Brief Overview of H-TCP

Doug Leith
Hamilton Institute
Ireland

Three elements

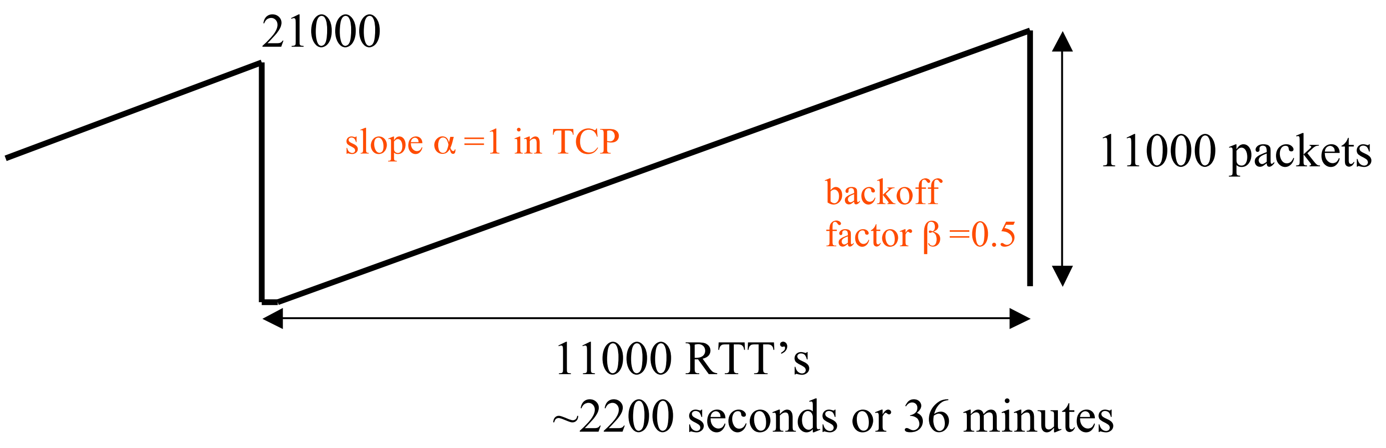
1. Improve scaling of congestion epoch duration with bandwidth-delay product (BDP). Aim is to maintain general purpose nature of TCP - scales from dial-up modems to multi-gigabit paths. **Primary objective**.
2. Mitigate RTT unfairness
3. Better decouple throughput efficiency from network buffer provisioning

Guiding Rationale

1. Seek to make smallest possible changes to current TCP algorithm. Also preserve as much of well understood standard TCP behaviour as possible. Clean slate design a different problem.
2. Pix ‘n’ Mix. Modular solution.

Congestion Epoch Duration

Long distance gigabit link …



- TCP becomes sluggish, and requires v.low drop rate to achieve reasonable throughput.

Simply making the increase parameter α larger is inadmissable – on low-speed networks we require backward compatibility with current sources.

