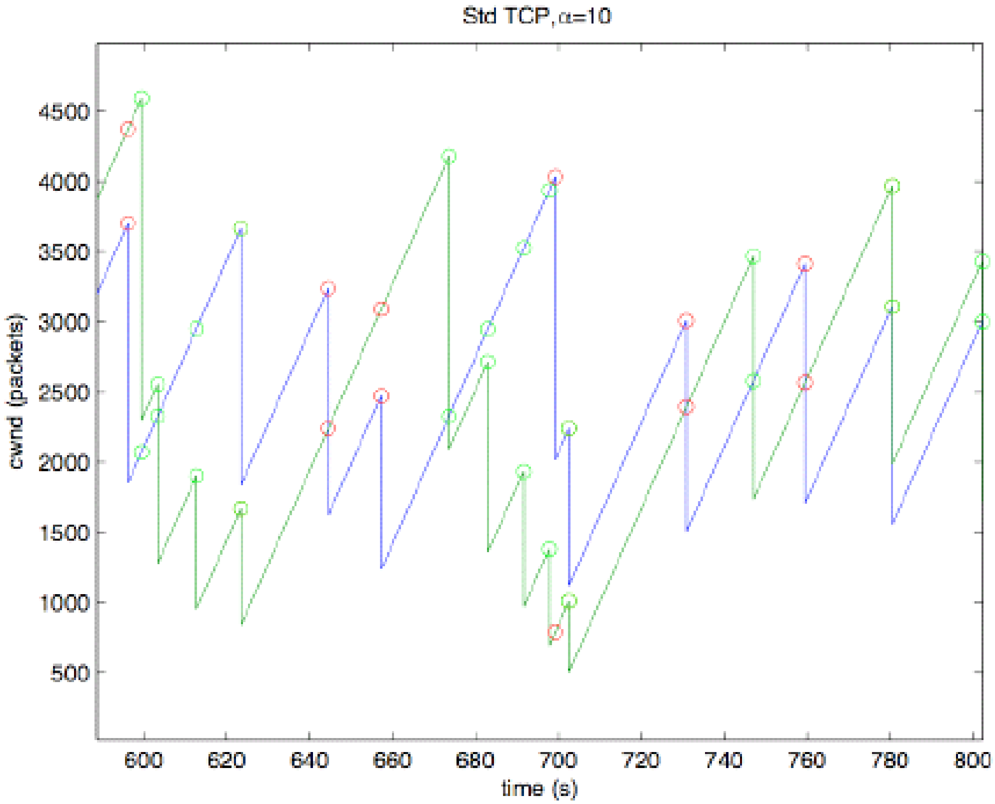
Impact of Drop Synchronisation on TCP Fairness in High Bandwidth-Delay Product Networks

Doug Leith
Hamilton Institute, Ireland

Thanks: Robert Shorten

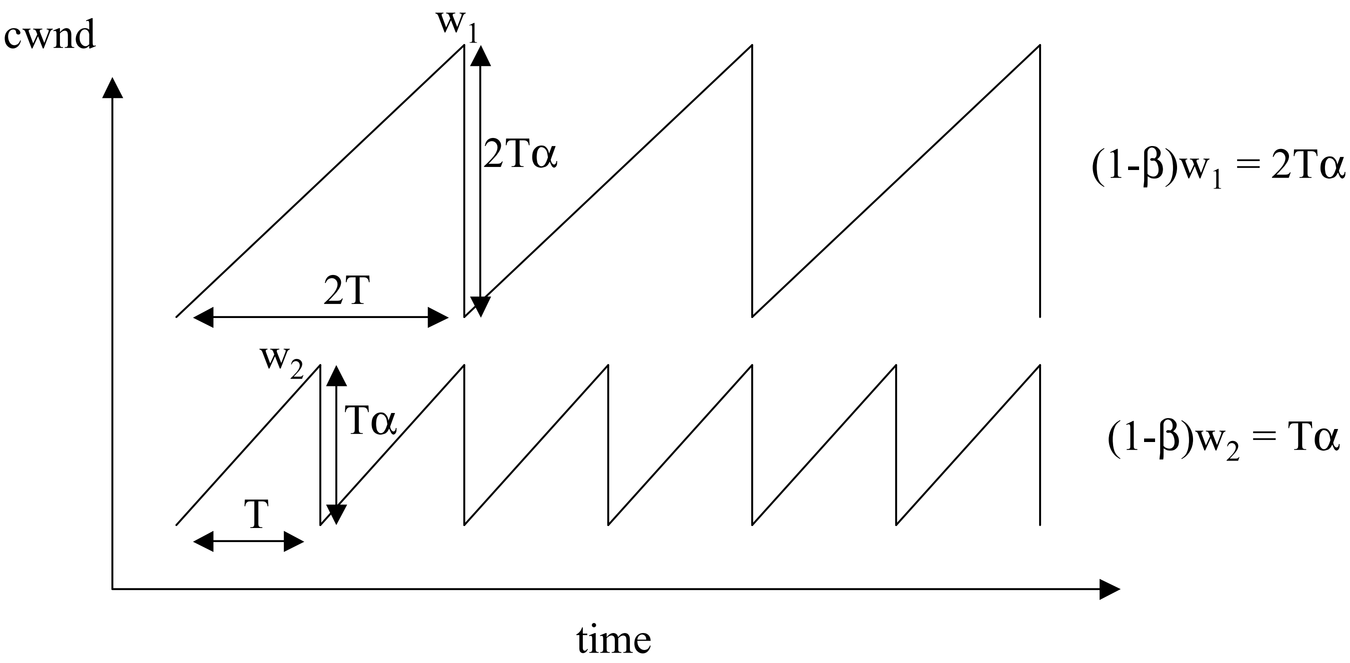
Introduction

Unsynchronised TCP operation is well known. Unsynchronised → "flows do not backoff at every network congestion event".



What do we know about impact on TCP fairness ? Do high-speed protocols exhibit qualitatively different behaviour from standard TCP ?

Long-Term Unfairness in Standard TCP - Periodic example



Let λ be the synchronisation factor, i.e. proportion of network congestion events at which a flow backs off. Then,

$$w = \frac{\alpha}{(1-\beta)}\frac{T}{\lambda}$$

So unfairness between flows scales (inversely) linearly with λ:
e.g ×2 difference in λ ⇒ ×2 difference in *w*.

$$\frac{w_1}{w_2} = \frac{\lambda_2}{\lambda_1}$$

Long-Term Unfairness in Standard TCP - Stochastic case

Same formula holds more generally.

AIMD:

$$w_i(k+1) = \beta_i(k) w_i(k) + \alpha T(k)$$

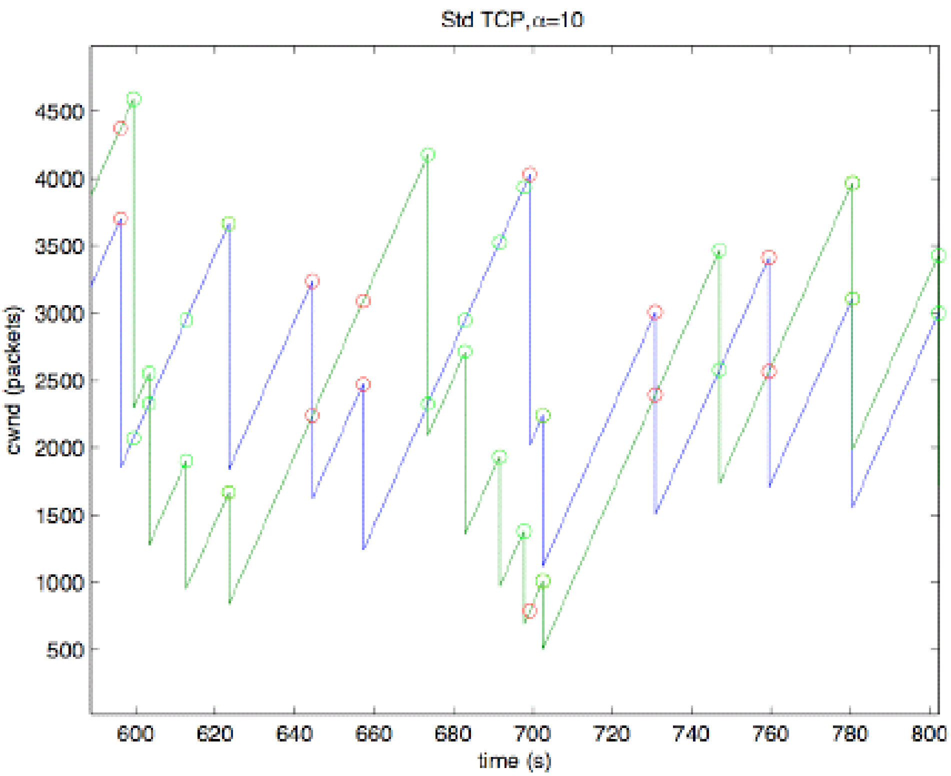
with β(k)=0.5 if backoff at k'th congestion event, otherwise β(k)=1.

Taking averages,

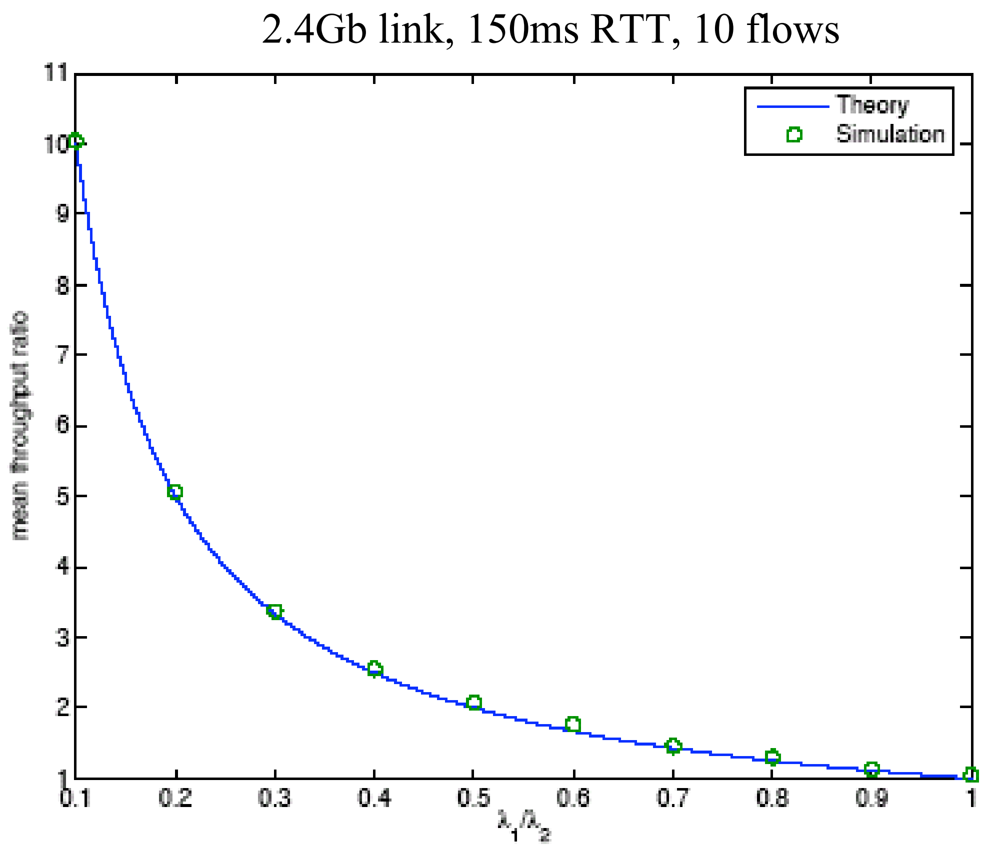
$$\bar{w}_i(k+1) = \bar{\beta}_i \bar{w}_i(k) + \alpha \bar{T}(k)$$

