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

D.J. Leith and R. Shorten
Hamilton Institute, National University of Ireland, Maynooth, Ireland
{robert.shorten, doug.leith}@nuim.ie

***Abstract*— In this paper we consider the performance of several well known high speed protocols in environments where individual flows experience different probabilities of seeing a drop in drop-tail buffers. Our initial results suggest the properties of networks in which these protocols are deployed can be sensitive to changes in these probabilities. Our results also suggest that AQM-protocol co-design may be helpful in mitigating this sensitivity.**

I. Introduction

In recent years, several new TCP congestion control algorithms have been proposed for deployment in long-distance and high-speed networks. A primary objective in developing these algorithms has been to achieve improved scaling of performance with increasing bandwidth. In particular, most authors have focussed on developing AIMD-like algorithms whose probing behaviour becomes more aggressive as bandwidth increases: BIC-TCP[10], Scalable-TCP[5], HS-TCP[4], and H-TCP[6] all fall into this category. A basic problem with standard TCP, when deployed on such links, is that the time taken by a flow to recover after a backoff event can be prohibitively long, thereby leading to long data transfer times. The aforementioned protocols all seek to solve this problem by probing more aggressively in high-speed environments.

Unfortunately, adjusting the manner in which individual flows probe for available bandwidth serves not only to keep the time between consecutive congestion events short, but it also changes the way in which flows compete for available bandwidth. In fact, the fundamental properties of such networks may be very different to networks of standard TCP flows, and while some aspects of their behaviour have been expored, many fundamental questions pertaining to their behaviour remain unanswered. The purpose of this note is to address, in part, this basic observation. In particular, our objective is to explore the 'cost of missing drops' for high speed protocols. Three important questions arise in this context.

(i) The first of these is related to the long-term behaviour of networks in which different flows have differing synchronisation rates. By synchronisation rate λ we mean the proportion of network congestion events at which a flow experiences packet loss (thus the synchronisation rate is 1 when a flow sees a drop at every network congestion event). It is known[9] that networks of TCP flows are well behaved with respect to changes in synchronisation rate; namely, long-term relative bandwidth allocation amongst competing flows scales inversely with synchronisation rate. For high-speed protocols, however, the manner in which individual flow synchronisation rates impact network behaviour is currently not clear. If the allocation of bandwidth amongst competing flows is very sensitive to synchronisation rate, then one concern is that this may lead gross unfairness in the throughputs achieved by flows experiencing different synchronisation rates.

(ii) The second important issue is concerned with the short-term variations in rate that arise when networks are unsynchronised. For very aggressive protocols, missing a drop may result in an individual flow temporarily seizing a large proportion of the network bandwidth. As a result, while flow throughputs might average out to be fair over long time-scales, they may be very unfair over short time-scales.

(iii) Thirdly, the interaction of some of these new protocols and AQM's is an unexplored topic. The work documented here, namely characterising the variation in throughput for each flow as a function of synchronisation rate, represents a first step in this direction.

In this paper we present initial results on the above topics. We begin with a basic review of TCP. We then present empirical results for a number of well-known protocols. Finally, we discuss modifications for these protocols to reduce the side-effects of aggressive window growth.

II. Properties of networks employing TCP

The standard TCP congestion control algorithm updates the congestion window $cwnd$ according to an Additive Increase Multiplicative Decrease (AIMD) control law. In the congestion avoidance phase, when a source i receives a TCP ACK, it increments $cwnd$ according to $cwnd \rightarrow cwnd + \alpha/cwnd$ where $\alpha = 1$ for the standard TCP algorithm. When packet loss is detected, $cwnd$ is reduced by a backoff factor β: thus $cwnd \rightarrow \beta cwnd$, where $\beta = 0.5$ for standard TCP.

The properties of networks that employ standard TCP are well known and have been reviewed in a number of publications. In particular, it has been shown by a number of authors (e.g. see [9], [8]) that:

$$E(w_i) = E(T)\frac{\alpha_i}{\lambda_i(1-\beta_i)RTT_i}, \qquad (1)$$

where $E(w_i(k))$ denotes the mean window size of i'th flow at the k'th network wide congestion event, $E(T)$ is the average time between network congestion events, λ_i is the synchronisation rate of the $i'th$ flow (assumed to be constant) and RTT_i

is the round-trip-time of the i'th flow (again assumed to be approximately constant). A number of important properties are evident.