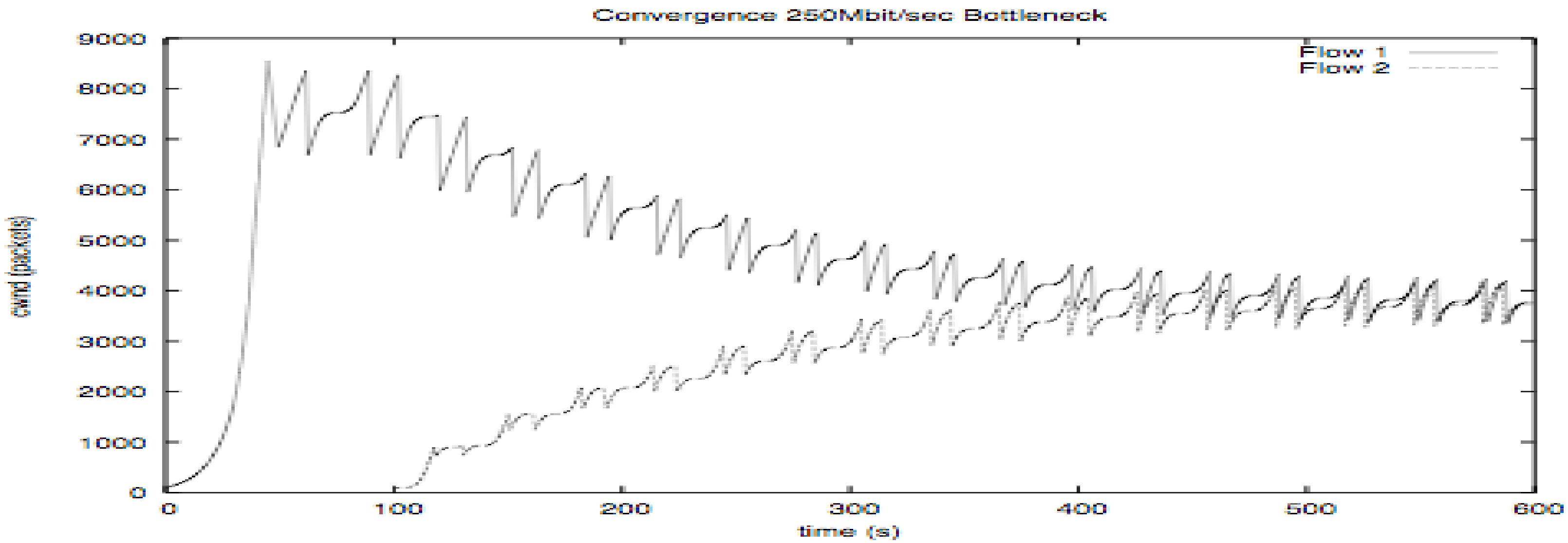
Large α in high-speed regimes, α=1 in low-speed regimes suggests some sort of mode switch.

Congestion Epoch Duration

Care is needed in how increase rate α is adjusted.

For example, say we make α vary with flow cwnd. Natural, as large cwnd implies a large BDP.

But flows with large cwnd now are more aggressive than flows with low cwnd. Say a new flow starts up, then it is at a disadvantage to the incumbent flows and can take a long time to win its fair share of the network bandwidth. E.g. Cubic



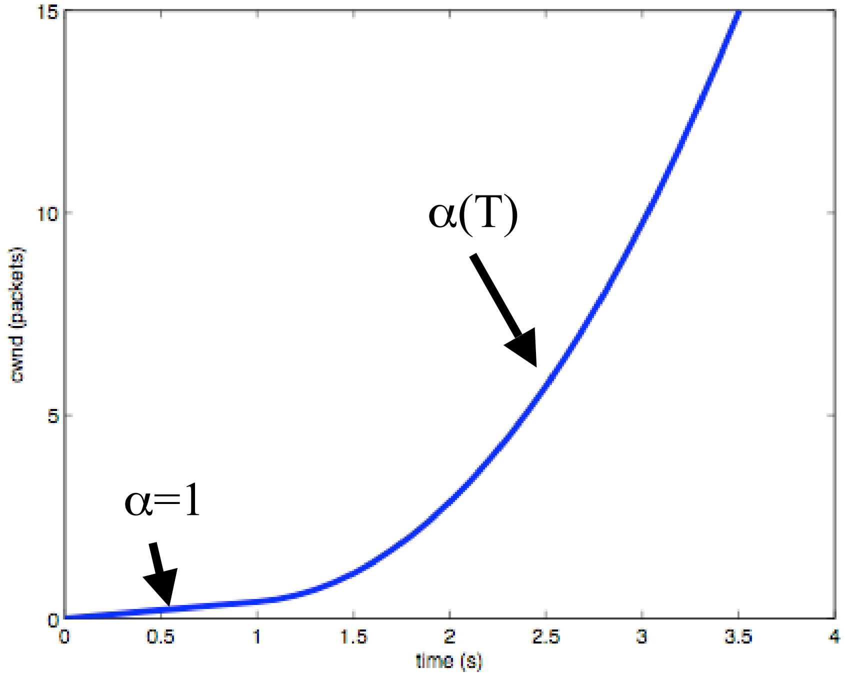
Does it matter ? Yes !

Congestion Epoch Duration: H-TCP

In H-TCP, increase rate α is a function of the elapsed time since last backoff.