So in steady-state,

$$\bar{w} = \frac{\alpha \bar{T}}{(1-\bar{\beta})} = \frac{\alpha}{(1-\beta)} \frac{\bar{T}}{\lambda} \quad \Rightarrow \quad \boxed{\frac{\bar{w}_1}{\bar{w}_2} = \frac{\lambda_2}{\lambda_1}}$$



Long-Term Unfairness in Standard TCP

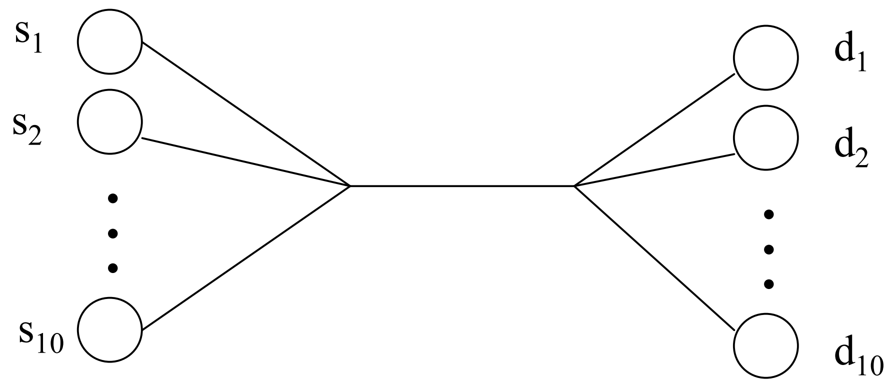


Fairness formula vs simulation

$$\frac{\bar{w}_1}{\bar{w}_2} = \frac{\lambda_2}{\lambda_1}$$

Long-Term Unfairness - NewTCP

Setup:

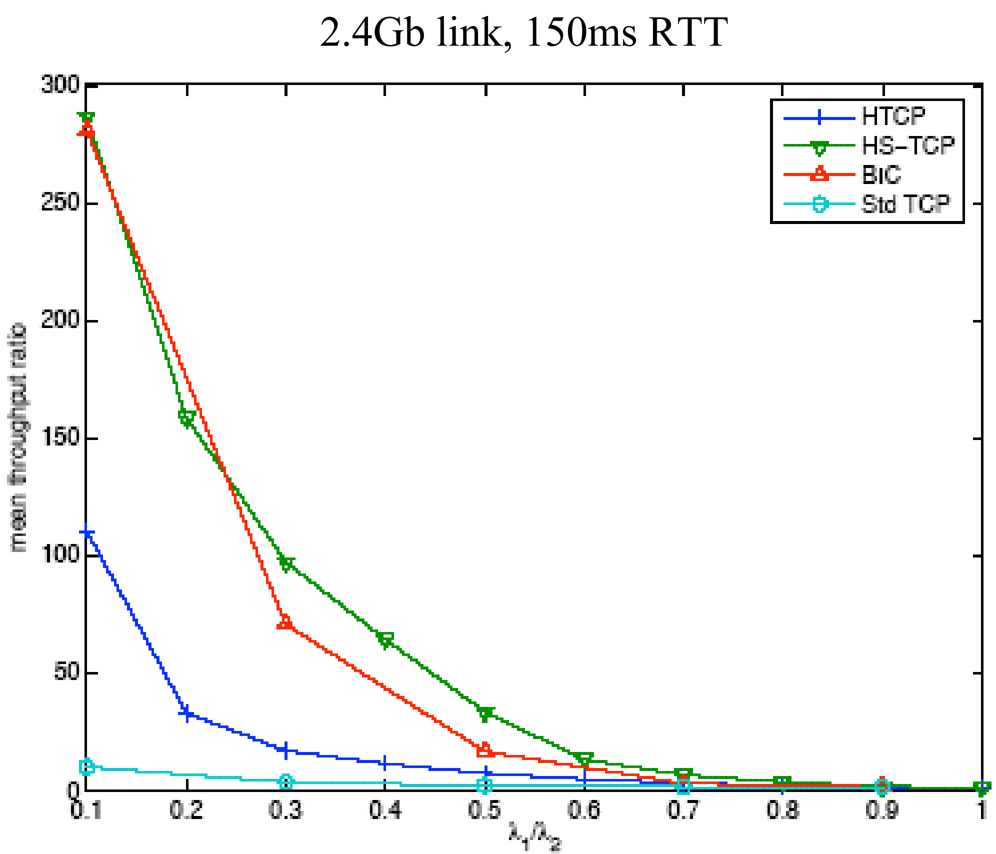


•Dumbell topology, 10 flows, same RTT.

•Modified queue at bottleneck link so that we can adjust the flow λ's. Here, we use $\lambda_1=1$ (flow 1) and $\lambda_i=\lambda$, $i\in[2,10]$ (other other flows).

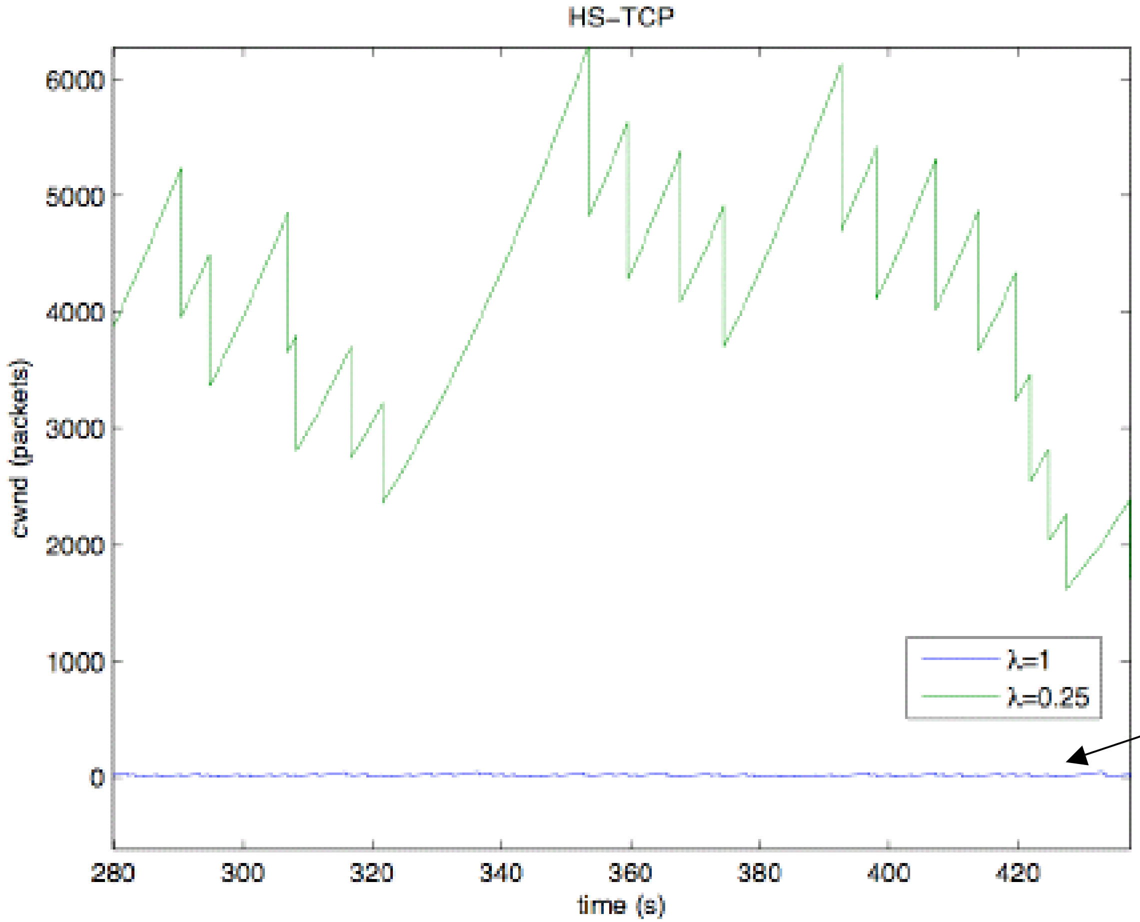
Long-Term Unfairness - NewTCP

Results:



Long-Term Unfairness - NewTCP

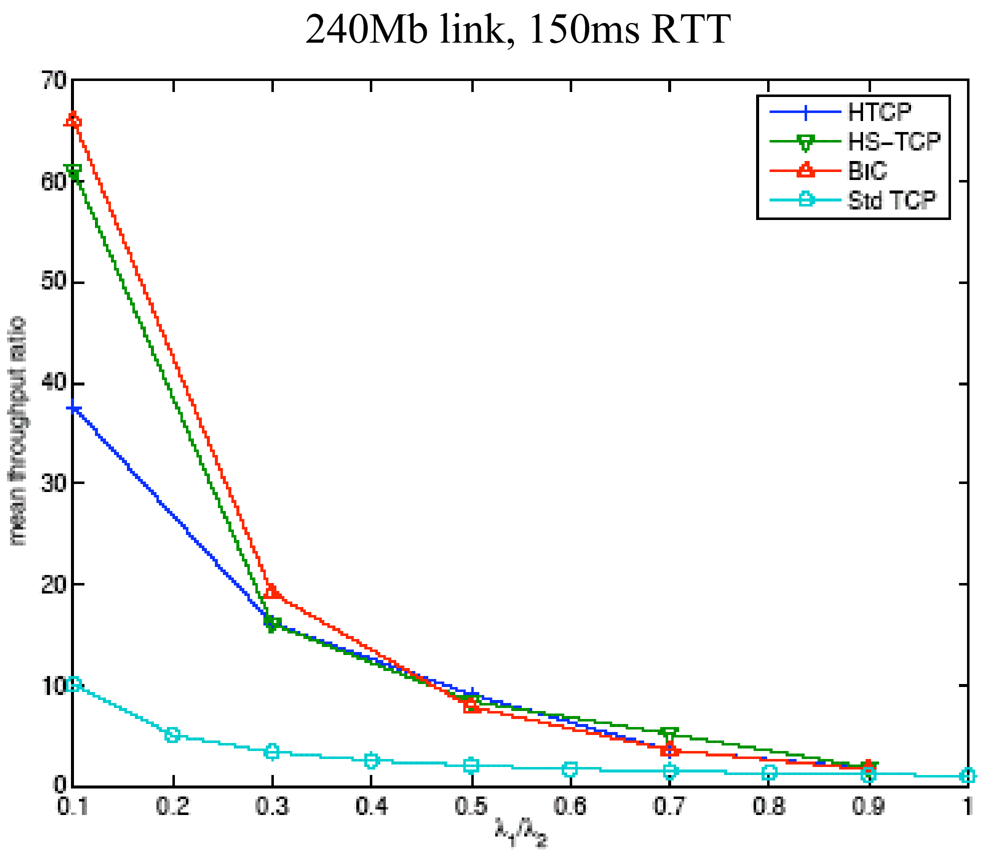
Example of >100:1 unfairness:



