# Next Generation TCP: Open Questions

Douglas .J. Leith, Robert N. Shorten
Hamilton Institute, Ireland

***Abstract*— While there has been significant progress in recent years in the development of TCP congestion control algorithms for high BDP paths, consensus remains lacking with regard to a number of basic issues. The aim of the present paper is to highlight some of these key bottleneck issues and present a number of new results with a view to promoting discussion and fostering progress. Issues highlighted include: impact of shape of cwnd evolution (concave, convex etc), increased variability in throughputs in unsynchronised environments when more aggressive algorithms are used, impact of proposed changes on convergence rates and network responsiveness and the associated impact on user experience.**

## I. Introduction

It is now over five years since proposals were first mooted for changes to the TCP congestion control algorithm to improve performance on high bandwidth-delay product (BDP) paths. In that time, while there has been significant progress consensus remains lacking with regard to a number of basic issues. The aim of the present paper is to highlight some of these key bottleneck issues and present a number of new results with a view to promoting discussion and fostering progress towards some degree of consensus. In particular, we focus on the following important open questions.

1) *Shape of cwnd increase function*. An active and ongoing debate continues as to the appropriate shape of cwnd increase with time. The authors of TCP Illinois [6] argue for a concave shape, Cubic TCP [10] for a mixed concave-convex shape and H-TCP [2] for a convex shape. Recently, in [1] it is argued from an analytic viewpoint that a concave-convex shape offers advantages with respect to rate variation.

2) *Responsiveness requirements*. Many of the proposed changes to the TCP congestion control algorithm are observed to significantly increase the time for a network of flows to converge following a disturbance (such as the startup of new flows), e.g. see [8], [11], [5]. This issue is related not only to the scaling of congestion epoch duration with path BDP, but also to the new network dynamics created by proposed changes. Convergence behaviour is dependent on network conditions and the prevalence of such behaviour is "typical" network conditions is unclear at present. Moreover, the relationship between responsiveness and user quality of service has not been well explored in the present context but nevertheless seems to be of key importance for clarifying whether or not rapid convergence is an important design driver.

This work was supported by by Science Foundation Ireland grants 07/IN.1/I901 and 04/IN3/I460.

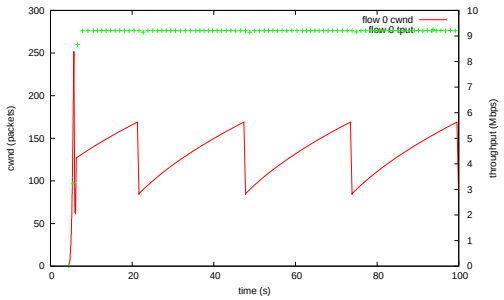



Fig. 1. Throughput and cwnd histories for Reno when buffer is sized at 1×BDP. 10Mbps link, 100ms RTT.

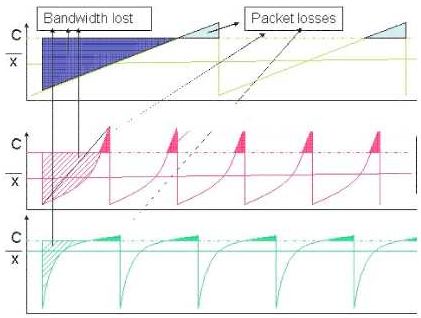



Fig. 2. Illustrating common argument confusing cwnd and throughput[1].

## II. Shape of cwnd increase function

### A. Impact of buffering

We begin by making the straightforward but important observation that when buffering is present (as is always the case on network links), flow throughput and flow congestion window are fundamentally different quantities that are generally only weakly related. This is illustrated for example in Figure 1 which plots the measured cwnd and throughput time histories for a Reno flow on a link with a BDP worth of buffering (the experimental setup used for these measurements and others in this paper is detailed in the Appendix). It can be seen that, due to the link buffering, the throughput remains constant while the cwnd evolves according to the usual cwnd pattern. This has immediate implications for discussions regarding the shape of cwnd increase. For example, in Figure 1 the cwnd increase can be changed to be convex, concave etc without impacting flow throughput and this directly challenges recent arguments such as that illustrated in Figure 2 and used to motivate a concave cwnd increase.

