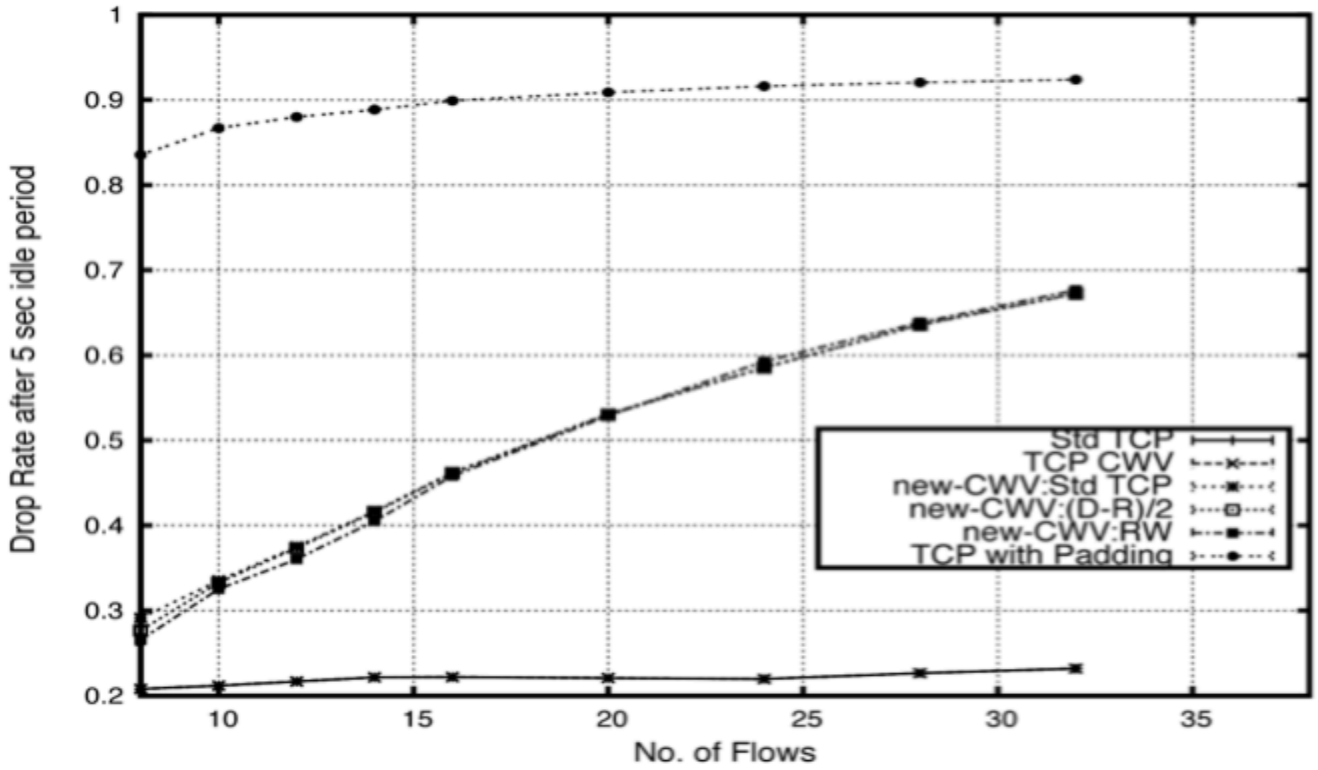
200 ms path RTT, BDP router buffer, 100 Mbps capacity, app rate 512kbps, 5 sec idle period,
Capacity changes to 2 Mbps , Flow monitor duration 10RTT