Mean cwnd at congestion 27 packets

Long-Term Unfairness - NewTCP

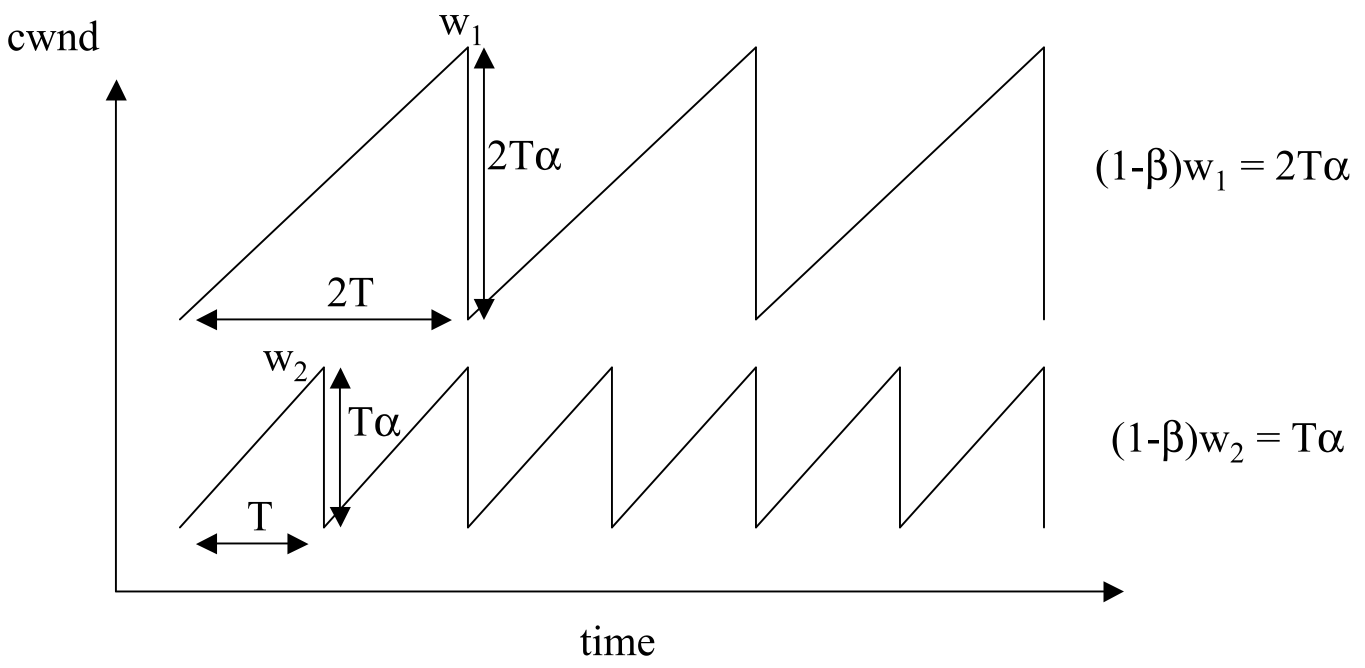
The level of unfairness depends on BDP:



NB: Unfairness lower here, as BDP 10 times smaller. But, unfairness also gets larger when BDP is increased ie. Previous slide is **not** the worst case.

Long-Term Unfairness - NewTCP

Why is this happening ? Recall insight from simple periodic case with standard TCP



But NewTCP flows all seek to become more aggressive when they detect a high BDP path.

HS-TCP - increase α with cwnd.

H-TCP - increase α with congestion epoch duration

BIC

→ high-speed action reinforces basic AIMD unfairness when differences in synchronisation rate.

Long-Term Unfairness - NewTCP

Why is this happening ?

High-speed action reinforces basic AIMD unfairness when differences in synchronisation rate ….

This seems a problem for all approaches considered.

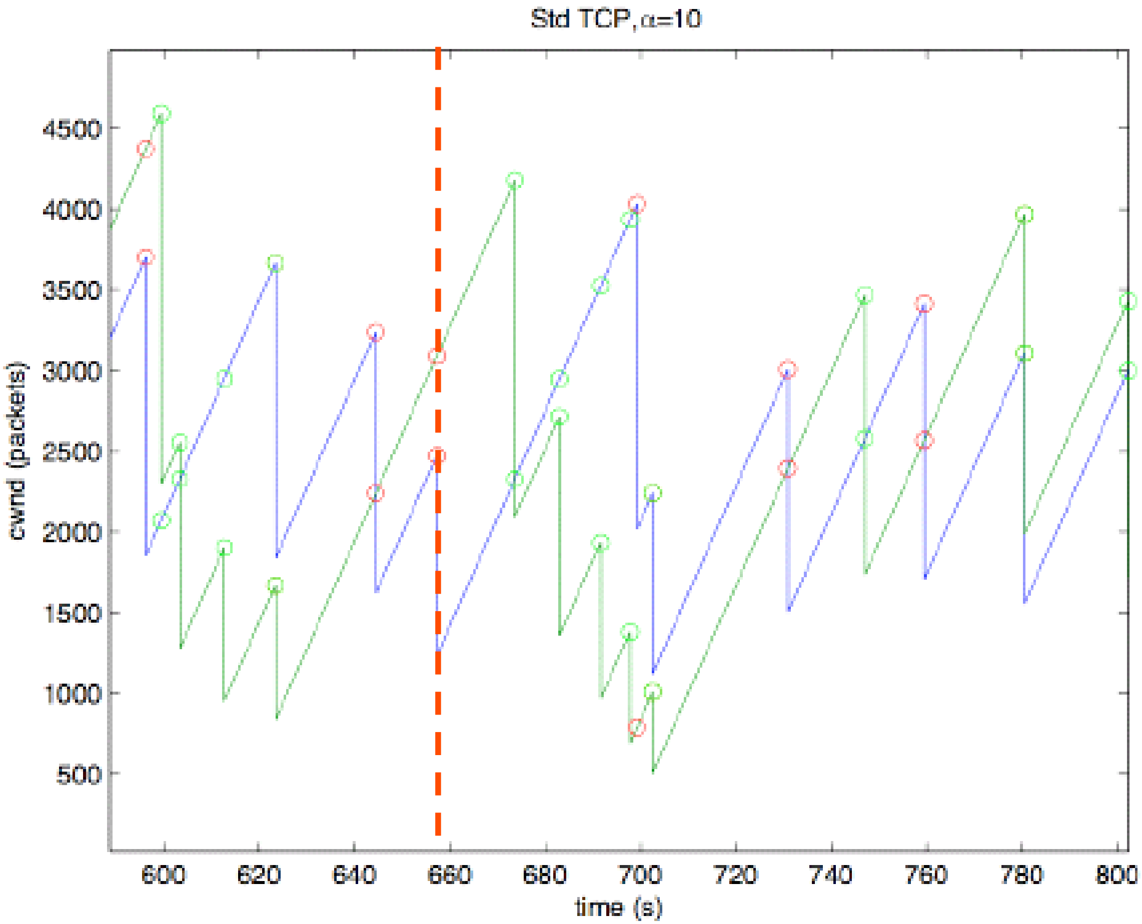
Is it something fundamental/unavoidable ?

Some unfairness seems inevitable when local sensing of network conditions is used and local view of network is non-homogeneous.

But does it need to be so extreme ? Note that if use standard linear AIMD increase unfairness scales linearly (not so bad) … so its something to do with high-speed adaptation. Come back to this later.

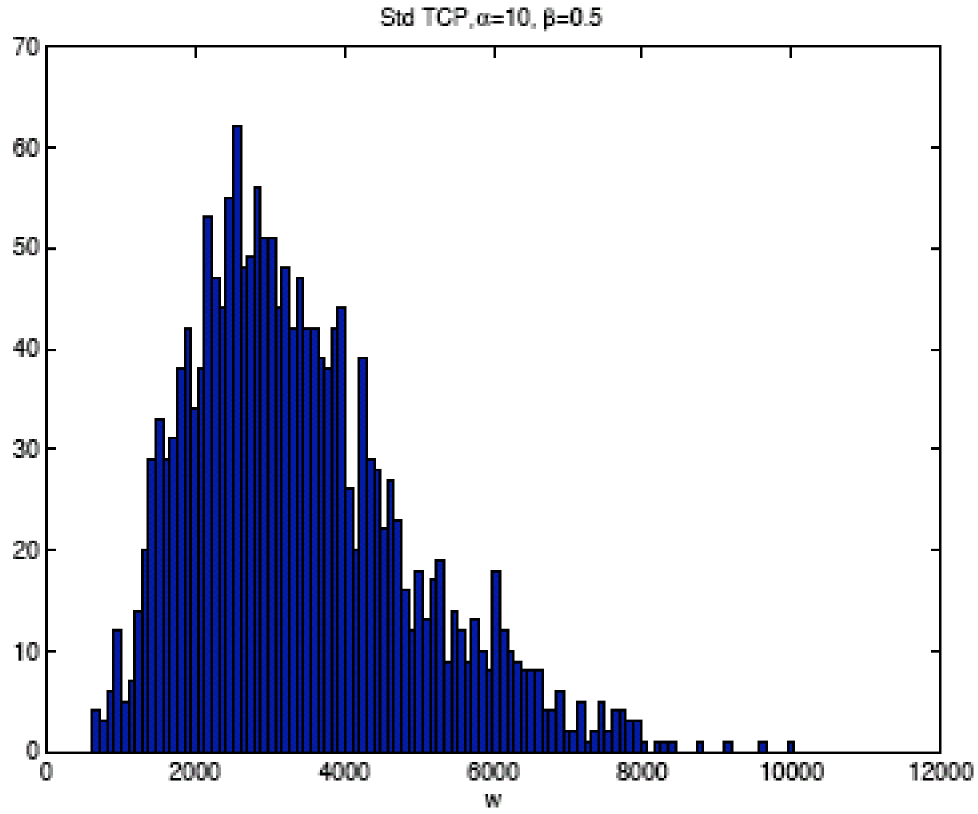
Short-Term Unfairness in Standard TCP

What do we mean by "short-term fairness" in context of long-lived flows ?



Short-Term Unfairness in Standard TCP

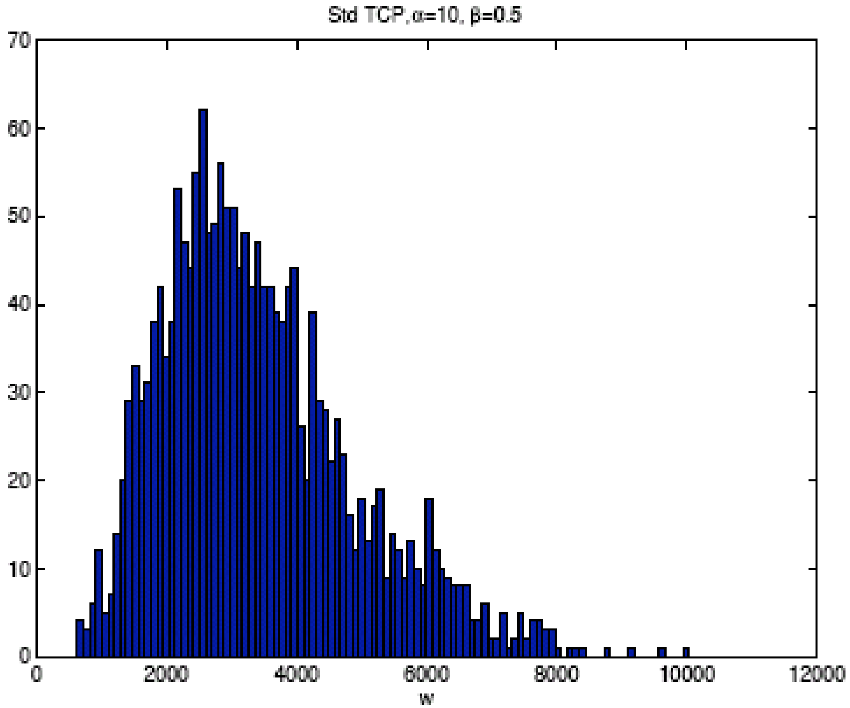
Start by looking at cwnd distribution for one flow. How much does it vary ?
10 flow example, λ=0.25:

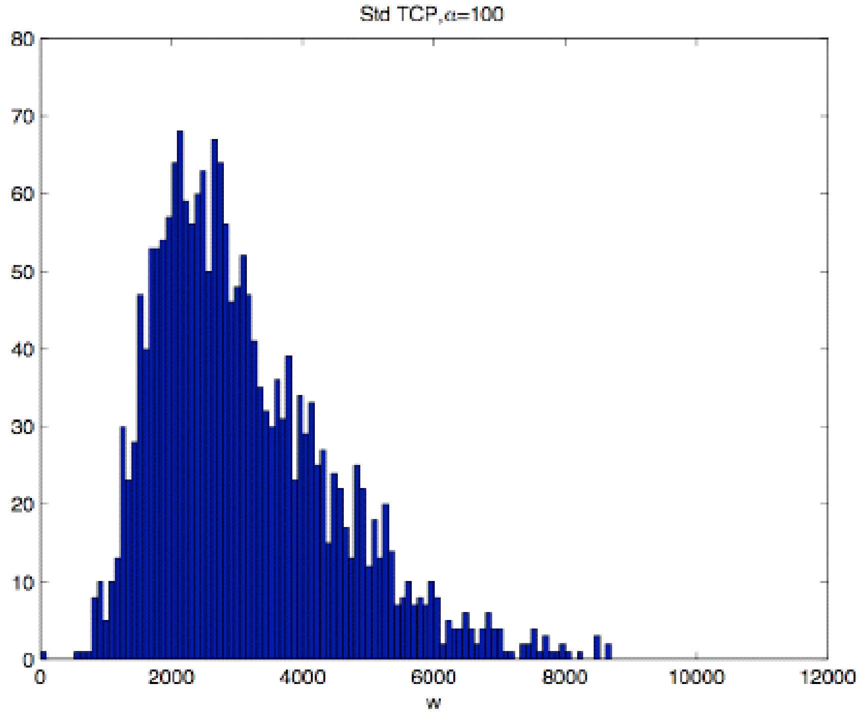


2.4Gb link, 150ms RTT

Short-Term Unfairness in Standard TCP

Impact of varying α on cwnd distribution.
10 flow example, λ=0.25:





2.4Gb link, 150ms RTT

This invariance property is not at all obvious. Has anyone noticed it before ?