Of course the amount of buffering affects the relationship between throughput and cwnd. However, this is more related

to the choice of cwnd backoff factor than to the cwnd increase shape. By adjusting the backoff factor we can ensure that a network buffer just empties at backoff, thereby decoupling throughput and cwnd shape similarly to Figure 1. This can be achieved for any reasonable level of buffering (i.e. other than such small buffers that micro-scale burst effects dominate behaviour) [9]. Indeed, an adaptive backoff scheme based on this simple observation has already been proposed elsewhere and demonstrated experimentally, see [9]. When flow backoffs are fixed, statistical multiplexing of flow backoffs can lead to a similar effect whereby high throughput is maintained with small buffers.

When *multiple* flows share a link the cwnd increase shape can affect how bandwidth is shared, and this is discussed in more detail in the next section.

### B. Rate of variation

A common feature of loss-based high-speed protocols is their aggressive additive increase phase. However, a consequence of this action is that when drops are unsynchronised flows are able to rapidly grab additional bandwidth when some flows observe a loss event that others miss. This raises concerns about the magnitude of fluctuations in the throughput achieved by flows, and of the time-scale over which such fluctuations occur. This issue has been previously discussed by a number of authors, e.g. see [3], [1], [5] and references therein, but remains controversial.

Intuitively, on a high BDP path *any* responsive loss-based algorithm must increase its cwnd aggressively following backoff after a loss. Otherwise the congestion epoch duration will be unduly long and responsiveness will suffer. This is therefore a primary design driver for all loss-based algorithms.

To illustrate this point, the increase functions used by Cubic TCP and H-TCP are shown in Figure 3. Attention is drawn to the fact that while Cubic uses a concave-convex shape and H-TCP a convex shape, overall the increase functions are very similar. Based on the foregoing discussion, this is unsurprising since both need to be aggressive in order to reduce the congestion epoch duration in high BDP paths.

This insight suggests that the *shape* of increase is perhaps only a secondary factor when considering rate of variation in cwnd. Also, it highlights that when making comparisons to evaluate the impact of different shapes it is vitally important to control for the aggressiveness of the increase function used as otherwise we may well be comparing the impact of features other then increase shape.

To explore this issue further, we consider two increase functions defined by:

$$cwnd(k+1) = cwnd(k) + cT^3$$

and

$$cwnd(k+1) = w + c(T - \frac{1-\beta}{c}w)^{1/3})^3$$

where $w$ denotes the cwnd value after the last backoff, $T$ is the elapsed time since the last backoff, $c$ is a design parameter and $\beta$ is the backoff factor. In steady state, for both increase functions the peak cwnd before backoff is $c/(1-\beta)T_w^3$ where $T_w$ is the congestion epoch duration. Thus, the increase functions are identically aggressive, but the first is convex while the second is concave-convex. We also use the same backoff factor $\beta = 0.5$ for both to control for its impact on behaviour (the impact of differences in backoff factor is discussed later).

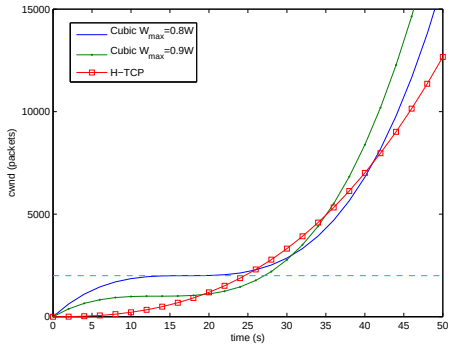
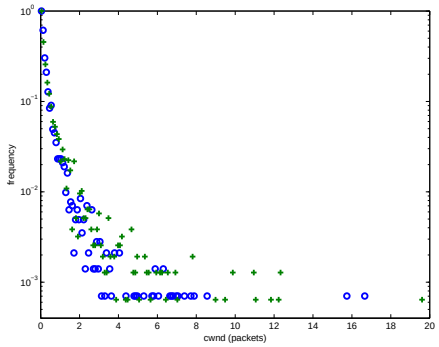



