Feedback, Transport Layer Protocols and Buffer Sizing

Shankar Raman*, Shashank Jain* and Gaurav Raina†
India-UK Advanced Technology Centre of Excellence in Next Generation Networks
**Department of Computer Science and Engineering,* †*Department of Electrical Engineering*
Indian Institute of Technology Madras, Chennai - 600 036, India
Email: mjsraman@cse.iitm.ac.in, shashank@cse.iitm.ac.in, gaurav@ee.iitm.ac.in

Abstract*—A key aspect of network performance is coupled with the design of transport layer protocols, the choice of feedback from queues, and by the buffer sizing requirements at routers. In this paper, we consider some transport protocols which use different feedback mechanisms to manage their flow and congestion control. We study the performance of these protocols under the influence of different buffer sizes. The transport protocols considered include CUBIC TCP, Compound TCP and an illustrative protocol that could utilize Explicit Congestion Notification (ECN) marks. CUBIC TCP, which is the current default implementation in Linux, uses *packet loss* as the primary feedback signal. Compound TCP, which is the current default implementation in the Windows platform, uses both *packet loss and queuing delay*. In the aforementioned transport protocols, using NS-2 simulations and some analysis, we exhibit that irrespective of the feedback signal used, buffer sizes play a very important role in network performance. In particular, we highlight that even minor variations in buffer size can readily lead to the emergence of limit cycles. These limit cycles tend to destabilize the queue dynamics, induce deterministic oscillations in the packet losses and can degrade link utilization. Using a combination of currently deployed protocols and an illustrative protocol, our work serves to exhibit the importance for a combined study of transport protocols, different feedback mechanisms and sizing router buffers.

***Keywords*-Feedback; Transport protocols; Buffer sizing.**

I. INTRODUCTION

Transport protocols play an integral part in delivering end-to-end quality of service. However, the design of transport protocols is affected by the choice of feedback mechanisms from the queue. Router buffers, which traditionally have been used to smooth statistical fluctuations in the demand for transmission capacity, also play a key role in providing end-to-end performance. In this paper, we highlight the inter-related nature of transport protocols, feedback and buffers.

A. Buffer sizing

Buffers in routers are a key architectural component of the Internet, and have played an important role in store-and-forward communication networks. Despite their importance, they can also have a detrimental effect by introducing queuing delay and jitter. In the Internet, buffers are currently sized using a rule of thumb which says that each link needs a buffer of size $B = C * \overline{RTT}$, where C is the data rate, and $\overline{RTT}$ is the average round-trip time of the flows passing across the link which is currently taken to be 250 ms [19]. For example, a 10 Gbps router line card needs approximately 10 Gbps * 250 ms = 2.5 Gbits of buffers, which is enough to hold roughly 200k packets. This rule of thumb is clearly not scalable with the growth of transmission capacity. Additionally, such large buffers also have a significant influence on the energy consumption of routers [16].

B. Transport protocols and small buffers

A body of work is emerging that takes a rather radical approach to the issue of buffer sizing: it suggests that it *might* be possible to have buffer sizes of the order of tens of packets. This small buffer sizing rule does not depend on C or $\overline{RTT}$. For work on the development of scaling regimes for queuing delay see [5], for work on TCP see [11], [12], and for a more recent overview see [16]. A key conclusion of [12] is that small buffers have a stabilizing effect on the end-to-end dynamics of Additive Increase Multiplicative Decrease (AIMD) TCP flow control. In essence, in a large bandwidth-delay product environment with small drop-tail buffers, anything larger than a few dozen packets may lead to synchronization effects. Synchronization, in this context, is synonymous with (stable) limit cycles; for definitions and an exposition of the requisite theory, see [10]. The aforementioned analysis was, however, limited to the standard AIMD TCP.

C. Feedback, transport protocols and small buffers

One way to classify transport protocols is via the feedback signals that they use to manage flow and congestion control. The feedback signals that the end-systems may use are queuing delay, packet loss, explicit congestion notification (ECN) marks, or rates. ECN marks are intended to be used in conjunction with transport protocols, and there are numerous proposals for queue management strategies on how to mark packets; for example RED [2].

Recent work has begun to focus on the aspect of sizing router buffers under the influence of different forms of feedback and queue management strategies. For example, the study of rate based feedback with different notions of fairness among the flows was analyzed in [6], [17]. For a study of some queue management schemes, with small

ISBN: 978-1-61208-183-0

buffers, see [9] and for the impact of delay based transport layer protocols (like FAST TCP [18]), see [13]. Additionally, some recent work has also been carried out on a mixture of real time traffic (open-loop) and TCP traffic (closed-loop) with respect to the issue of sizing router buffers [15].