Short-Term Unfairness in Standard TCP

Revisit standard TCP stochastic dynamics. $w_i(k+1) = \beta_i(k) w_i(k) + \alpha T(k)$

Collect update equations for individual flows together and write in matrix form. For network of n sources we have:

$$W(k+1) = \mathbf{A}(k) W(k)$$

where $W^T(k)=[w_1(k), w_2(k), \ldots, w_n(k)]$ is the vector of window sizes at congestion and

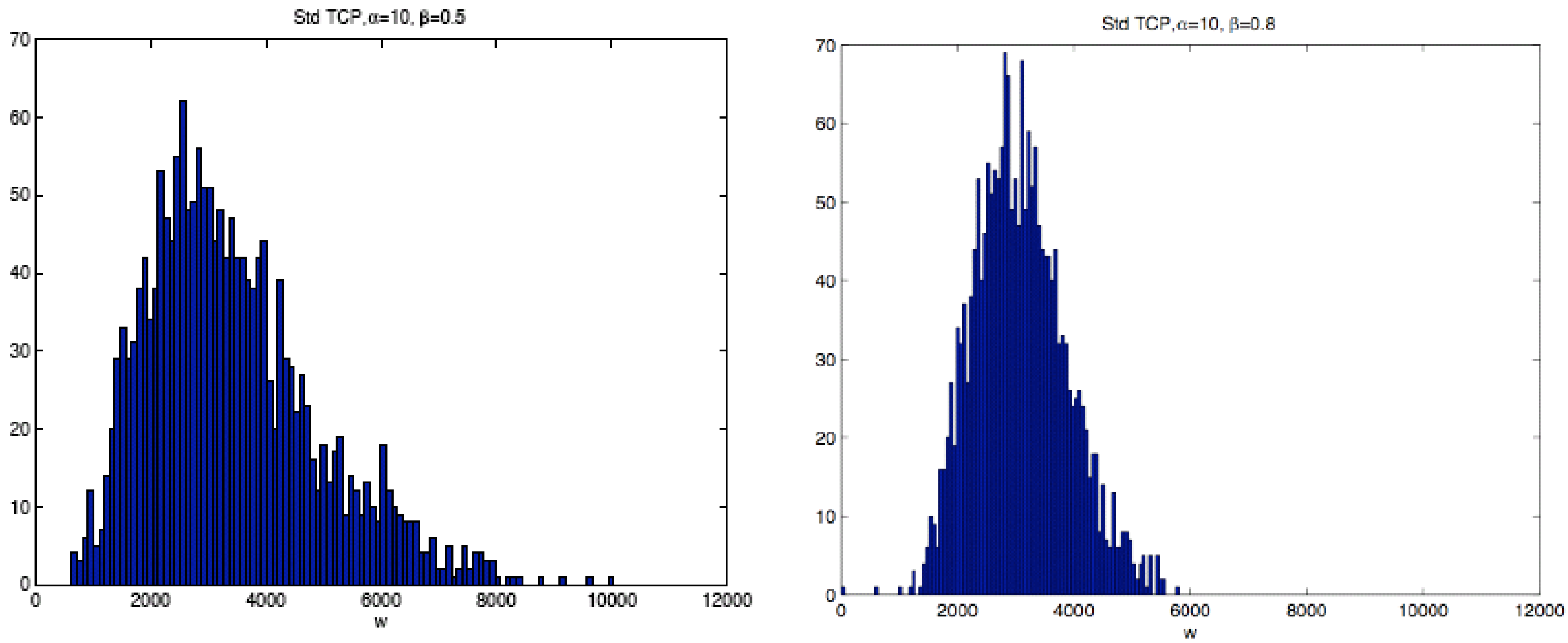
$$\mathbf{A}(k) = \begin{bmatrix} \beta_1(k) & 0 & \cdots & 0 \\ 0 & \beta_2(k) & \cdots & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & \beta_n(k) \end{bmatrix} + \frac{1}{\sum_{i=1}^{n} \alpha} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \begin{bmatrix} 1-\beta_1(k) & \cdots & 1-\beta_n(k) \end{bmatrix}$$

Observe that the dynamics do *not* depend on absolute values of the α_i ⇒ short-term unfairness is invariant with scaling of the α_i

Short-Term Unfairness in Standard TCP

Note that varying beta *does* significantly change the distribution.

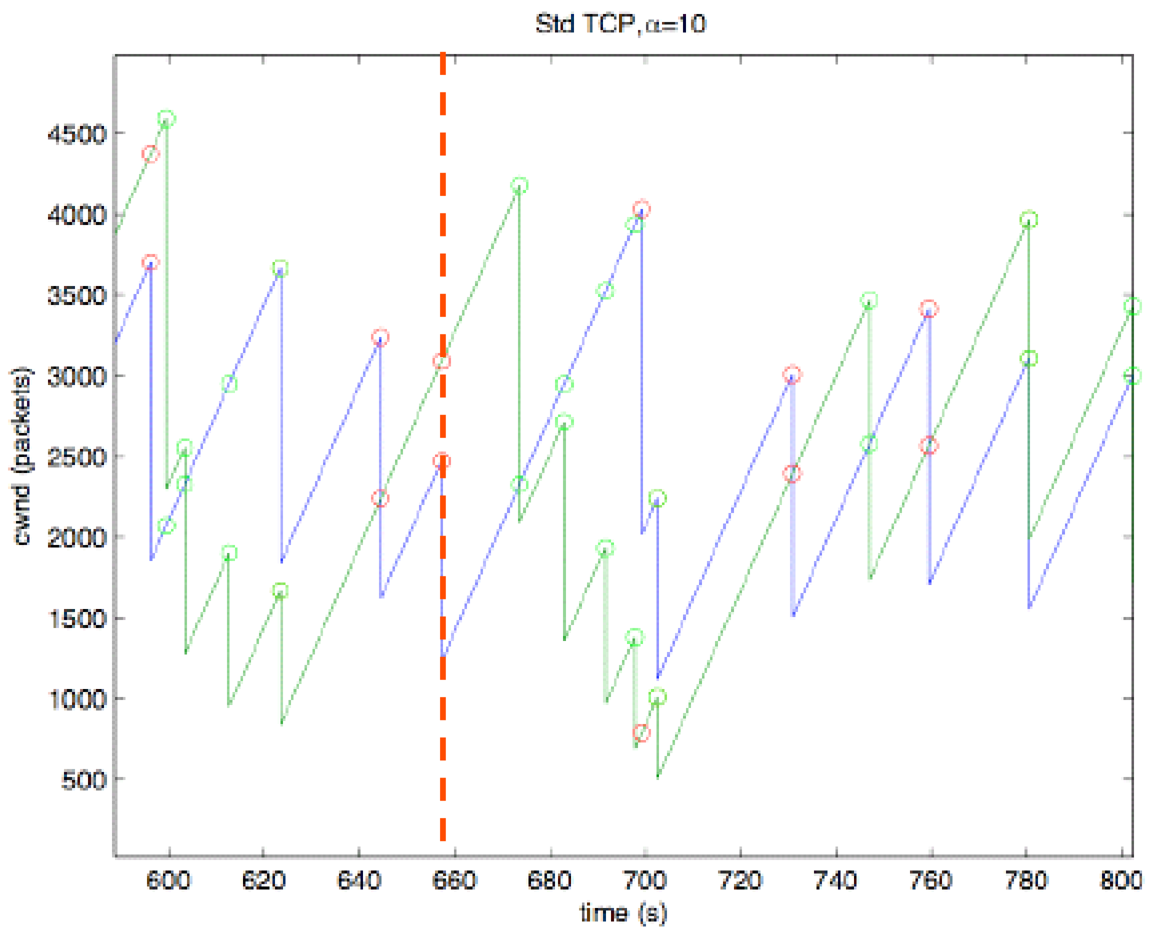
Smaller backoff “stiffens” the network as flows release bandwidth more slowly
→ harder to cause large variations in allocation between flows.



2.4Gb link, 150ms RTT

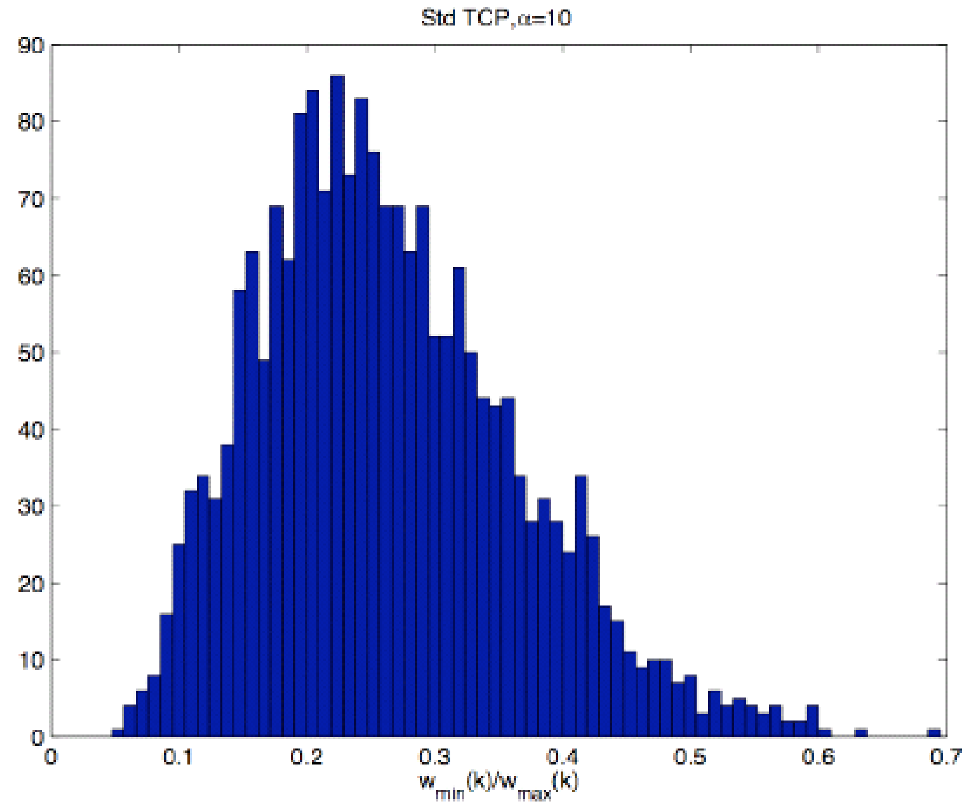
Short-Term Unfairness in Standard TCP

Unfairness - distribution of min cwnd/max cwnd snapshots …



Short-Term Unfairness in Standard TCP

Unfairness - distribution of min cwnd/max cwnd snapshots for flows with same λ.
10 flow example, λ=0.25:



2.4Gb link, 150ms RTT

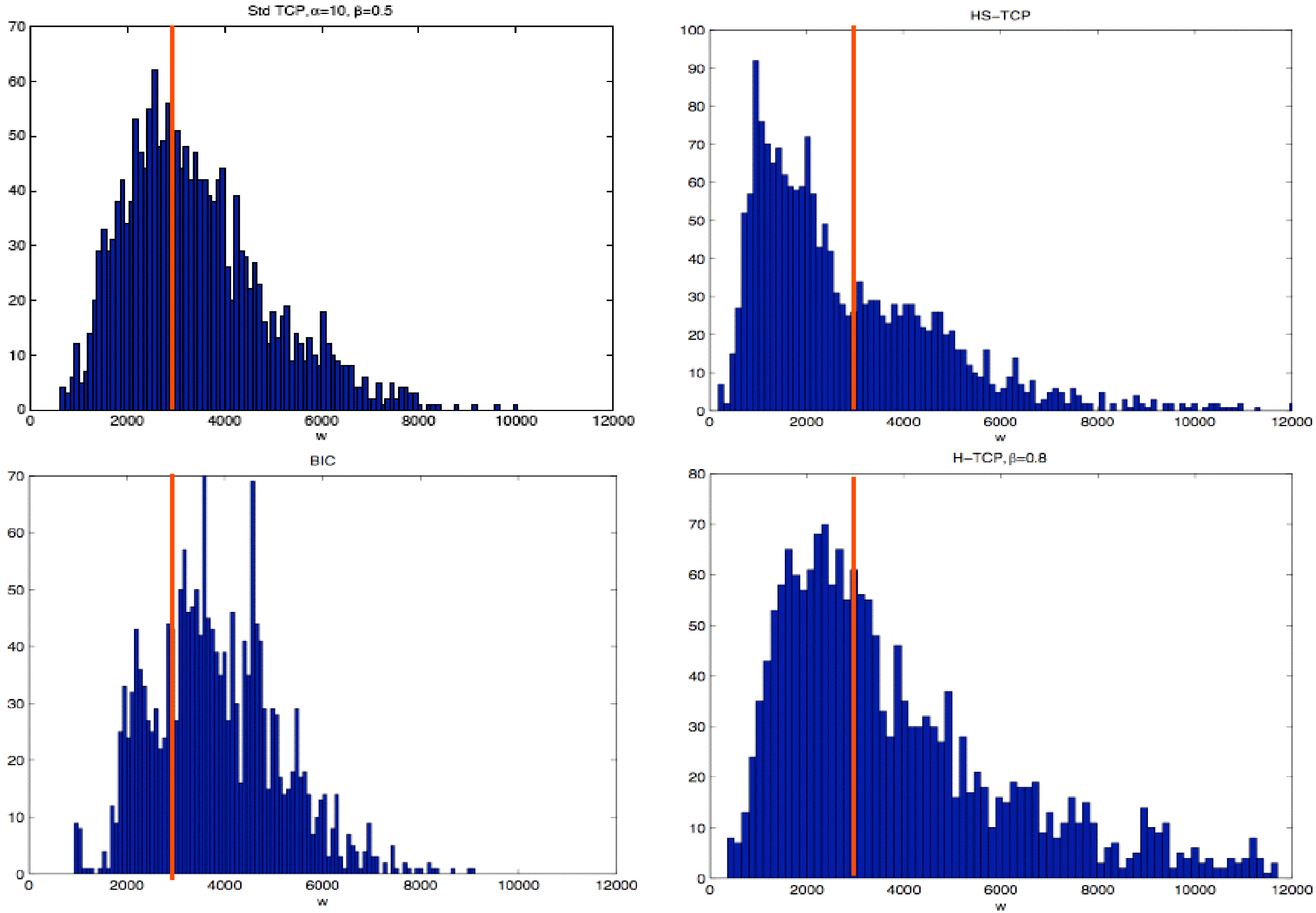
Observe mean unfairness is around 0.25 - seems quite large as flows have same RTT/λ.

Short-Term Unfairness - NewTCP

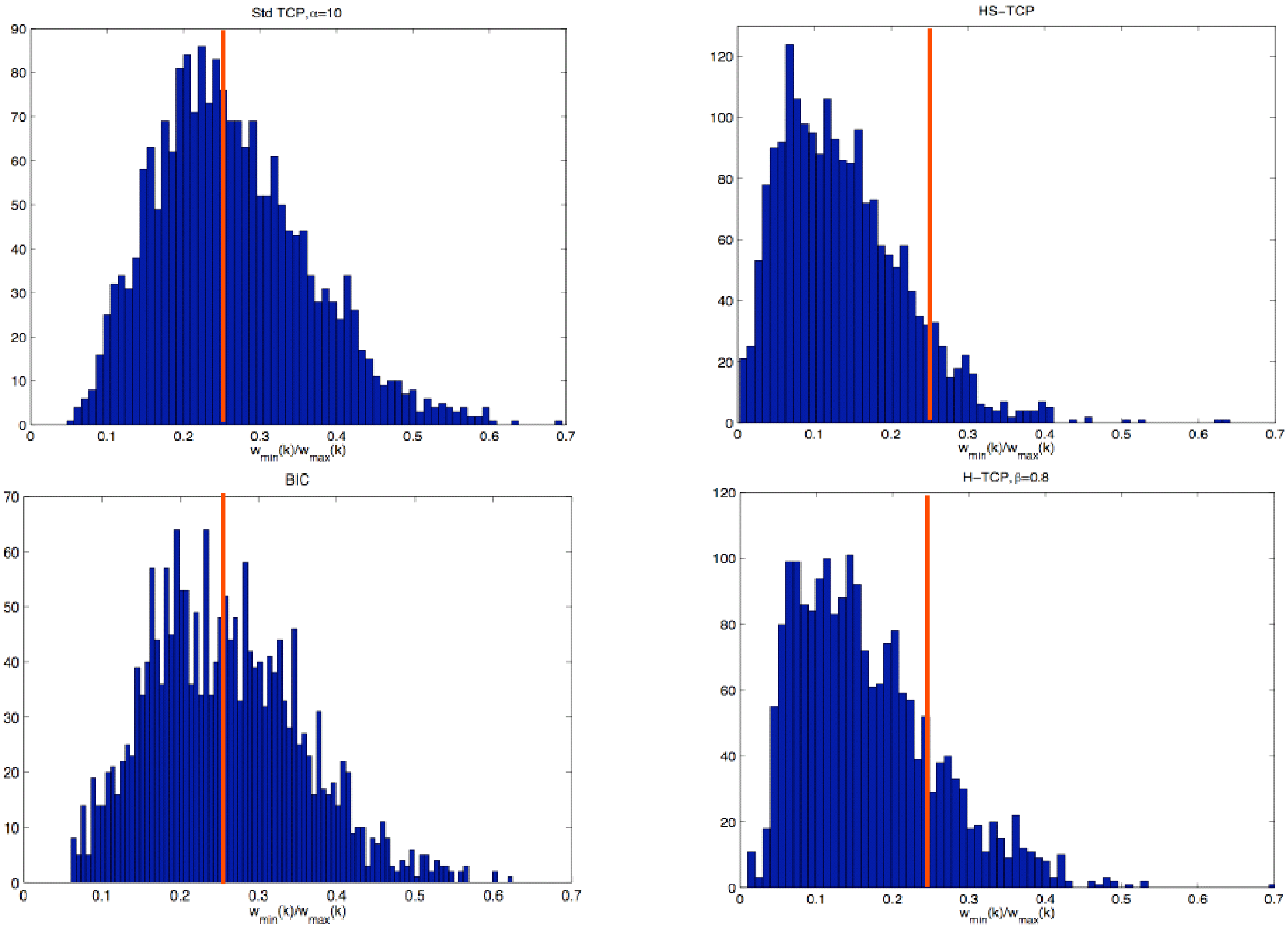
So how do corresponding measurements look for NewTCP proposals ….

Short-Term Unfairness - NewTCP

Cwnd distribution at congestion

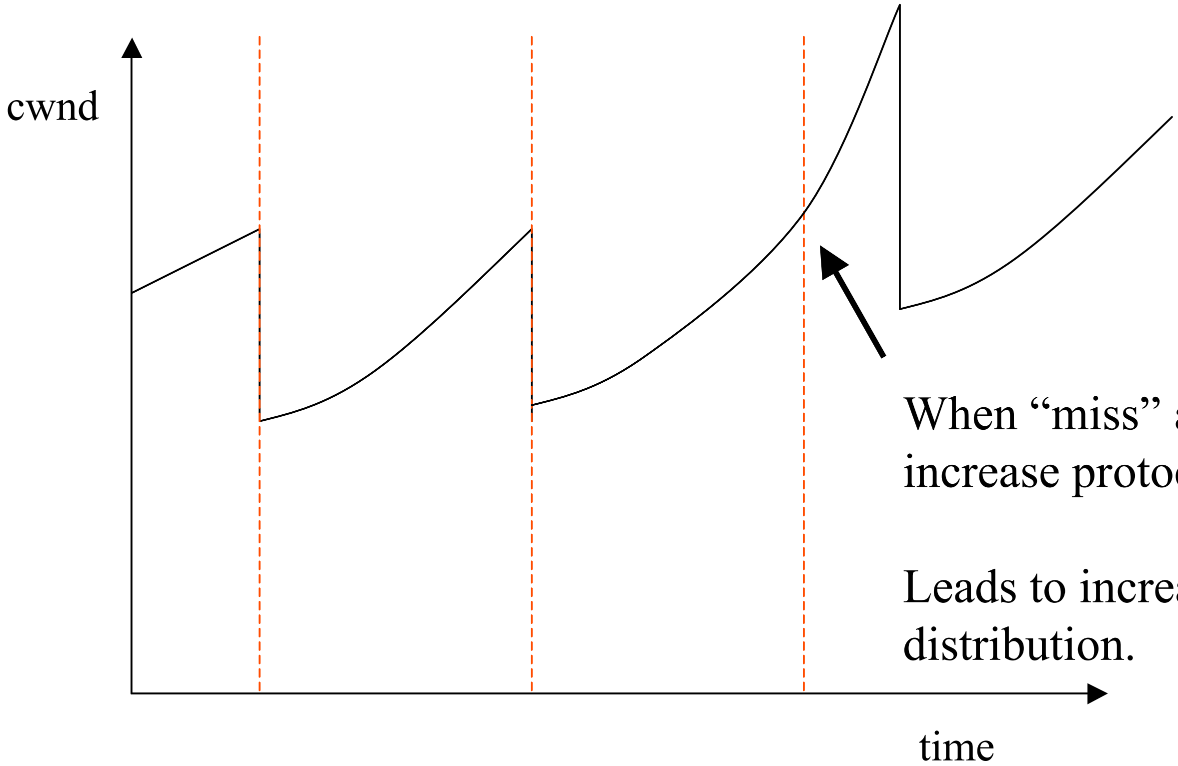


Short-Term Unfairness - NewTCP   min/max cwnd at congestion



Short-Term Unfairness - NewTCP

What’s going on ?



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

Leads to increased variance in cwnd distribution.

Basic problem is that *short-term* variations in drop pattern can induce large changes in increase rate.