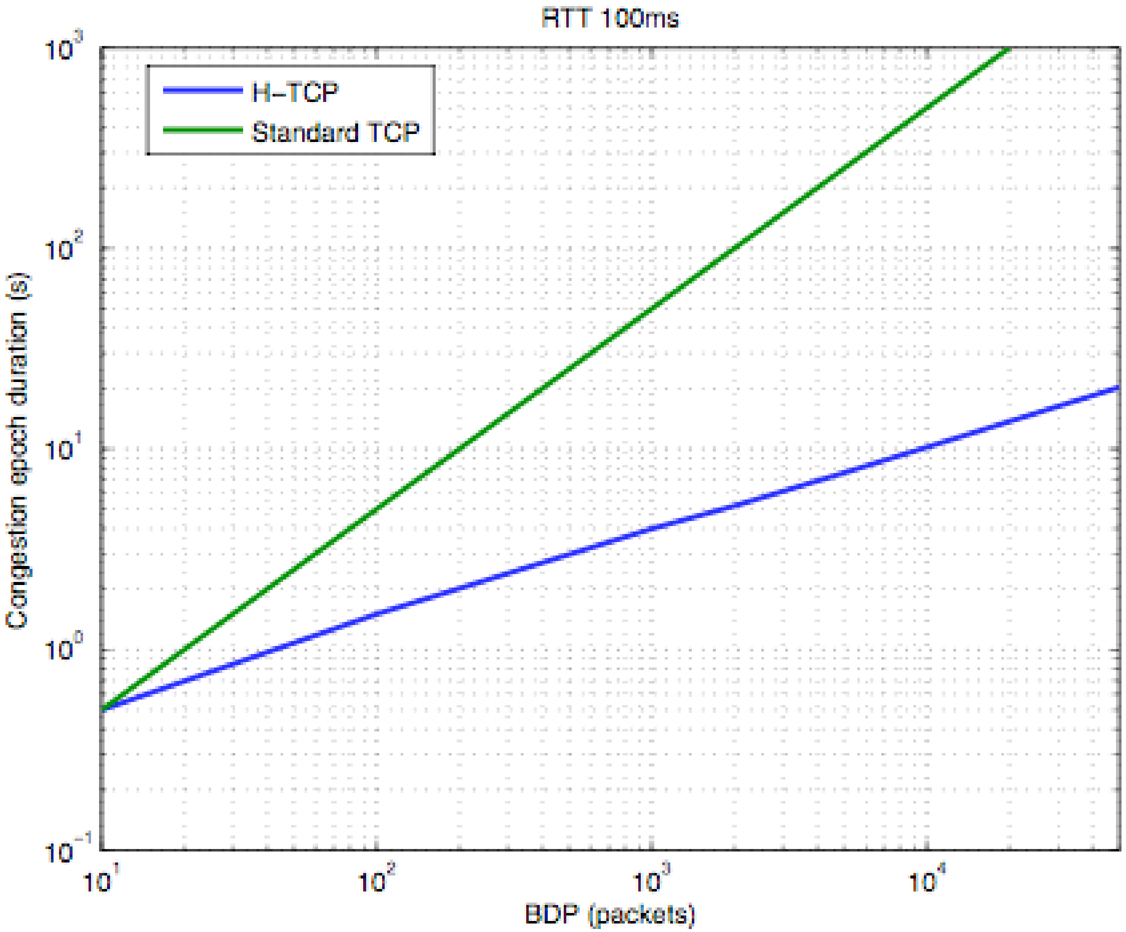
-avoids giving an advantage to established flows with large cwnd - all flows compete for bandwidth on an equal footing

-guarantee backware compatibility on slow links by using standard value for α=1 initially.