(i) λ **unfairness.** It can be seen that, for standard TCP, the long-term unfairness between flows due to different synchronisation factors is an inverse linear function of synchronsation rates, see Figure 1. This is independent of the AIMD increase parameter α (so long as α is same for all flows) and is independent of the path bandwidth-delay product.

(ii) **Short-term unfairness.** Unfairness between flows over a short time-scale is related to the variance of the window variables of the network flows. For a given set of synchronisation rates, the variance of the i'th flow is directly related to (a) the amount of bandwidth that is released by the network at each congestion event and (b) to the speed at the i'th flow acquires this bandwidth. Intuitively, the variance therefore depends on the network backoff factors and the speed at which flows grab bandwidth from other sources in the network.

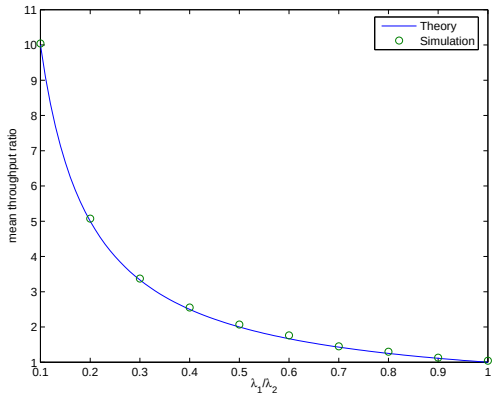

Fig. 1. Impact on throughput fairness of differences in flow synchronisation factor for standard TCP (150ms propagation delay, 2.4Gbs bottleneck bandwith, 10 TCP flows.

III. High-Speed Protocols

We now present preliminary empirical results on λ-unfairness and short-term unfairness for H-TCP, BIC-TCP and HS-TCP. We do not present results for Scalable TCP as this protocol suffers from basic stability issues [12], [11]. A brief description of these protocols is given in the Appendix. It is important to emphasise that all of the results presented are for long-lived flows. Hence, for example, when we discuss short-term unfairness it refers to the level of short-term variations in the throughput of competing long-lived flows. Fairness between short-lived flows, or between flows with different connection lengths, is not considered here.

A. λ Unfairness

We begin this section by documenting the effect of synchronisation rate on the long-term average throughput fairness between competing flows: see Figure 2. Our measurements reveal a considerable sensitivity of the flow rate allocation to differences in synchronisation rate for all of the high speed protocols. For example, $\frac{\lambda_1}{\lambda_2} = 0.5$ induces a $6:1$ ratio of flow throughputs for H-TCP, a ratio of $20:1$ for BIC-TCP and a ratio of $30:1$ for HS-TCP (compared with a ratio of $2:1$ for standard TCP), while $\frac{\lambda_1}{\lambda_2} = 0.1$ induces a $110:1$ ratio of flow throughputs for H-TCP and a ratio of approximately $280:1$ for HS-TCP and BIC-TCP (compared with a rato of $10:1$ for standard TCP). Thus, differences in synchronisation rate induces very substantial unfairness with HS-TCP, BIC-TCP and H-TCP. It can also be seen from Figure 2 that the level of additional unfairness over standard TCP is dependent on the path bandwidth-delay product (BDP), becoming more severe as the BDP rises .