Fig. 3. Cubic TCP and H-TCP increase functions for the situation where the $cwnd$ at last backoff is 10000 packets. The y-axis is normalised so that the origin lies at the $cwnd$ immediately after backoff, the dashed line then marks the normalised $cwnd$ at last backoff. Two lines are shown for Cubic since in the Linux implementation its response is dependent on whether the last backoff occurred at a cwnd value before or after the inflection point in the increase function – see [5]. The H-TCP increase function shown is for the 0.8 backoff factor used by Cubic TCP in order to facilitate direct comparison.

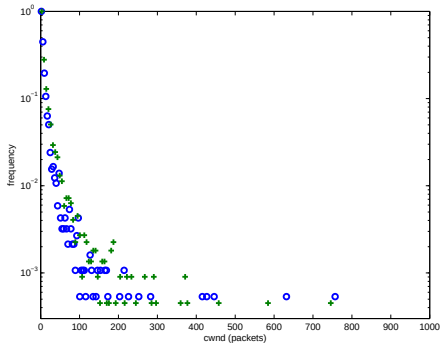
To control for possible differences in synchronisation rate induced by differences in increase function, we generate packet losses randomly, with a geometric distribution based on elapsed time since the last backoff. Figure 4 plots the measured distributions of cwnd at backoff for both increase functions for a range of loss rates $p$. It can be seen that the distributions are remarkably similar across a wide range of loss conditions, providing a degree of support for the conjecture that it is the aggressiveness of the increase function that is of primary importance and shape is only secondary, at least from the point of view of cwnd fluctuations in unsynchronised condtions.

We note that while the shape of increase has a relatively minor impact on the cwnd distribution, in contrast the choice of backoff factor $\beta$ can have a considerable impact. Figure 5 plots cwnd distributions for $\beta = 0.25$ and $\beta = 0.8$ (while adjusting $c = 1 - \beta$ to maintain the same level of aggressiveness). It can be seen that for the larger value of $\beta$ the variability in cwnd is much less. This occurs because with the larger backoff factor the rate of increase of cwnd must be decreased if we are to maintain the same level of aggressiveness. Note that this has important implications when evaluating proposed changes to TCP congestion control since it is common to adjust the backoff factor to be larger than 0.5 and we can therefore expect that this alone can have a significant impact on the observed magnitude of variations in cwnd. Unless this is taken into account, there is a risk of incorrectly attributing differences in cwnd variability to factors such as the shape of increase function.

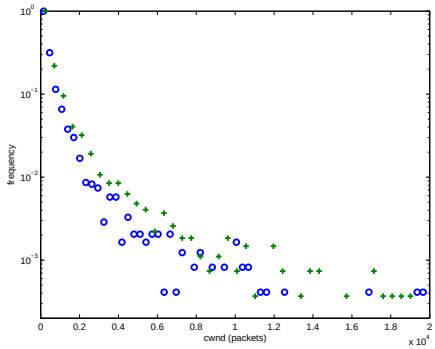
[1]Figure is taken from `http://www.ews.uiuc.edu/ shaoliu/tcpillinois/background2.html`