Today, there is no consensus on the desired feedback mechanism, the transport protocol variant for a given feedback mechanism, or on the optimal rule for next-generation router buffer size. Our work exhibits a relationship between feedback, transport protocols, router buffer sizing and network performance. Using a combination of simulations and some theory, we show how the choice of router buffer size may affect performance irrespective of the feedback used; incorrect buffer sizes may induce the onset of limit cycles.

The rest of the paper is organized as follows. In Section II, we briefly describe CUBIC TCP, Compound TCP and also consider a model for an ECN based transport protocol. In Section III, we conduct simulations with CUBIC and Compound TCP in the Network Simulator (NS-2) [20]. In Section IV, we analyze the ECN based protocol with different resource design functions. In Section V, we present our conclusions and some discussions.

II. SOME TRANSPORT PROTOCOLS

In this section, we describe the variants of transport protocols that have been implemented in Linux and the Windows platforms. CUBIC TCP [3] is currently deployed in Linux and uses loss as the feedback signal. Compound TCP [14] is the current default implementation in the Windows platform, and uses both delay and loss for congestion control. In addition to these TCPs, we also consider a theoretical model of a transport protocol that may use ECN marks.

A. CUBIC TCP

In CUBIC TCP, upon detecting the loss of a packet, the congestion window is reduced by a *multiplicative factor* β, where β is a constant multiplication decrease factor. The window size prior to the reduction is set to W_{max} and the current window is increased using the following cubic window growth function

$$W(t) = C_s(t-K)^3 + W_{max}, \tag{1}$$

where C_s is a parameter called a *scaling factor*, t is the elapsed time since the last window reduction, and K is the time period the above function takes to increase from W to W_{max} when no loss is detected. The functional form of K is given by

$$K = \sqrt[3]{W_{max}\beta/C_s}\cdot \tag{2}$$

If standard TCPs like TCP Reno increase their window size by α per RTT, then the window size of CUBIC in terms of elapsed time is given by

$$W_{tcp(t)} = W_{max}(1-\beta) + \frac{3\beta}{2-\beta}\frac{t}{RTT}, \tag{3}$$

where

$$\alpha = 3\beta/(2-\beta)\cdot \tag{4}$$

Depending on the value of the current window size *(cwnd)*, CUBIC operates in the following three different regimes:

$$cwnd = \begin{cases} W_{tcp(t)} & cwnd < W_{tcp(t)} \\ cwnd + \frac{W(t+RTT)-cwnd}{cwnd} & cwnd < W_{max} \\ \text{probe for new } W_{max} & cwnd > W_{max}\cdot \end{cases} \tag{5}$$

The increased growth rate helps to achieve scalability, whereas the fairness and stability is maintained by forcing an almost linear growth when the window size is far from W_{max}. For further details on the protocol design see [3].

B. Compound TCP

Compound TCP is a loss-based congestion control algorithm with a scalable delay-based component [14]. This additional delay-based component, derived from TCP Vegas, serves for better efficiency, RTT fairness and TCP friendliness. The delay-based component is effective only in the congestion avoidance phase where the sender side congestion window is determined by

$$win = min(cwnd + dwnd, awnd), \tag{6}$$

where $cwnd$ is the normal loss-based component, $dwnd$ (delay window) controls the delay-based component and $awnd$ is the advertised window from the receiver. Compound TCP also maintains the number of backlogged packets in the queue, *Diff*, for every connection.

$$\mathit{Diff} = (Expected - Actual) * BaseRTT, \tag{7}$$

where $Expected = WindowSize/BaseRTT$ and $Actual = WindowSize/RTT$. The delay-based component gracefully reduces its window if $\mathit{diff} > \gamma$ (the threshold value), i.e. we need at least γ packets in the system to detect an early congestion. The changes in the window size for Compound TCP can be summarized as

$$dwnd(t+1) = \begin{cases} dwnd(t) + (\alpha.win(t)^k - 1) & \mathit{diff} < \gamma \\ dwnd(t) - \zeta.\mathit{diff} & \mathit{diff} \geq \gamma \\ win(t).(1-\beta) - cwnd/2 & \text{loss} \end{cases} \tag{8}$$

where α, β and k are tunable parameters. ζ is a parameter which determines how rapidly the window size should be reduced when early congestion is detected.