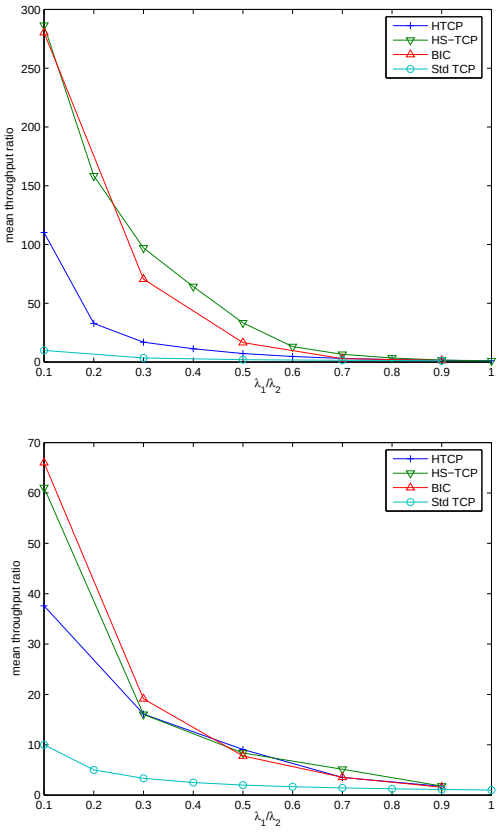

Fig. 2. Impact on throughput fairness of differences in flow synchronisation factor. Upper plot is for a bottleneck link bandwidth of 2.4Gbs and lower plot for bandwidth of 240Mbs. It can be seen that the level of unfairness induced by differences in synchronisation rate grows with the bandwidth-delay product. (150ms propagation delay,10 TCP flows).

B. Short-term Fairness

In this section we present some preliminary results to illustrate short-term unfairness issues that arise in networks in which high speed protocols are deployed. All our results are for a network of 10 flows, each operating one of standard TCP, HTCP, BIC or HS-TCP. First we plot the distribution of the $cwnd$ values at network congestion for a single flow, see Figure 3. It can be seen that while BIC-TCP and standard