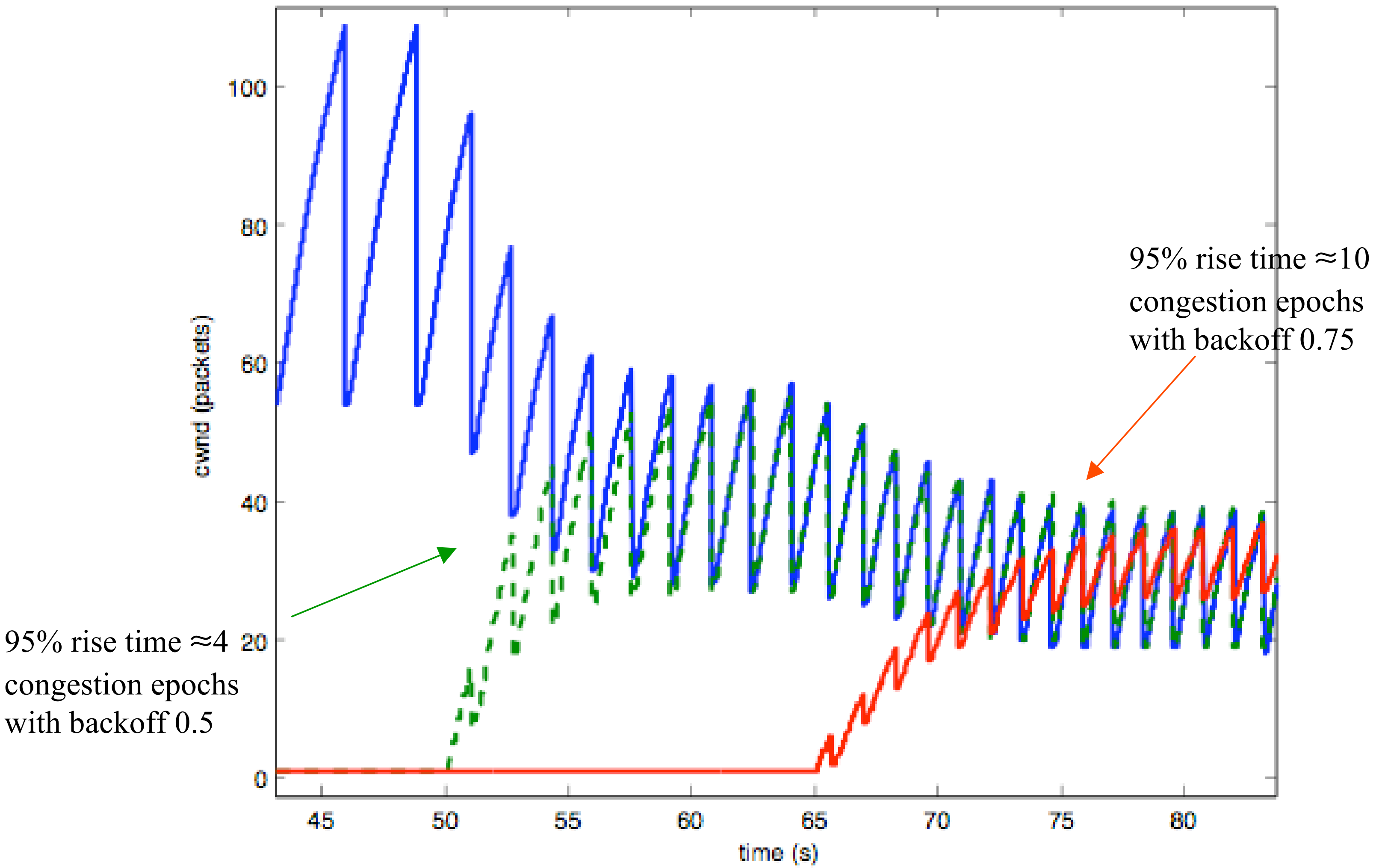
Short-Term Unfairness - NewTCP

What could we do to address this ?

• Could increase β to "stiffen" things up. Flows release bandwidth more slowly and so harder to cause large variations in allocation between flows.

- HS-TCP, BIC both increase β from its standard value of 0.5.
- But we know that increasing β also makes network sluggish to respond to legitimate changes e.g. new flows starting up.

Short-Term Unfairness - NewTCP



Slow convergence here translates into unfairness between connection with different sizes

Short-Term Unfairness - NewTCP

What could we do to address this ?

Our discussion suggests another simple solution, at least to short-term unfairness.

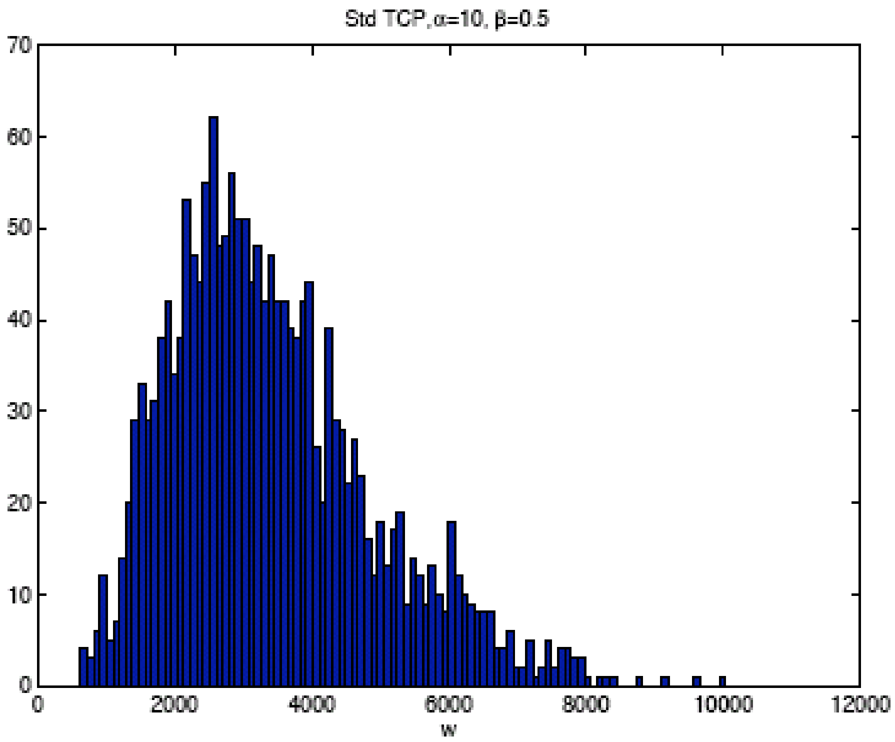
Basic problem is that *short-term* variations in drop pattern can induce large changes in increase rate.

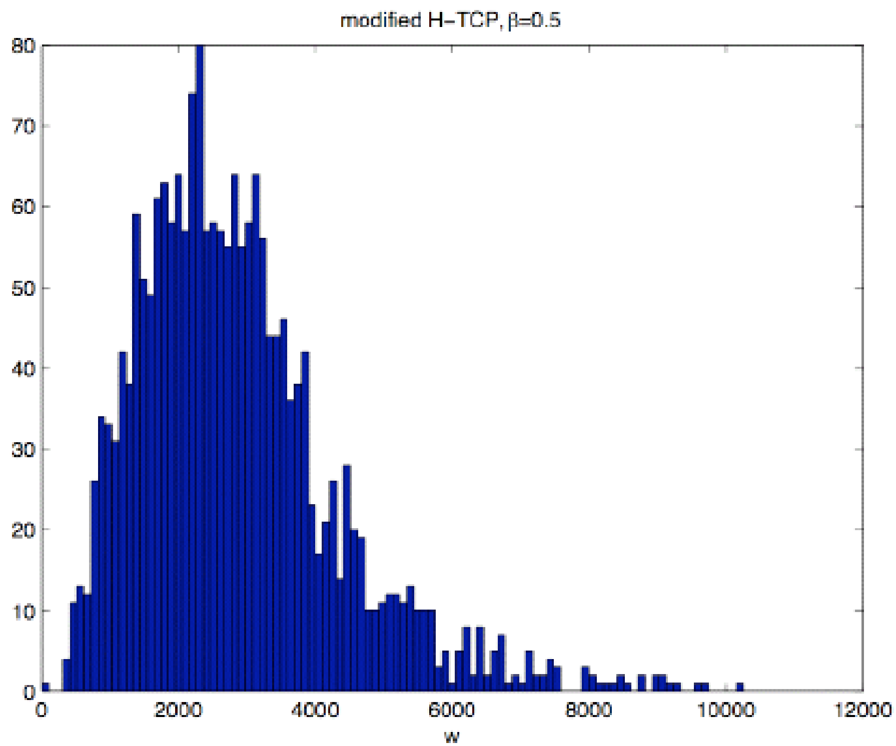
→So why not adapt the aggressiveness of flows on average rather than instantaneous values e.g. on average congestion epoch duration (H-TCP) or average cwnd (HS-TCP).

Easy to implement …

Short-Term Unfairness - H-TCPav

Cwnd distribution at congestion.   10 flow example, λ=0.25:

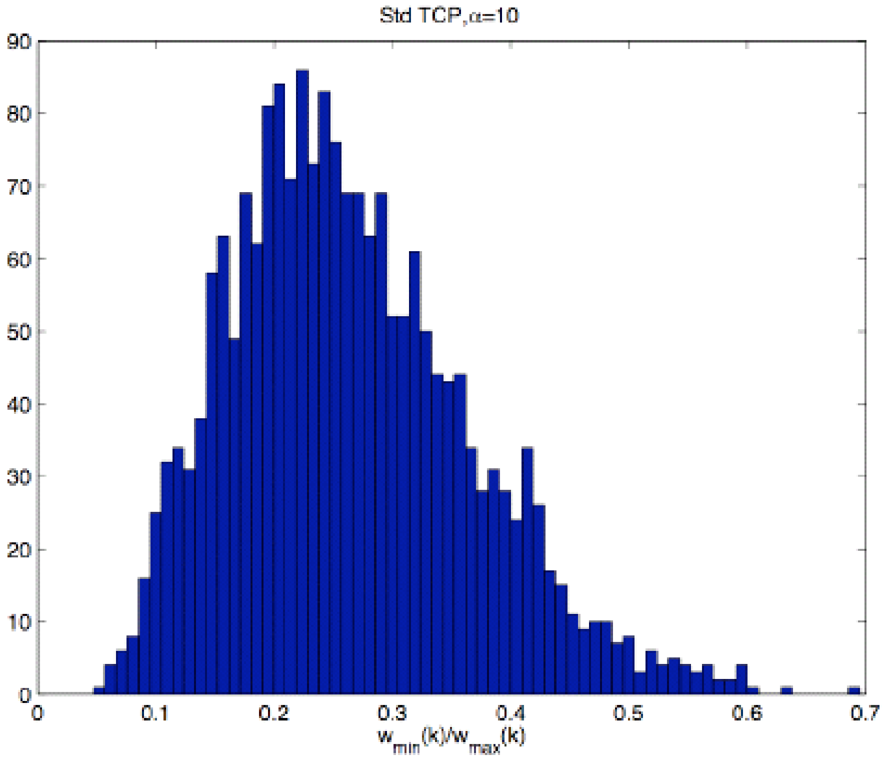


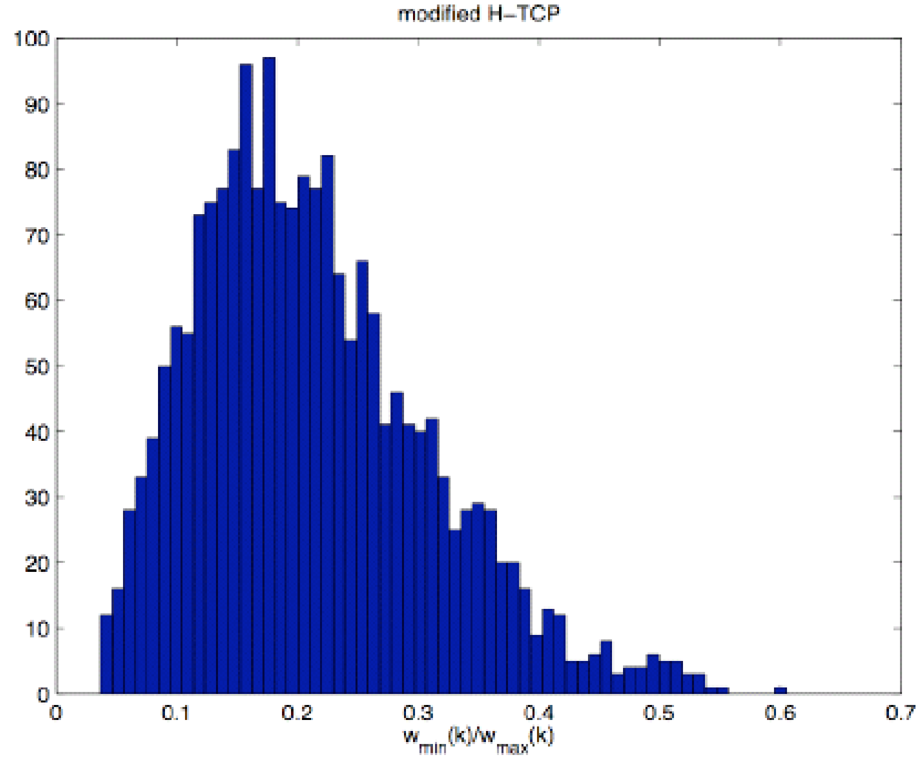


2.4Gb link, 150ms RTT

Short-Term Unfairness - H-TCPav

min/max cwnd at congestion. 10 flow example, λ=0.25:





2.4Gb link, 150ms RTT

Short-Term Unfairness - H-TCPav

Any side-effects of this change ?