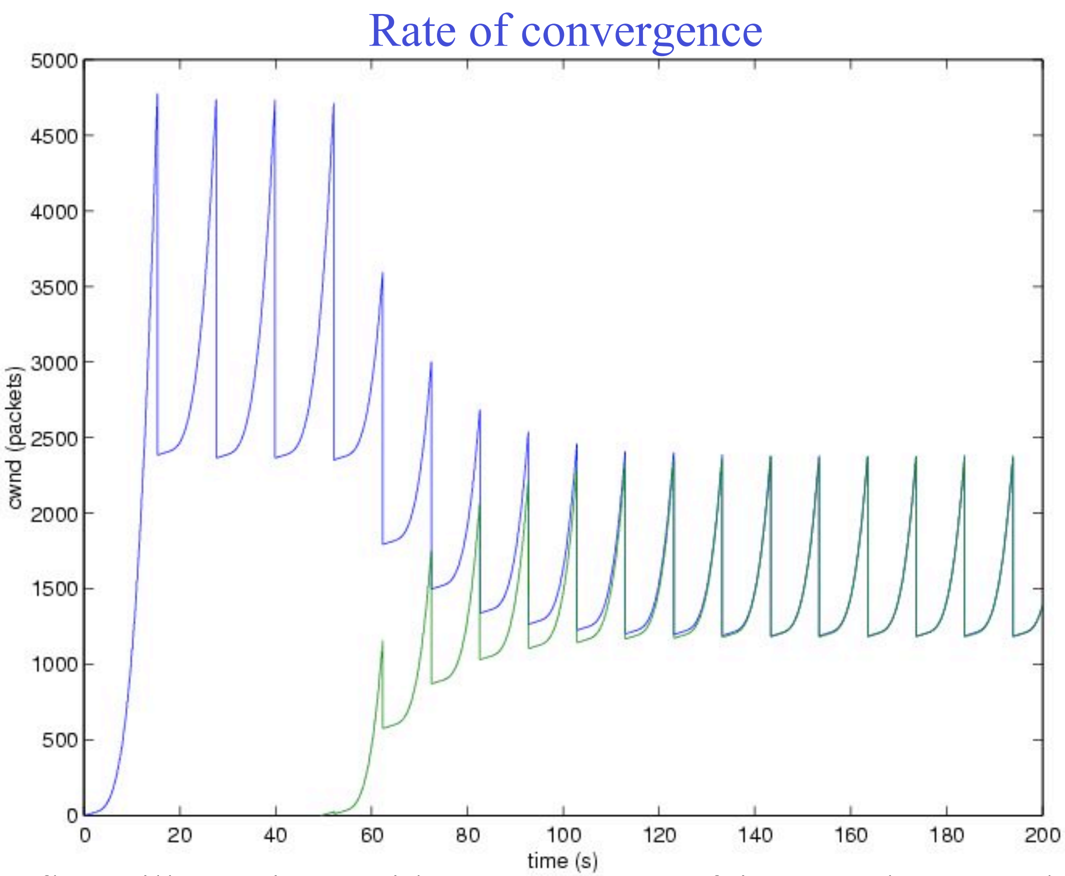


For details, see *H-TCP: TCP Congestion Control for High Bandwidth-Delay Product Paths*, Internet Draft draft-leith-tcp-htcp-03.

Congestion Epoch Duration: H-TCP



Congestion Epoch Duration: H-TCP



Example of two H-TCP flows illustrating rapid convergence to fairness - the second flow experiences a drop early in slow-start focusing attention on the responsiveness of the congestion avoidance algorithm. (500Mb link, 100ms delay, queue 500 packets)

Key Properties

Fairness

Preserves key fairness properties of standard TCP
- flows with same RTT achieve same average throughput
- flows with different RTT's unfair (roughly as $1/RTT^2$).

Note. Scaling α with flow RTT can mitigate RTT unfairness. Optional.

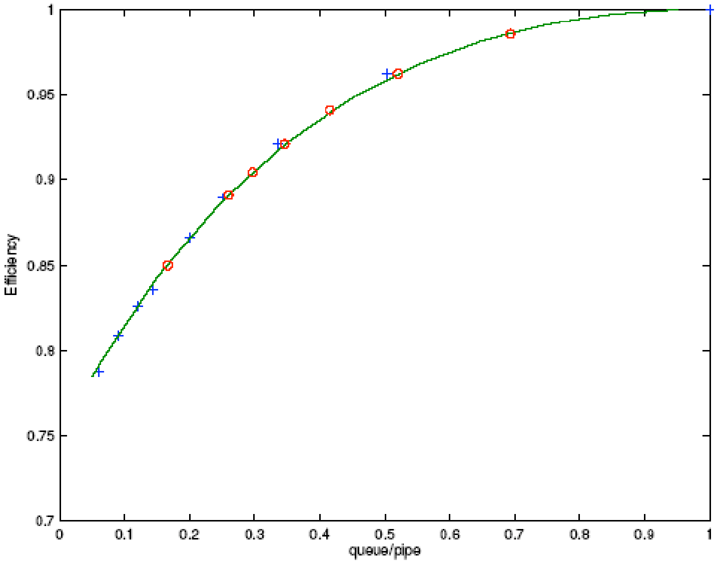
Pros/cons discussed in detail in IETF discussion doc at www.hamilton.ie/net/htcp/rtt.pdf

Efficiency

Preserves standard TCP relationship between buffering and link utilisation.