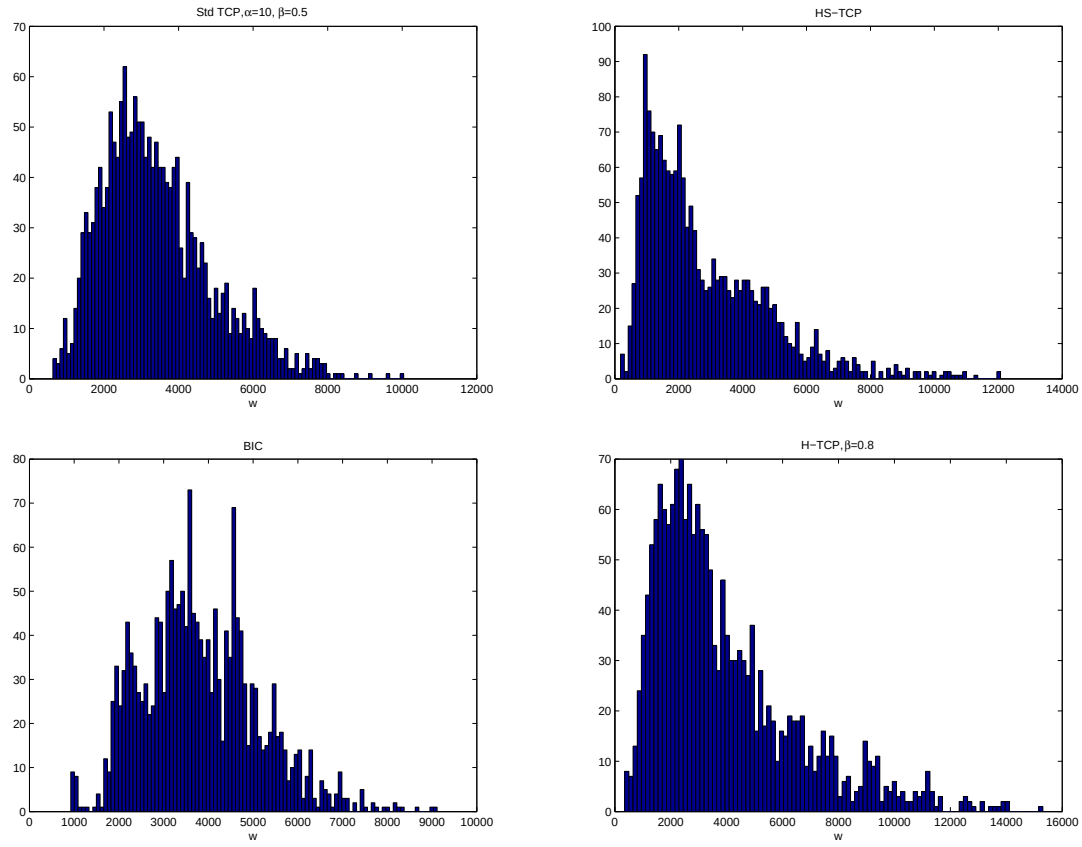

Fig. 3. Distribution of peak $cwnd$, w in packets. Results are for a flow with $\lambda = 0.25$. Network has 10 flows in all, 9 with $\lambda = 0.25$ and 1 with $\lambda = 1$ (synchronised). (150ms propagation delay, 2.4Gbs bottleneck bandwith, 10 TCP flows).

TCP have similar distributions of congestion window, the congestion window distributions for HS-TCP and H-TCP have significantly longer tails; that is, HS-TCP and H-TCP are more liable to large excursions in congestion window under unsynchronised conditions. This accords with our intuition regarding the behaviour of the more aggressive increase algorithms employed by these high-speed protocols when a flow misses a drop at a network congestion event. While such excursions have little impact on long-term fairness as they occur in all flows and their effect on throughput averages out over time (as can be seen from Figure 2, the long-term mean throughput of flows with the same synchronisation rate and round-trip time is the same), they do have an impact on short-term fairness. This can be seen from Figure 4. In this figure we use the ratio at a network congestion event of the minimum to the maximum congestion windows of the competing flows as snapshot of the local unfairness. The measured distribution of this short-term unfairness snapshot is shown for standard TCP, HTCP, BIC and HS-TCP. We can see that while standard TCP and BIC-TCP once again exhibit similar distributions, both HS-TCP and H-TCP yield a shift to the left in the distribution that corresponds to an increase in the mean short-term unfairness.

A number of important facts can be discerned from the above plots.

(i) Large variations in the rate of a given long-lived flow are often a feature of networks in which high-speed protocols are deployed. More aggressive protocols are more prone to large rate variations. In our experiments, HS-TCP is the worst offender, with BIC performing best.

(ii) For a given protocol, the distribution of rate variation depends on the network backoff factors. This is illustrated, for example, in Figure 5 for H-TCP but similar results are obtained for other high-speed protocols. Roughly speaking, the larger the backoff factors, the smaller the variation in rate and thus the less short-term unfairness. This comes at the cost, however, of increased long-term unfairness with respect to differences in synchronisation rate, see Figure 6. As noted elsewhere [6], [9], increasing the backoff factor also generally reduces network responsiveness e.g. for the startup of new flows, thereby increasing the unfairness between short and long-lived flows.

(iii) More aggressive protocols can lead to an increase in short-term unfairness.

It is important to note that the actual level of variation and short-term unfairness observed in an actual network is, of course, dependent on the degree of unsynchronisation (with variations becoming smaller as flows become more synchronised). The degree of unsynchronisation in actual networks remains only poorly understood at present and warrants further investigation. Anecdotal evidence suggests, for example, that high levels of synchronisation may in fact be common in high-