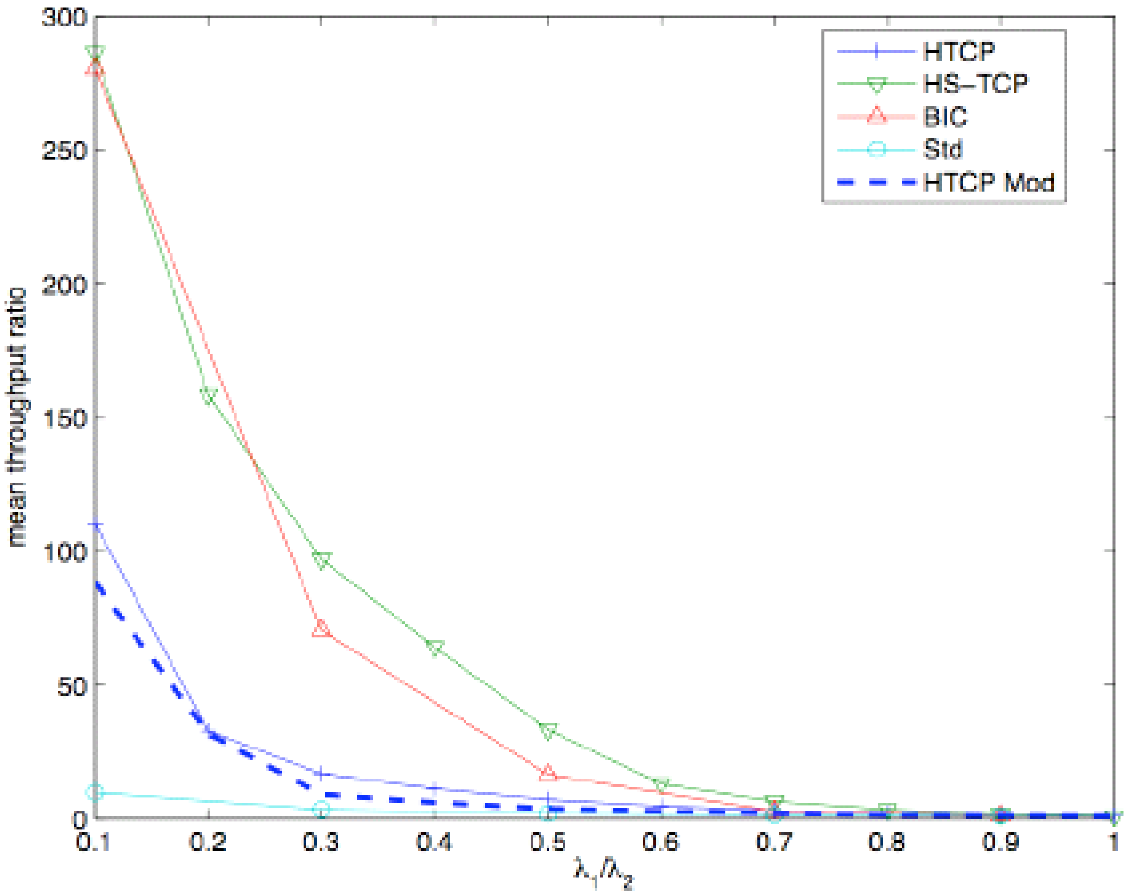
•Now inherit similar short-term unfairness properties as standard TCP.

•No change to long-term properties
- RTT unfairness characteristics.
- efficiency vs queue provisioning (backoff is unchanged).
- friendliness

•Does reduce responsiveness to changes in network conditions (new flows starting etc).
- worst case is when abruptly go from 1 flow to many flows i.e. have sudden large change in congestion epoch duration.
- impact depends on our averaging horizon.

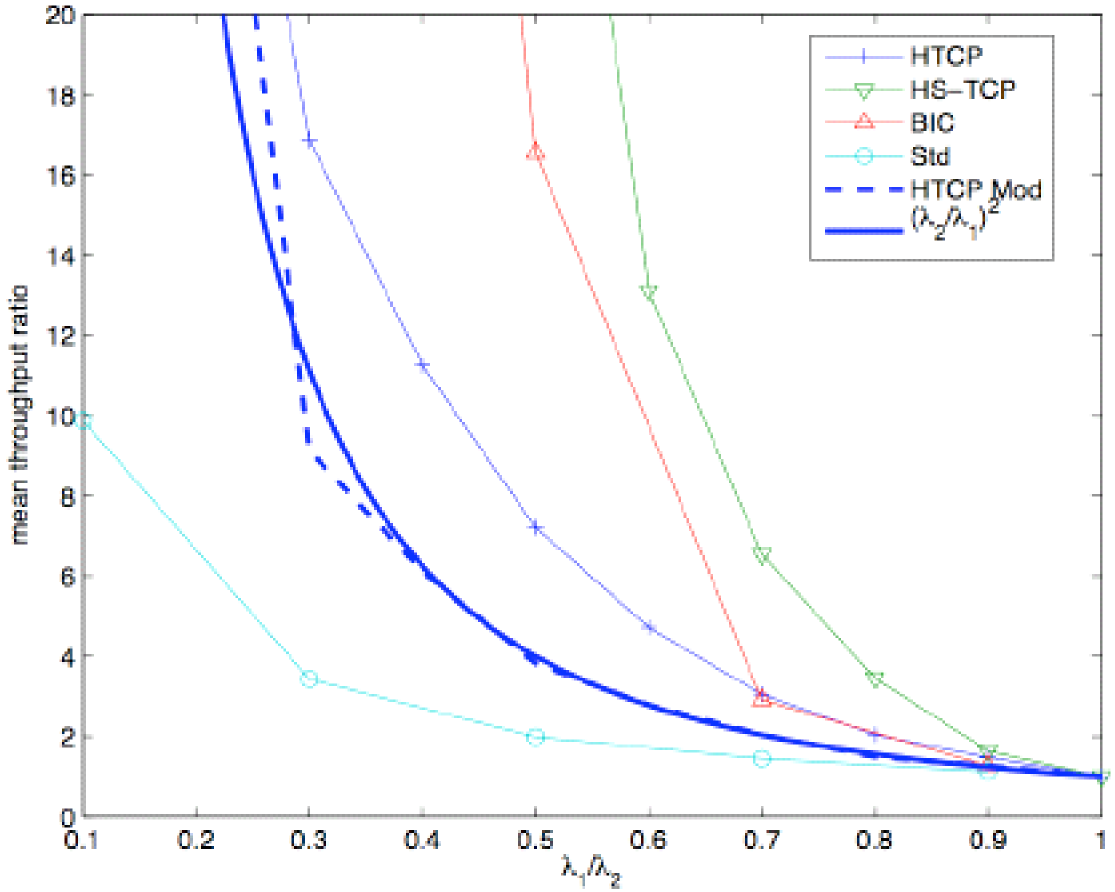
•Impact on long-term unfairness due to differences in synchronisation rate ?

Long-Term Unfairness



2.4Gb link, 150ms RTT

Long-Term Unfairness



2.4Gb link, 150ms RTT

Why quadratic ?

Recall, steady-state formula:

$$\bar{w} = \frac{\alpha \bar{T}}{(1-\bar{\beta})} = \frac{\alpha}{(1-\beta)} \frac{\bar{T}}{\lambda}$$

We are adjusting α as a function of mean congestion epoch duration seen by that flow, which is roughly $\bar{T}/\lambda$

$$\frac{w_1}{w_2} = \frac{\alpha_1 (\bar{T}/\lambda_1)}{\alpha_2 (\bar{T}/\lambda_2)} \frac{\lambda_2}{\lambda_1} = \left(\frac{\lambda_2}{\lambda_1}\right)^2$$

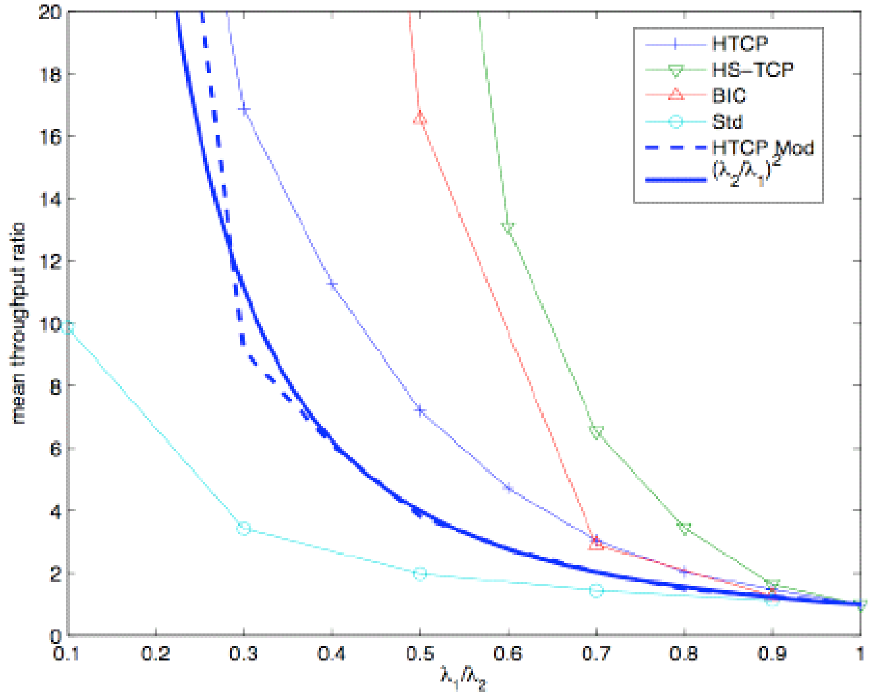
when α_i is scaled linearly with $\bar{T}/\lambda_i$

Long-Term Unfairness

Still quadratic though.
→so have ×4 unfairness for λ difference of 2, ×16 for λ difference of 4.

Is this ok ?

λ differences in the range from 1-4 seems reasonable, but we really need some measurements on what happens in reality.