Note. Adapting cwnd backoff factor can largely decouple link util from buffer Size. Optional.

Leith,D.J., Shorten,R.N., 2006, *On queue provisioning, network efficiency and the Transmission Control Protocol*. IEEE Trans on Networking, to appear

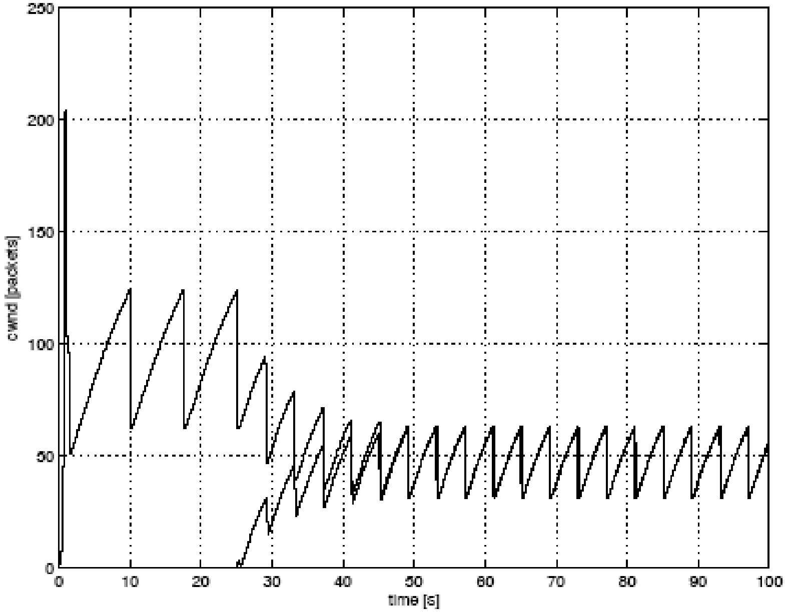


Key Properties

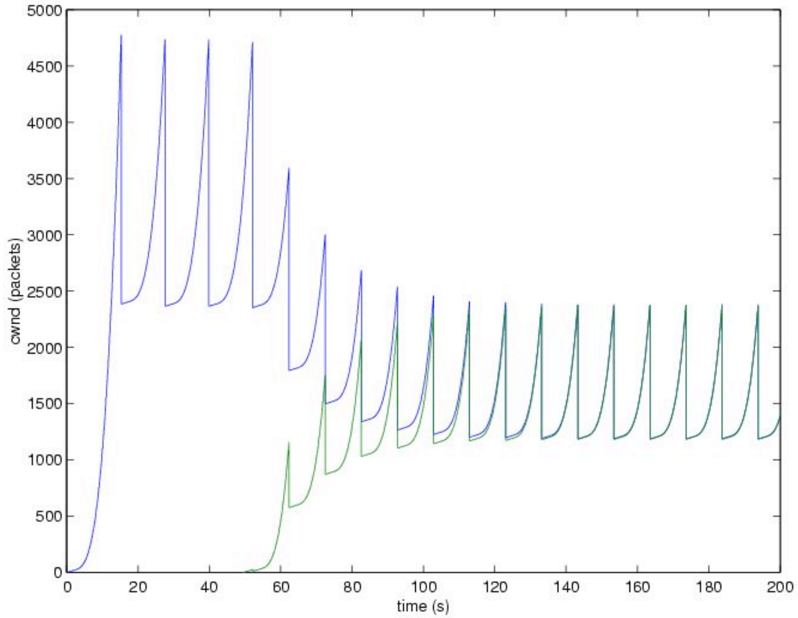
Responsiveness

Preserves fundamental convergence properties of standard TCP[1]

Standard TCP



H-TCP



Congestion epoch duration scaling with BDP is, of course, v. different for standard TCP and H-TCP