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

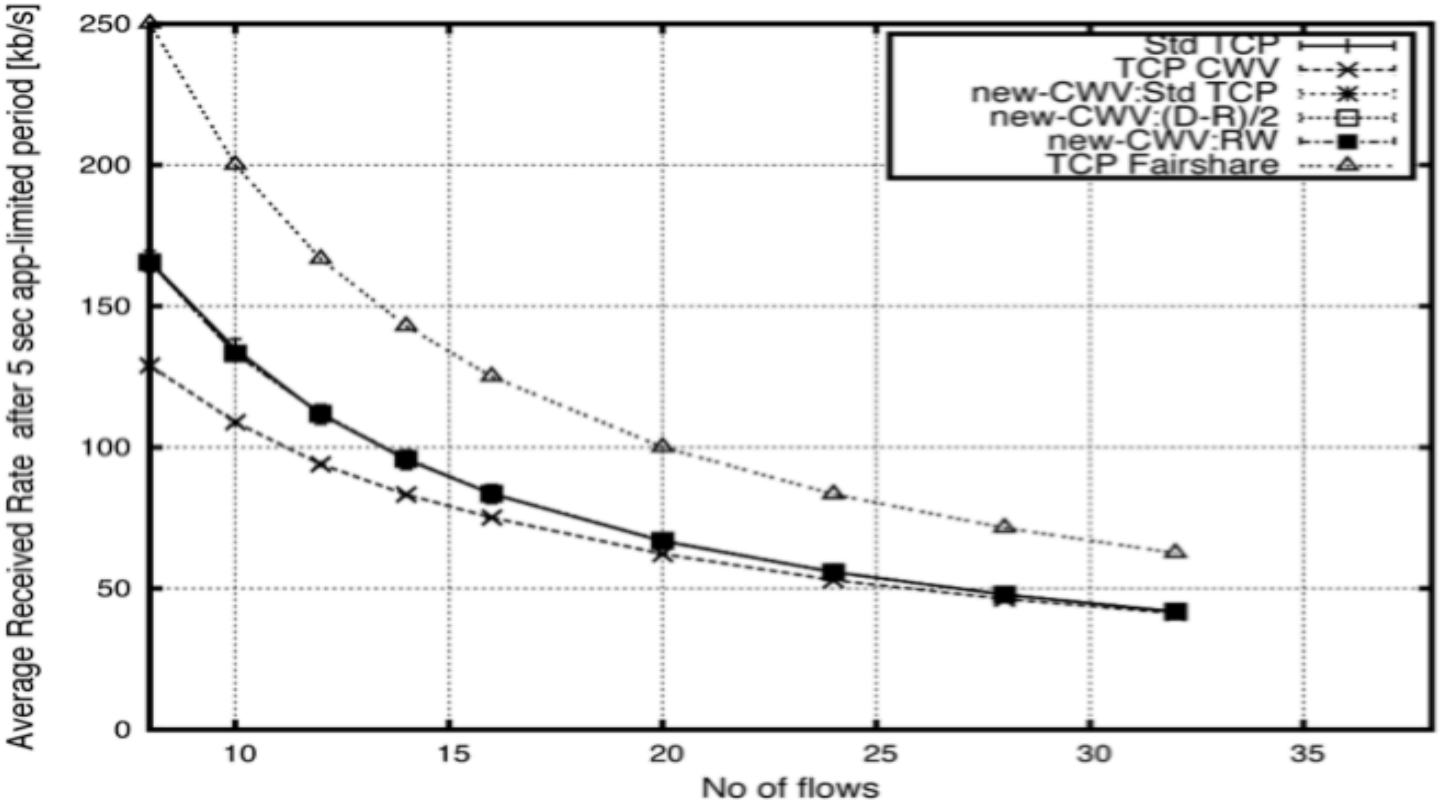


new-cwv quickly reduces cwnd after first RTT

Reduced drop rate at bottleneck router compared to padding

Pathology: Path capacity change while app-limited

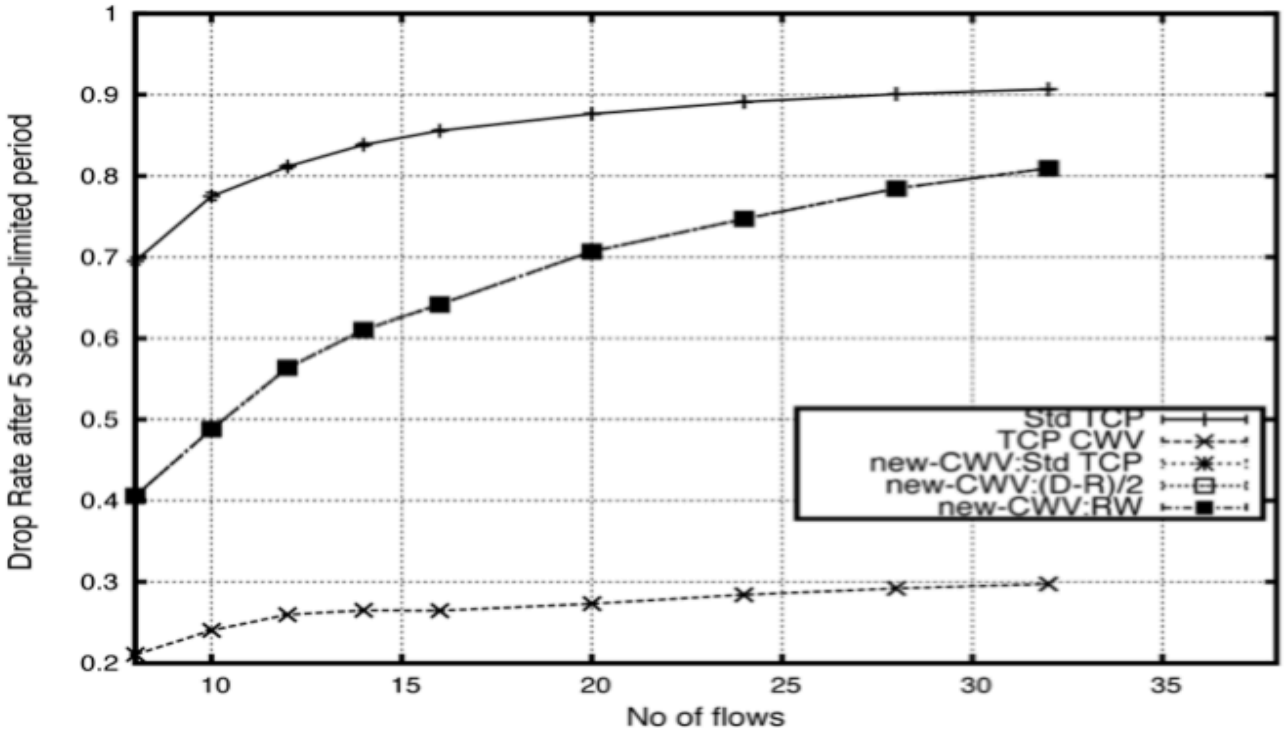
200 ms path RTT, BDP router buffer, 100 Mbps capacity, app rate 512Kbps with a 5 sec app-limited period (12kb/s),
Capacity changes to 2 Mbps, Flow monitor duration 10RTT



Higher average receive rate and better TCP fair share for new-cwv

Pathology: Path capacity while app-limited

200 ms path RTT, BDP router buffer, 100 Mbps capacity, app rate 512Kbps with 5 sec app-limited period (12kb/s),
Capacity changes to 2 Mbps, Flow monitor duration 10RTT



Induces fewer packet drops than Standard TCP

Summary

Many simulations of new-cwv

- Hard to draw universal conclusions
- We explored corner cases
  - Benefits rate-limited applications
  - Appropriate response if congestion experienced

New-cwv responds faster to network and app

- Recommend to reduce cwnd to (D-R)/2
- Reduces overshoot after a path/capacity change

Updates planned for -06

Require reset of pipeACK after congestion

Added comment on effect of congestion after a short burst (M. Allman)

Correction of minor typos to improve consistency

Next Steps

Outstanding issues:

- IW has similarities, but is different (see draft)
- Laminar is different (but this proposal for STD TCP)
- Tail loss can also be an issue for bursty apps

We are looking for more reviews of the draft...

Is anyone else interesting in implementing this?

We would like to see this adopted as a TCPM work item!

http://trac.tools.ietf.org/group/irtf/trac/wiki/ICCRG_newcwv