(a) $p = 0.2$

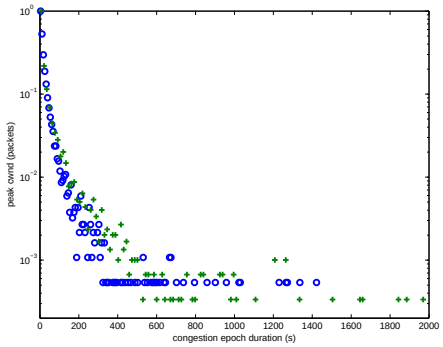



(b) $p = 0.05$

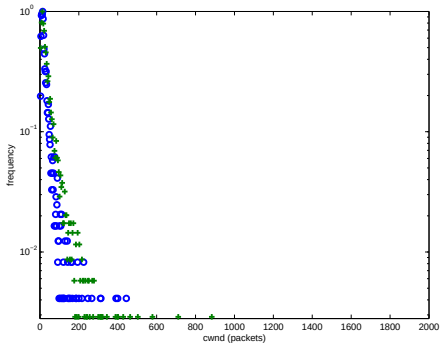



(c) $p = 0.02$

Fig. 4. Distribution of cwnd at backoff for convex and concave-convex updates vs loss probability $p$. Key: $+$ convex increase, $o$ concave-convex increase. $c = 0.5, \beta = 0.5$




(a) $\beta = 0.25$




(b) $\beta = 0.8$

Fig. 5. Distribution of cwnd at backoff for convex and concave updates vs loss probability. Loss model based on time between congestion events. Key: $+$ convex increase, $o$ concave-convex increase. $c = 1 - \beta, p = 0.05$

The foregoing results are for a clean setup that seeks to control for factors unrelated to the shape of cwnd increase in order to provide insight. Factors such as the impact of shape of increase function on synchronisation rate may well be important but are beyond the scope of the present paper. Nevertheless, experimental measurements suggest that the insight provided from the foregoing simple setup is indeed relevant to more complex network conditions. For example, Figure 6 presents measurements for the Cubic TCP and H-TCP algorithms. To control for the differences in backoff factor used in Cubic and the standard H-TCP algorithm, measurements are taken using a backoff factor of 0.8 for H-TCP (but without any other change to the algorithm). It can be seen that the $cwnd$ distributions are extremely similar, as might be expected from the foregoing discussion. These initial results suggest that reported differences between the coefficient of variation of the $cwnd$ distributions of Cubic TCP and H-TCP may well be mainly associated with differences in the backoff factors used by the algorithms, rather than to the increase functions.

## III. Convergence rate

A great many of the proposed changes to the TCP congestion control algorithm have been observed to sometimes induce a significant increase in the time for a network of flows to converge following a disturbance (such as the startup of new flows). While first highlighted for High-Speed TCP [8], this behaviour has since also been noted for BIC [11], Cubic [5] and more recently TCP Illinois and Compound TCP [4]. See for example Figure 7. This issue is related not only to the scaling of congestion epoch duration with path BDP, but also to the new network dynamics created by proposed changes. For example, algorithms such as High-Speed TCP, BIC and Cubic create asymmetry within the network whereby flows with small cwnd are less aggressive than those with large cwnd. As a result, newly started flows can be at a disadvantage to established flows and so take a considerable time to gain their fair bandwidth share. In addition, many proposals make use of a larger value of backoff factor and this is also known to

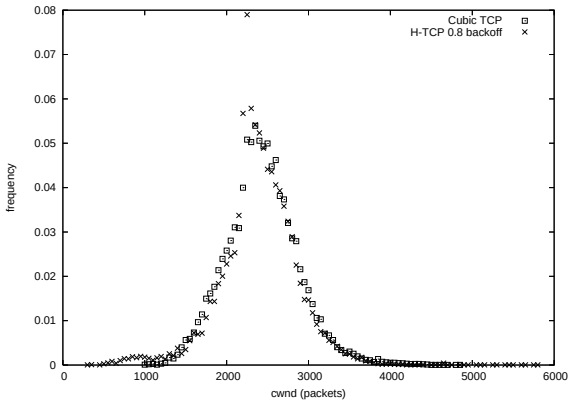



Fig. 6. Experimentally measured distribution of $cwnd$. Measurements are shown for both Cubic-TCP and H-TCP using a backoff factor of 0.8. Three long-lived flows and 25 background sessions. Bandwidth is 250 Mbit/sec, RTT 200ms, queue size 100% BDP.