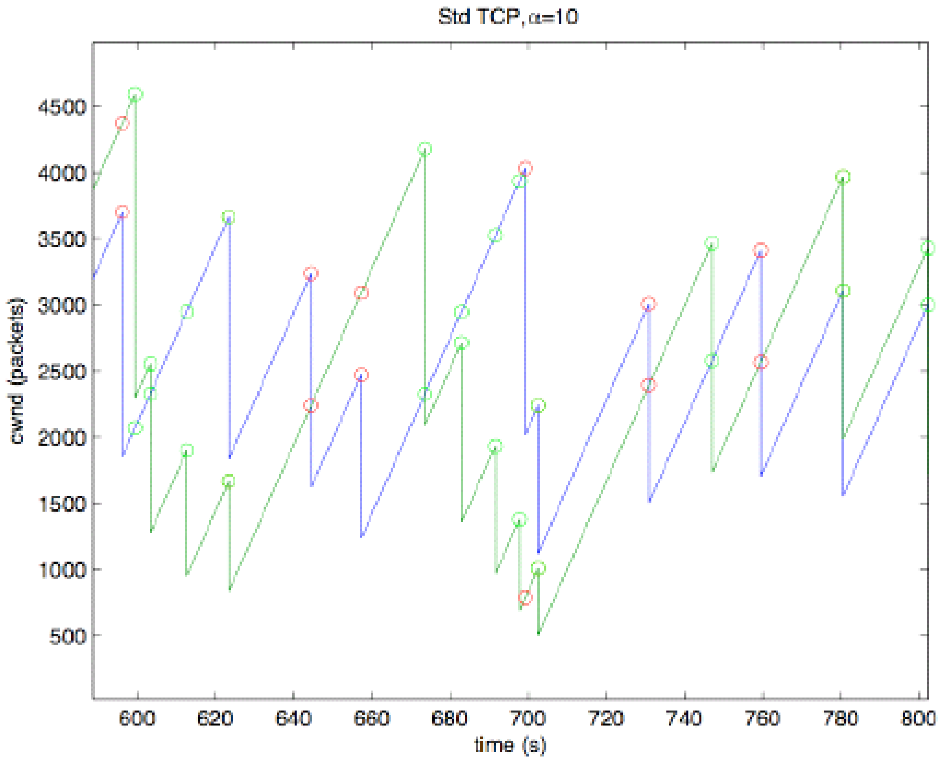
[1]Shorten,R.N., King, C.,Leith,D.J., 2007, *Modelling TCP congestion control dynamics in drop-tail environments*. Automatica, to appear. Also *On the ergodicity of AIMD networks*. Proc. American Control Conference.

Key Properties

Impact of unsynchronised drops

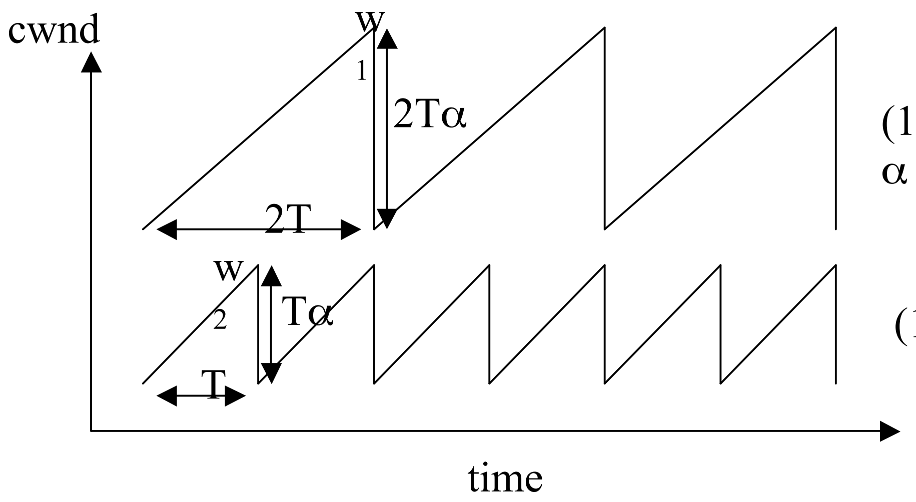
Increased unfairness between flows experiencing different synchronisation rates

… what is meant by *synchronisation rate* ?