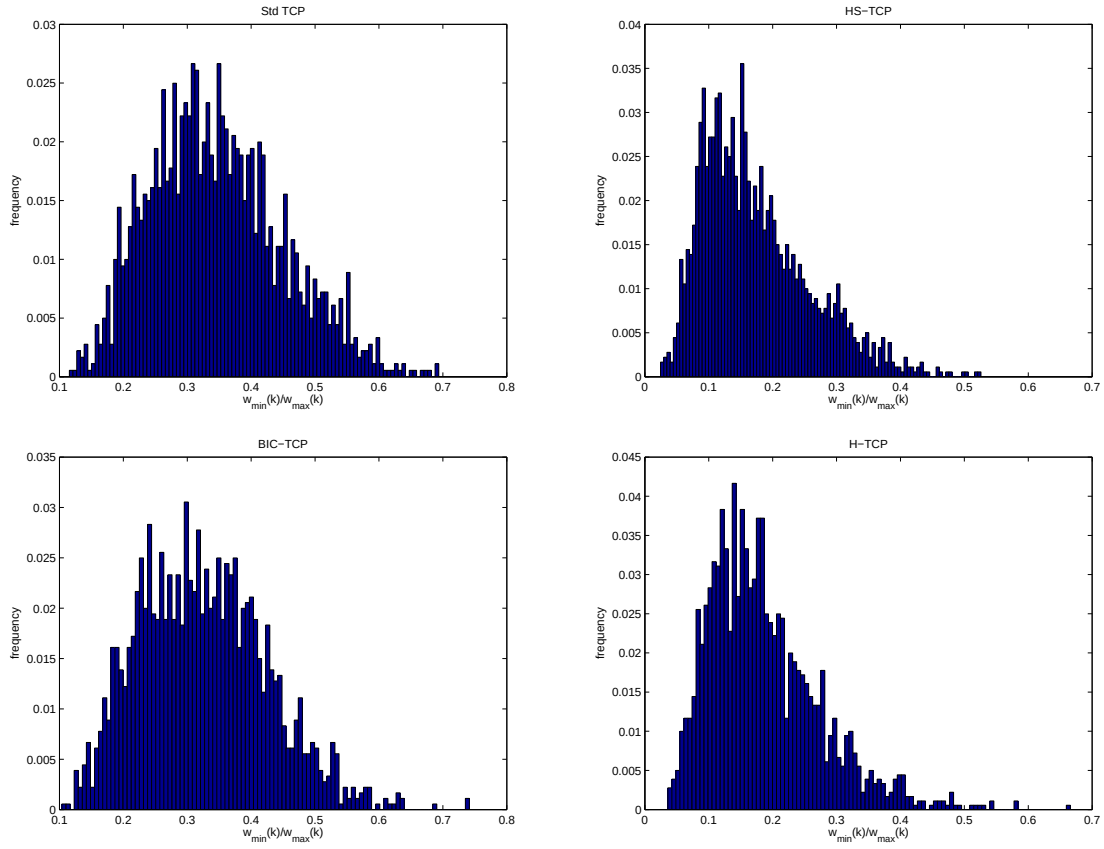

Fig. 4. Short-term fairness as measured by ratio of minimum to maximum congestion windows of competing flows at a network congestion event. (150ms propagation delay, 2.4Gbs bottleneck bandwith, 10 TCP flows, 9 with $\lambda = 0.25$ and 1 with $\lambda = 1$; results plotted are for the 9 flows with the same synchronisation rate).

speed networks and this would have a direct impact on the present discussion.

IV. Conclusions

In this paper we have studied the effect of synchronisation rate on network performance. Our findings can be summarised as follows.

- The long-term stochastic equilibria of the networks using high-speed protocols can be extremely sensitive to the flow synchronisation rates.
- Large variations in flow rate are a feature of networks in which high-speed protocols are deployed.
- Short-tem unfairness can be a feature of these network types.

Each of the above observations seem to be undesirable properties of high-speed networks and may render them unsuitable for certain classes of network traffic. A basic approach to alleviating some or all of these properties is to control the flow synchronisation rates. This can be achieved either by the design or deployment of an appropriate AQM's at the router (highspeed protocol - AQM codesign), or by introducing a dropping mechanism as a feature of the end-to-end protocols. Both of these approaches are being pursued by the authors and initial results are promising.

Finally, we note that a particularly worrying aspect of our results is that the protocol behaviours appear to be strongly dependent on the network capacity. Although, limited results are given here, we have observed that the degree of network unfairness depends crucially on the network capacity. Hence, even though the high-speed protocols that we have considered achieve their stated goal of scalable network behaviour in terms of time-between congestion consecutive congestion events, other basic properties of high-speed networks are certainly not invariant with changing network bandwidth. This is likely to become an issue as network capacities increase further.

V. Acknowledgements

This work was supported by Science Foundation Ireland grant 00/PI.1/C067.