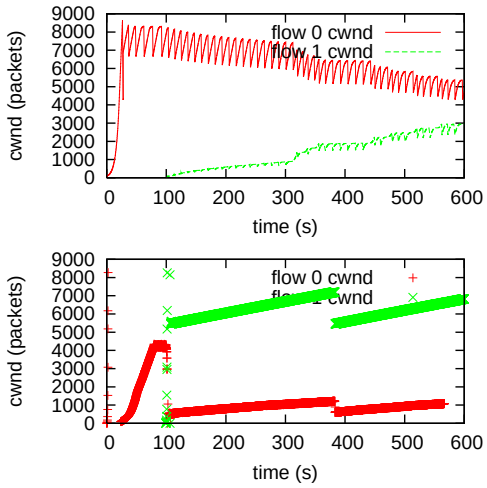



Fig. 7. $cwnd$ time histories following startup of a second flow. Cubic TCP (top), Compound TCP (bottom). 250Mbit/sec, 200ms RTT, 1×BDP queue.

decrease the responsiveness of the network since established flows then release bandwidth less quickly (reinforcing the impact of the asymmetry in increase rates noted previously).

Since so many proposals exhibit slow convergence, it appears that this is not currently a primary design driver. In part this is perhaps due to lack of clarity as to whether slow convergence is indeed a feature of operation in realistic network environments (the measurements in Figure 7 are taken in a specific scenario, albeit including background web traffic). There are also few studies at present exploring the impact, if any, of convergence rate on the quality of service experienced by users.

To begin to explore these issues further, we propose that measurements are needed of completion times vs choice of TCP congestion control algorithm for a range of network conditions in order to gain insight into the applications where use of high-speed algorithms is likely to be most beneficial. We argue that completion time is a useful metric of user quality of service for TCP applications since it is directly related to the perceived responsiveness of client-server applications etc. Our initial tests indicate that http traffic is dominated by slow-start performance and is relatively insensitive to the choice of congestion control algorithm (due to the small connection sizes relative to path BDP on high-speed links). Our measurements indicate that video streaming traffic (youtube etc) is typically rate limited at the server side and again is insensitive to the choice of congestion control algorithm. It is already known that FTP with large file sizes does exercise the congestion control algorithm. We have also found that rsync is similarly sensitive – this is particularly interesting as rsync is widely used for mirroring and backup of very large datasets and so represents an important application. Our tests indicate that FTP, rsync and iperf (with specified connection size) all exhibit similar behaviour at the network speeds studied (which are too low for disk access bandwidth to be a constraint), providing valuable validation of the use of iperf for TCP evaluation.

Building on this exploratory applications testing, we have carried out a number of detailed measurement studies on production paths. We selected production paths rather than lab testing in order to explore the sensitivity of completion time to choice of congestion control algorithm in a genuinely realistic context. Importantly, use of real paths allows us to directly address a key difficulty with current lab testing: namely, it remains both unclear and controversial as to what combinations of network conditions are “realistic”. Our testing is of course confined to specific paths and lacking a wider-scale measurement study it is not possible to argue strongly as to the representative nature of these paths. Nevertheless, we do emphasise that these are production paths carrying live traffic and so they are undoubtedly “realistic” (indeed, they are real rather than merely realistic) and so may help to move the discussion within the community forward.