Key Properties

Impact of unsynchronised drops

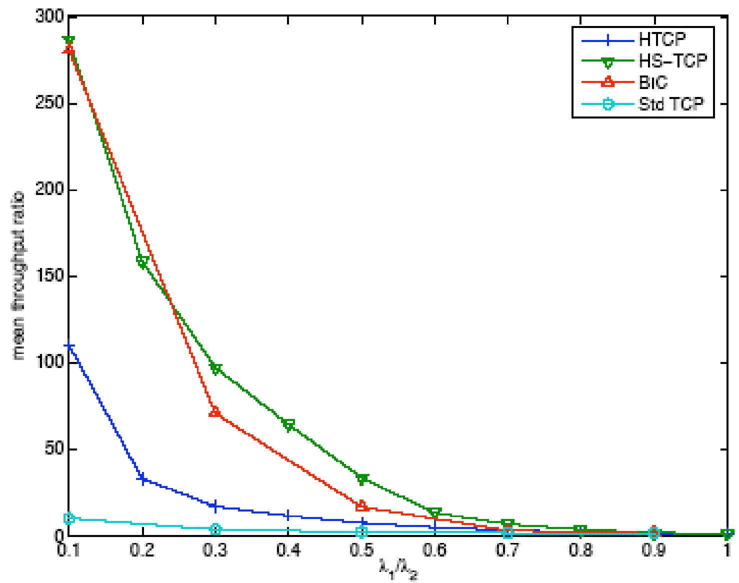


$(1-\beta)w_1 = 2T\alpha$

$(1-\beta)w_2 = T\alpha$

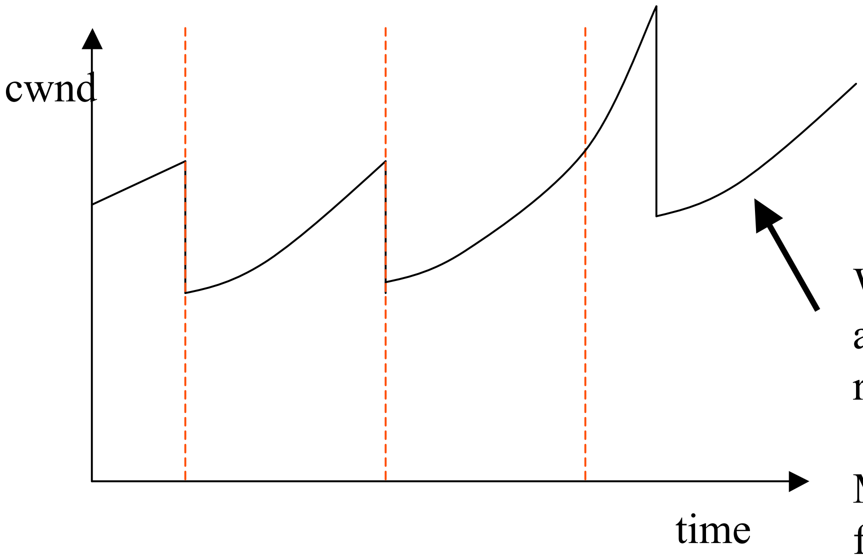
In standard TCP, unfairness is linear in synchronisation rate. More aggressive increase rates of new loss-based algorithms yield greater unfairness.

Shorten, R.N., Leith, D.J., 2006, *Impact of Drop Synchronisation on TCP Fairness in High Bandwidth-Delay Product Networks*. Proc. PFLDnet, Nara, Japan.



Key Properties

Impact of unsynchronised drops



When "miss" a drop, an adaptive increase protocol can run away.

May lead to increased fluctuations in cwnd value.

An issue for loss-based high-speed algorithms. All have aggressive increase functions of one form or another.

Summary

•H-TCP algorithm makes a simple change to cwnd increase function.

-Key differentiating feature is that aggressiveness is a function only of elapsed time since last backoff.

•Preserves many of the key properties of standard TCP. Fairness, responsiveness, relationship to buffering.

• Options to improve RTT unfairness and decouple throughput from buffering.

• Increases sensitivity to differences in synchronisation rate. Common feature of loss-based high-speed algorithms. View of community on this unclear.

•At this stage, H-TCP has undergone quite extensive experimental testing.

-initial SLAC tests Sept/Oct 2003. Li & Cottrell, PFLDnet 2003
-follow up UCL/SLAC tests 2004. Li & Cottrell, PFLDnet 2004
-Hamilton Institute tests, Spring 2005. IEEE ToN 2007.
- Caltech tests (2005 ?)
-North Carolina tests, Feb 2006. PFLDnet 2006
-plus a variety of simulation studies

-Available in standard Linux distributions, IETF draft-leith-tcp-htcp-03