Summary

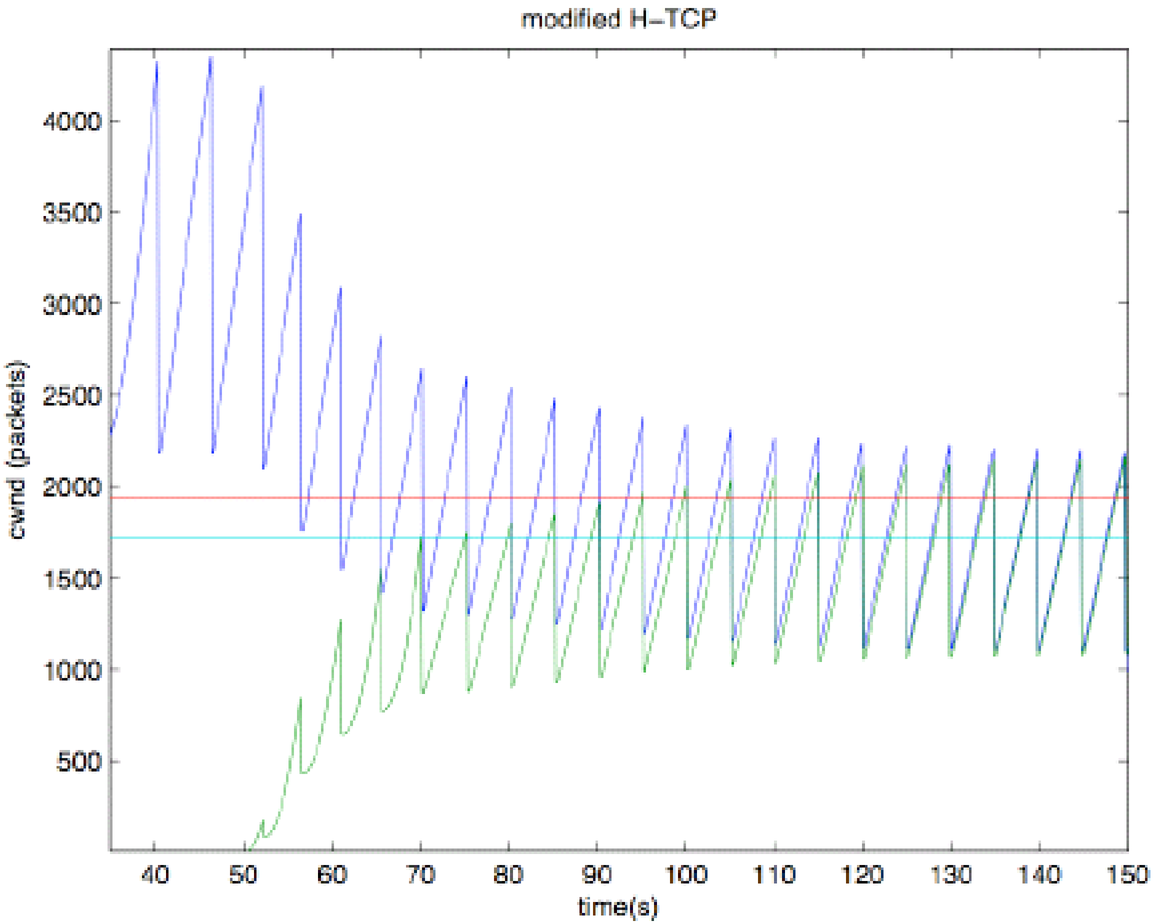
•Discussed impact of loss of synchronisation.
 - Highlighted gross long-term λ unfairness of all protocols considered.
 - Highlighted increased (compared with standard TCP) short-term unfairness of HS-TCP and H-TCP.

•Suggested a simple, principled fix for short-term unfairness. Leaves long-term properties unchanged. Illustrated with H-TCP, but could also be applied to HS-TCP.

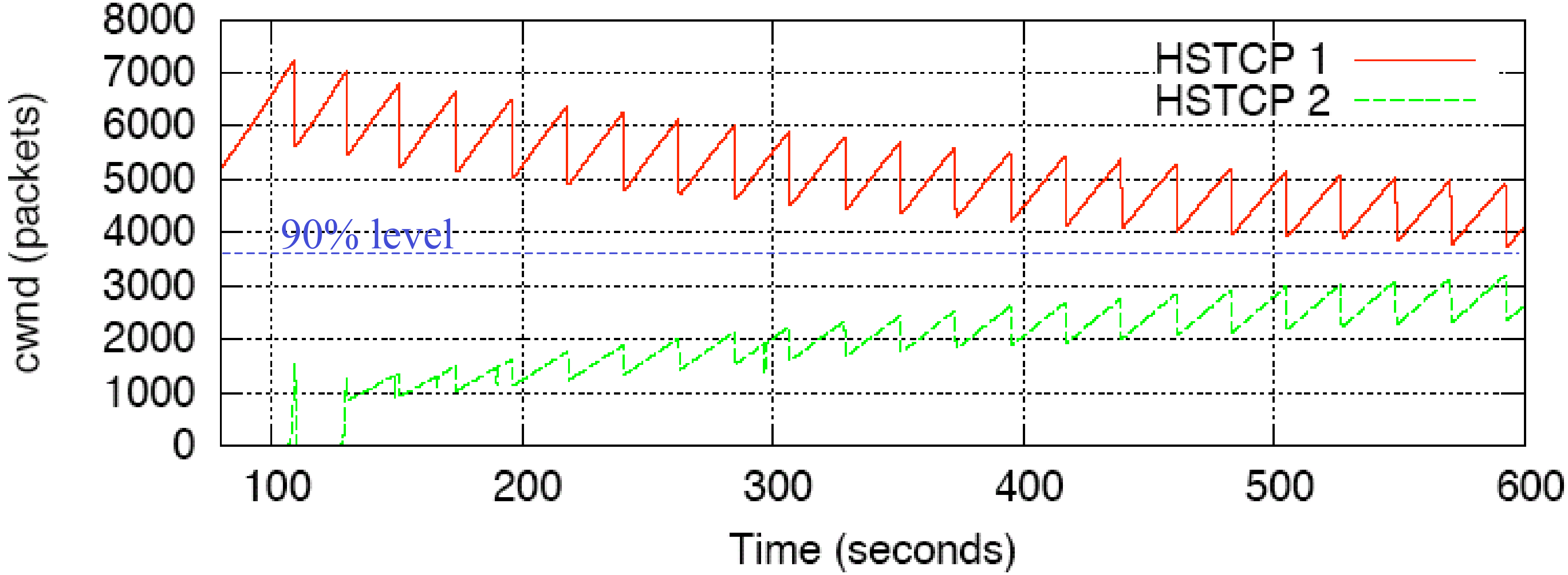
•No fix for long-term λ unfairness, although have reduced it to being quadratic. Is this enough ?

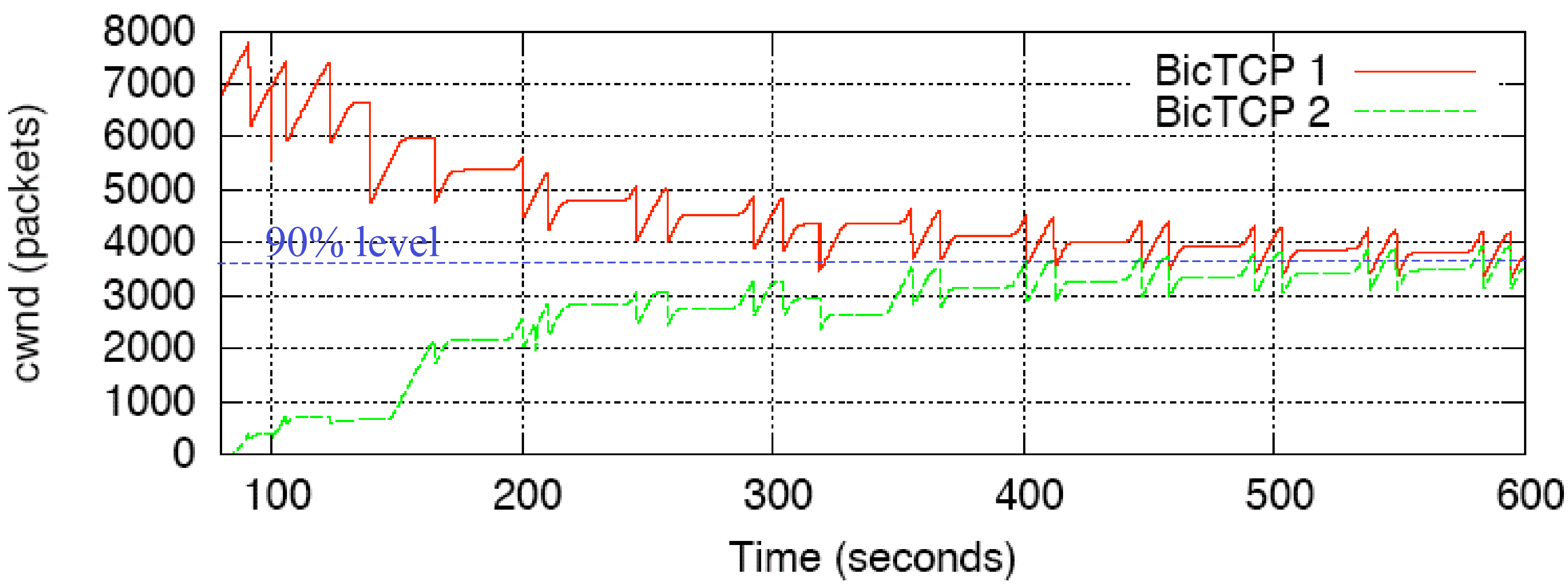
•TCP-AQM co-design ?

•These sort of tests seem like they might make a good addition to benchmarks for evaluating TCP proposals.

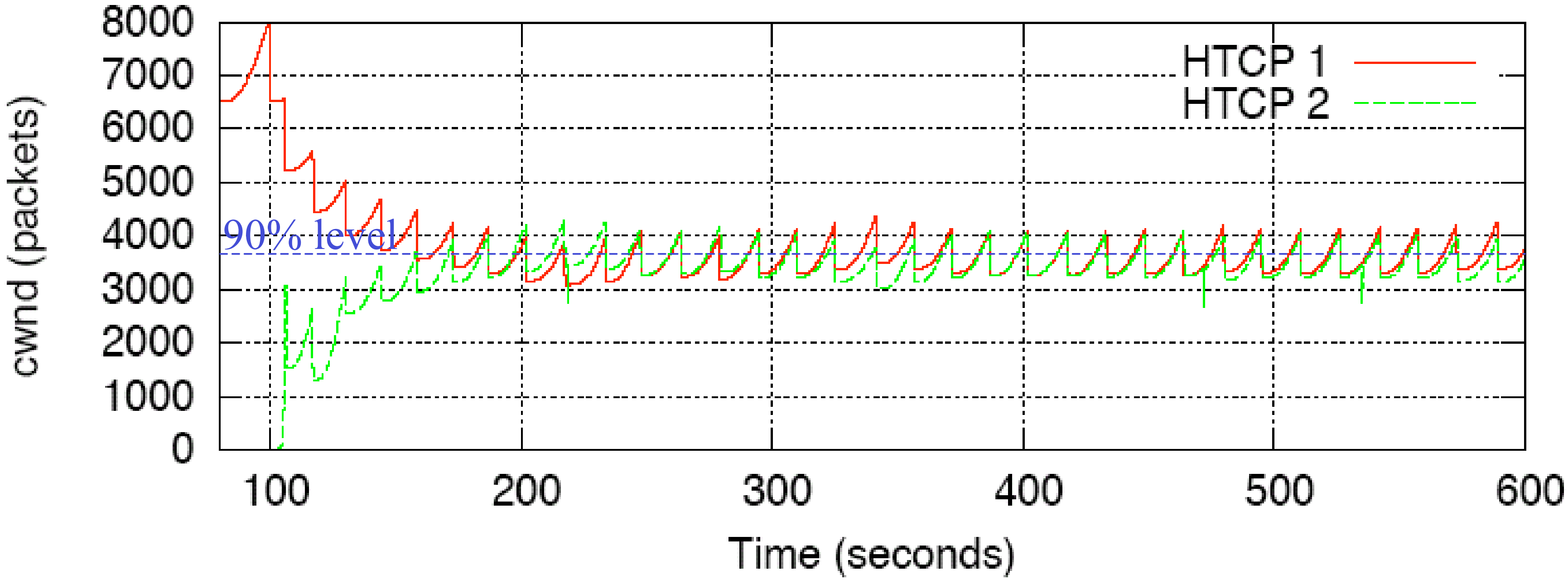


500Mbs, 100ms RTT


8000
7000
6000
5000
4000
3000
2000
1000
0
cwnd (packets)
100
200
300
400
500
600
Time (seconds)
HSTCP 1
HSTCP 2
90% level



8000
7000
6000
5000
4000
3000
2000
1000
0
cwnd (packets)
100
200
300
400
500
600
Time (seconds)
BicTCP 1
BicTCP 2
90% level


8000
7000
6000
5000
4000
3000
2000
1000
0
cwnd (packets)
100
200
300
400
500
600
Time (seconds)
HTCP 1
HTCP 2
90% level