We begin by presenting measurements taken on the path between the Hamilton Institute in Dublin, Ireland and WAN-in-Lab at Caltech in California. The bottleneck link on this path appears to be the 100Mbps hop at the Hamilton Institute. This hop lies between the main gateway at the institute and the main National University of Ireland Maynooth 1Gbps gateway and carries all of the production traffic from the institute. Tests were carried out during office hours and measurements were repeated 20 times over a period of one working day to try to capture representative conditions. In order to explore the impact of convergence rate we carried out tests using both the default Linux 2.6.23 Cubic TCP algorithm (which is known to exhibit slow convergence) and the Linux 2.6.23 H-TCP algorithm (which exhibits somewhat faster convergence). Test runs for each algorithm were interleaved to mitigate any bias due to time of day or fluctuations in network conditions. It is important to emphasise that our aim here is not to compare the Cubic and H-TCP algorithms per se, which are just used as examples, but rather to explore the impact of convergence rate on flow completion time and fairness with a view to encouraging discussion as to whether convergence rate should be a design driver.

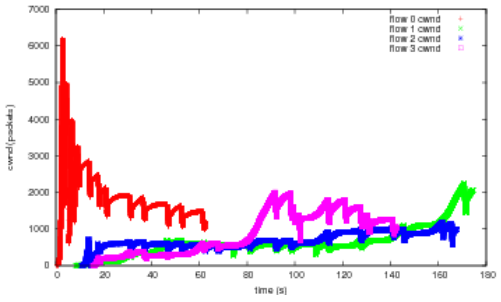
Figure 8 presents typical cwnd time histories measured over this path. In this test four 200MB sized iperf transfers are started at intervals 5s apart. It can be seen that with Cubic the first flow grabs almost all of the available bandwidth and since it is slow to release this to subsequent flows it gains a

significant advantge, even though the start times of the various flows are only 5s apart. As a result, completion time of the first flow is significantly lower (by around a factor of two) than for later flows. This behaviour is consistently observed – data from 20 tests is shown in Figure 9 and it can be seen that with Cubic the first flow on average obtains about a factor of two lower completion time than flows started slightly later. Also shown are the corresponding measurements with the H-TCP algorithm. It can be seen that the flow completion times are fairer, with the first flow gaining no significant advantage over later flows.

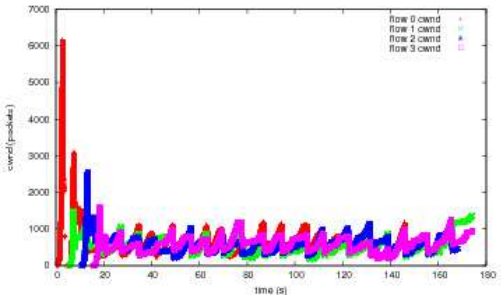
It can also be seen that the overall flow completion times are comparable to those with Cubic, which might suggest that the shorter completion time of the first flow with Cubic carries little actual cost for later flows. However, this is really an artefact of the (unrealistic) fact that the flow connection sizes are all identical in this experiment. To explore the impact of differences in connection size, we carried out a second set of tests where we measure the completion time vs connection size for a flow started 5s after a longer-lived flow. Measured results are shown in Figure 10(a) for both Cubic and H-TCP. It can be seen that the mean completion time with Cubic is approximately double that with H-TCP. This is a direct result of the slow convergence of a network of Cubic flows, which means that incumbent flows can be slow to release bandwidth to newly started flows.

We note that this sort of behaviour is not confined to higher speed links, although it is more pronounced on higher BDP paths. Figure 10(b) plots measurements taken on a domestic DSL link in Dublin, Ireland. This has a download link speed of 3Mbps and an upload speed of 256Kbps. Measurements are taken for downloads from the Hamilton Institute to a machine located behind the DSL link. Figure 10(b) plots measured completion time vs connection size for a flow started 5s after a longer-lived flow. Again, it can be seen that the mean completion time is consistently higher with Cubic TCP.

Although only a first step, with tests on a wider range of links required, we nevertheless argue that this sort of test provides a useful connection between convergence rate and the completion times experienced by users over a real link. It thus has the potential to progress the current discussion within the community regarding the importance, or otherwise, of convergence rate.