References

[1] H. Bullot, R.L. Cottrell, R. Hughes-Jones, Evaluation of Advanced TCP Stacks on Fast Long Distance Production Networks. J.Grid Comput, 2003.

[2] R.L. Cottrell, S. Ansari, P. Khandpur, R. Gupta, R. Hughes-Jones, M. Chen, L. MacIntosh, F. Leers, Characterization and Evaluation of TCP and UDP-Based Transport On Real Networks. . Proc. 3rd Workshop on Protocols for Fast Long-distance Networks, Lyon, France, 2005.

[3] S.Floyd, K.Fall, Promoting the use of end-to-end congestion control in the internet. IEEE/ACM Transactions on Networking, August 1999

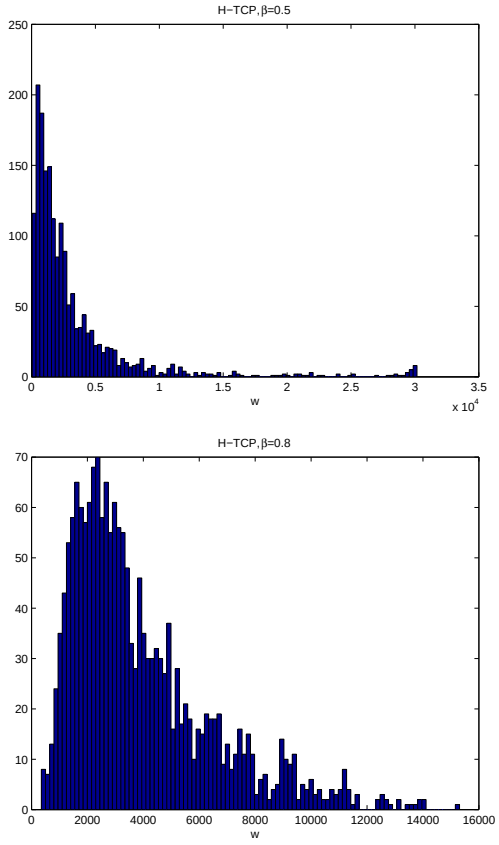

Fig. 5. Distribution of peak $cwnd$, w, for H-TCP. Upper plot is with backoff factor β 0.5, lower with β 0.8. Results are for a flow with $\lambda = 0.25$. Network has 10 flows in all, 9 with $\lambda = 0.25$ and 1 with $\lambda = 1$ (synchronised). (150ms propagation delay, 2.4Gbs bottleneck bandwith, 10 TCP flows.

[4] S.Floyd, HighSpeed TCP for Large Congestion Windows . Sally Floyd. IETF RFC 3649, Experimental, Dec 2003.

[5] T. Kelly, On engineering a stable and scalable TCP variant, Cambridge University Engineering Department Technical Report CUED/F-INFENG/TR.435, June 2002.

[6] D.J.Leith, R.N.Shorten, H-TCP Protocol for High-Speed Long-Distance Networks. Proc. 2nd Workshop on Protocols for Fast Long Distance Networks. Argonne, Canada, 2004.

[7] D.J.Leith, R.N.Shorten, H-TCP: TCP Congestion Control for High Bandwidth-Delay Product Paths. IETF Internet Draft draft-leith-tcp-htcp-00 June 2005, `http://www.hamilton.ie/net/draft-leith-tcp-htcp-00.txt`.

[8] R.N.Shorten, D.J.Leith,J.Foy, R.Kilduff, Analysis and design of congestion control in synchronised communication networks. Automatica, 2004.

[9] R.N.Shorten, F. Wirth,F., D.J. Leith, A positive systems model of TCP-like congestion control: Asymptotic results. IEEE/ACM Trans Networking, to appear.

[10] L. Xu, K. Harfoush, I. Rhee, Binary Increase Congestion Control for Fast Long-Distance Networks. Proc. INFOCOM 2004

[11] Leith, D. and R. Shorten, Small gain analysis of MIMD congestion control protocols, IEEE Communication Letters, Submitted.