(a) Cubic




(b) H-TCP

Fig. 8. Example cwnd time histories for four downloads between Dublin, Ireland and Caltech, California. Downloads started 5s apart, Linux 2.6.23.

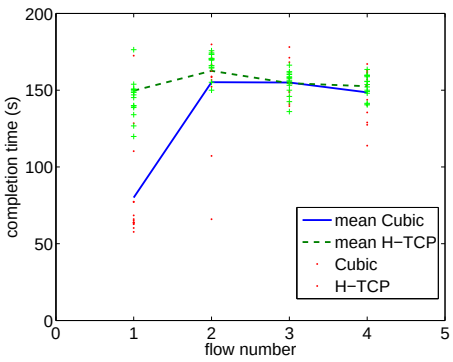


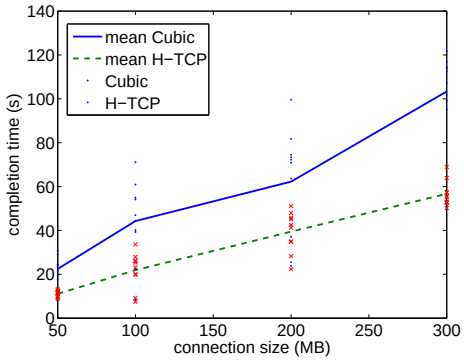

Fig. 9. Completion times for four downloads between Dublin, Ireland and Caltech, California. Lines plot mean completion times, markers indicate values from individual test runs. Downloads started 5s apart, Linux 2.6.23.

## IV. Other considerations

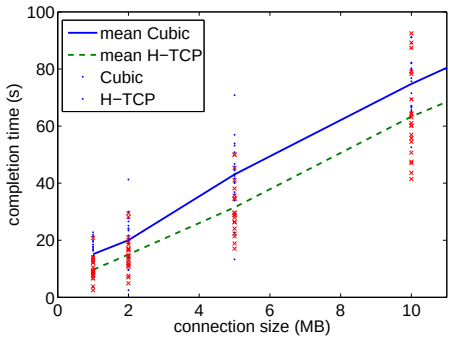
Finally, we mention briefly two considerations that are potentially of great practical importance.

1) *Scalability*. A feature of standard TCP (in congestion avoidance mode) is that many of its properties are independent of network capacity (the notable exception being the time elapsed between congestion events). For example, roughly speaking, for standard TCP, the average bandwidth division between competing flows depends only on the RTT distribution of the flows, and the proportion of congestion notifications experienced by each flow. Similarly, the rate of convergence rate, when measured in terms of congestion epoch, is also independent of the network capacity. These invariance properties stem directly from the fact that the additive increase is linear and consequently that the time between consecutive congestion events does not play a role in the equations governing the evolution of the network [7]. It is shown in [3] hat the situation is different when the growth functions are nonlinear. In this case, the aforemention properties depend on the available capacity and so exhibit fundamentally different scaling behaviour to standard TCP.

2) *Network dynamics*. The interactions when flows compete for available bandwidth defines a complex stochastic dynamical system, which ultimately governs the behaviour of the network. All of the proposed high-speed congestion control algorithms are highly nonlinear and, generally speaking, difficult to analyse. It is therefore often the case that simulations are




(a) Caltech WAN-in-lab, 100Mbps bottleneck




(b) Domestic DSL link located in Dublin, 3Mbps downlink, 256Kb uplink

Fig. 10. Completion time vs connection size. Lines plot mean completion times, markers indicate values from individual test runs. Downloads from Hamilton Institute to various destinations. Linux 2.6.23.

one of our most powerful analysis tools. It is therefore essential that we can trust our simulations. For example, it is very important to believe that our average network properties do not depend on the initial conditions of the network simulator, or in the order in which flows start-up. Mathematically speaking, this issue concerns whether or not the network dynamics are ergodic (average quantities derived from many simulations converge to same values irrespective of initial conditions). Generally, proving ergodicity is difficult – except for the case of standard TCP [3] – but we argue this issue is nevertheless one of the most pressing mathematical questions in the study of network congestion control.

## V. Acknowledgements

The assistance of Lachlan Andrew and WAN-in-Lab at Caltech in making available a machine to facilitate our tests over production links is gratefully acknowledged. The assistance of David Malone in providing access to the DSL link in Dublin is also gratefully ackowledged.

## VI. Appendix: Hardware and Software

WAN tests are conducted on a range of links connecting sites within Europe and sites in the US and Europe. Lab tests are conducted on an experimental testbed. Commodity high-end PCs were connected to gigabit switches to form the branches of a dumbbell topology. All sender and receiver machines used in the tests have identical hardware and software configurations as shown in Table I and are connected to the switches at 1Gb/sec. The router, running the FreeBSD dummynet software, can be configured with various bottleneck queue-sizes, capacities and round trip propagation delays to emulate a wide range network conditions.

| | Description |
|---|---|
| CPU | Intel Xeon CPU 3.00GHz 1066 FSB |
| Memory | 1 Gbyte |
| Motherboard | Dell PowerEdge PE860 |
| txqueuelen | 1000 |
| max_backlog | 2500 |
| NIC | Intel Pro 1000PT PCIe x4 |
| NIC Driver | e1000 5.2.52-k4 |
| TX & RX Descriptors | 4096 |

TABLE I
Hardware and Software Configuration.

Apart from the router, all machines run an instrumented version of the Linux 2.6.23 kernel. It is known that the at high bandwidth-delay products SACK processing etc in the Linux network stack can impose a sufficiently high burden on end hosts that it leads to a significant performance degradation. We performed tests to confirm, on our hardware, appropriate network stack operation over the range of network conditions tested. The kernel is instrumented using custom tcp-probe monitoring to allow measurement of TCP variables.

## References

[1] H. Cai, D. Y. Eun, S. Ha, and I. R. andLisong Xu. Stochastic ordering for internet congestion control. In *Proc. Workshop on Protocols for Fast Long Distance Networks.*, 2007.

[2] D.J.Leith and R.N.Shorten. H-TCP protocol for high-speed long-distance networks. In *Proc. 2nd Workshop on Protocols for Fast Long Distance Networks. Argonne, Canada, 2004*, 2004.

[3] D.J.leith and R.N.Shorten. Impact of drop synchronisation on TCP fairness in high bandwidth-delay product networks. In *Proc. Workshop on Protocols for Fast Long Distance Networks, Nara, Japan.*, 2006.

[4] D. Leith, L. L. H. Andrew, T. Quetchenbach, and R. N. Shorten. Experimental evaluation of delay/loss-based TCP congestion control algorithms. In *Proc. Workshop on Protocols for Fast Long Distance Networks.*, 2008.

[5] D. Leith, R. Shorten, and G. McCullagh. Experimental evaluation of Cubic-TCP. In *Proc. Workshop on Protocols for Fast Long Distance Networks.*, 2007.

[6] S. Liu, T. Basar, and R. Srikant. TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks. In *Proc. First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), Pisa, Italy, October 11-13, 2006*, 2006.

[7] R.N.Shorten, D.J.Leith, and F.Wirth. Products of random matrices and the internet: Asymptotic results. *IEEE Transactions on Networking, 14(6), pp. 616-629*, 2006.

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

[9] R. Shorten and D. Leith. On queue provisioning, network efficiency and the delay-bandwidth product. IEEE Transactions on Networking, 2007.

[10] L. Xu and I. Rhee. CUBIC: A new TCP-Friendly high-speed TCP variant. In *Proc. Workshop on Protocols for Fast Long Distance Networks, 2005*, 2005.

[11] Y.Lee, D. Leith, and R.N.Shorten. Experimental evaluation of TCP protocols for high-speed networks. IEEE Transactions on Networking, June 2007.