[12] E. Altman and K. Avrachenkov and C. Barakat and A.A. Kherani and B.J. Prabh, Analysis of MIMD Congestion Control Algorithm for High Speed Networks, url = citeseer.ist.psu.edu/altman04analysis.html

VI. Appendix

In this Appendix we very briefly review the basic operation of HS-TCP, H-TCP and BIC-TCP. The reader is referred to the original literature for more detailed information.

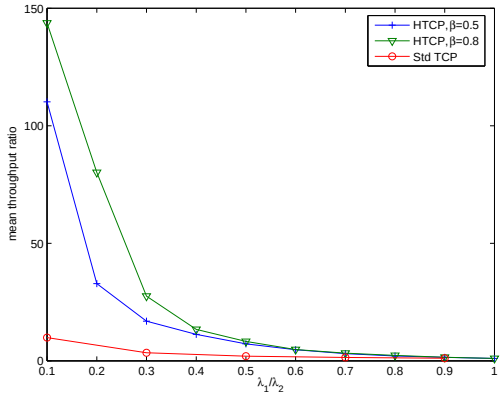

Fig. 6. Impact on throughput fairness of differences in flow synchronisation factor and backoff factor for H-TCP (150ms propagation delay, 2.4Gbs bottleneck bandwith, 10 TCP flows).

A. HS-TCP [4]

HS-TCP uses the current TCP $cwnd$ value as an indication of the bandwidth-delay product on a path. The AIMD increase and decrease parameters are then varied as functions of $cwnd$. That is, HS-TCP proposes that the TCP $cwnd$ be updated as follows

$$\begin{array}{rrl} \text{Ack:} & cwnd & \leftarrow cwnd + \dfrac{f_\alpha(cwnd)}{cwnd} \\ \text{Loss:} & cwnd & \leftarrow g_\beta(cwnd) \times cwnd \end{array}$$

In [4] logarithmic functions are proposed for $f_\alpha(cwnd)$ and $g_\beta(cwnd)$, whereby $f_\alpha(cwnd)$ increases with $cwnd$ and $g_\beta(cwnd)$ decreases. Similarly to Scalable-TCP, HS-TCP uses a mode switch so that the standard TCP update rules are used when $cwnd$ is below a specified threshold.

B. H-TCP [6]

HTCP uses the elapsed time Δ since the last congestion event, rather than $cwnd$, to indicate path bandwidth-delay product and the AIMD increase parameter is varied as a function of Δ. Optionally (but these options are not used in the present paper), the AIMD increase parameter may be scaled with path round-trip time and the AIMD decrease factor adjusted to improve link utilisation based on an estimate of the queue provisioning on a path. In more detail, the basic H-TCP algorithm [7] updates $cwnd$ as follows

$$\begin{array}{rrl} \text{Ack:} & cwnd & \leftarrow cwnd + \dfrac{f_\alpha(\Delta)}{cwnd} \\ \text{Loss:} & cwnd & \leftarrow \beta \times cwnd \end{array}$$

with

$$f_\alpha(\Delta) \quad = \quad \begin{cases} 1 & \Delta \le \Delta_L \\ \bar{f}_\alpha(\Delta) & \Delta > \Delta_L \end{cases}$$

where Δ_L is a specified threshold such that the standard TCP update algorithm is used while $\Delta \le \Delta_L$. A quadratic increase function $\bar{f}_\alpha$ is suggested in [6], [7], namely $\bar{f}_\alpha(\Delta) = 1 +$

$10(\Delta - \Delta_L) + 0.25(\Delta - \Delta_L)^2$. As in standard TCP, a backoff factor β of 0.5 is used.

C. BIC-TCP [10]

BIC-TCP employs a form of binary search algorithm to update $cwnd$. Briefly, a variable w_1 is maintained that holds a value halfway between the values of $cwnd$ just before and just after the last loss event. The $cwnd$ update rule seeks to rapidly increase $cwnd$ when it is beyond a specified limit $w_2 > w_1$, and update $cwnd$ more slowly when its value is close to w_1. Multiplicative backoff of $cwnd$ is used on detecting packet loss, with a suggested backoff factor β of 0.8.