C. An ECN-based transport protocol

We outline an illustrative transport protocol, mentioned in [5], and analyse it with different resource design choices. Consider a network with a set J of *resources*. Let a route r be a non-empty subset of J, and write R for the set of possible routes and suppose that route r carries a flow of rate x_r, for each $r \in R$. Consider the following equations

$$\frac{dx_r(t)}{dt} = k_r \Big[w_r - x_r(t) \sum_{j \in r} \mu_j(t) \Big] \tag{9}$$

for $r \in R$, where k_r is the gain factor and

$$\mu_j(t) = p_j \left(\sum_{s : j \in s} x_s(t) \right) \tag{10}$$

for $j \in J$, and where the weights w_r determine the share of the scare resources obtained by the different flows. We can interpret equations (9) and (10) as follows. Suppose that resource j *marks* a proportion $p_j(y)$ of packets with a feedback signal when the total flow through resource j is y. Thus equation (9) corresponds to a rate control algorithm for user r that comprises two components: a steady *increase* at a rate proportional to w_r, and a steady *decrease* at rate proportional to the stream of congestion indication signals received. We now consider two functional forms for the resource. Suppose that

$$p_j(y) = (y/C_j)^{B_j} \tag{11}$$

This form arises if the resource j were to be modelled as a $M/M/1$ queue, with a service rate of C_j packets per unit time, at which a packet is marked with a congestion indication signal if it arrives at the queue to find at least B_j packets already present. This functional form has also been proposed to represent small buffer drop-tail networks while modeling long-lived TCP flows; see [11], [12]. Another simple functional form for the resource could be

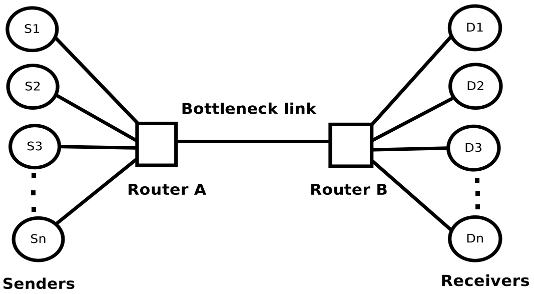
$$p_j(y) = [y - C_j]^+ / y, \tag{12}$$

where $p(.)$ is the proportion of packets overflowing a large buffer. This functional form has been devised to represent drop-tail networks; see [11], [12] and references therein.

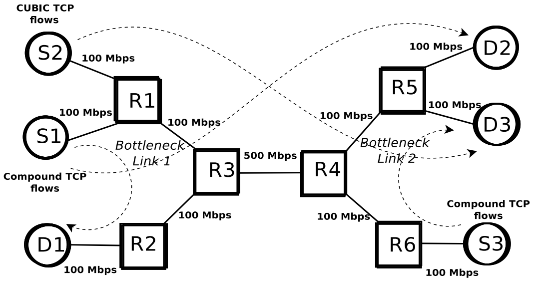
III. SIMULATIONS FOR CUBIC AND COMPOUND TCP

Given that both CUBIC and Compound TCPs are implemented today, it is appropriate to perform simulations when both these protocols are present in the network. We highlight some representative simulations over a single and multi-bottleneck topology where we vary buffer sizes, round-trip times and also the traffic mix.

In a previous evaluation, the slow convergence rate of CUBIC TCP was noted [7]. A comparison of CUBIC with Compound TCP [1], [8] has revealed that CUBIC TCP has a propensity to be very aggressive, which readily translates into unfairness towards competing Compound TCP flows. However, neither CUBIC nor Compound TCP have undergone evaluation with respect to the issue of buffer sizing prior to their implementation in Linux and Windows platforms.



(a) Single bottleneck link topology

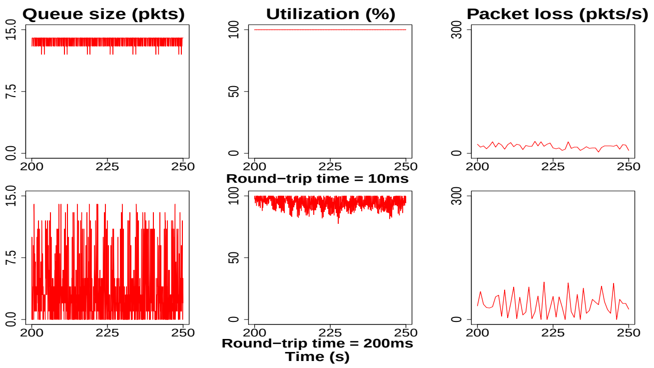


(b) Multi-bottleneck link topology

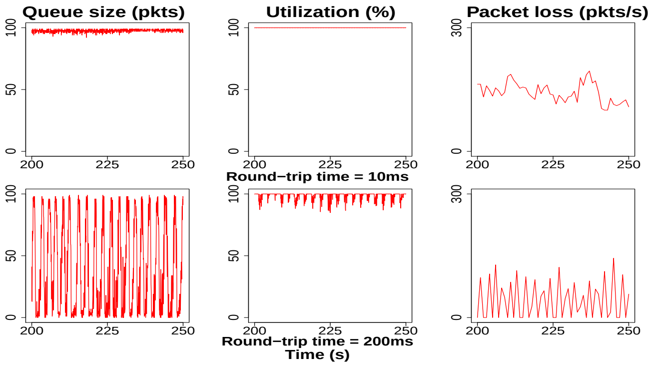
Figure 1: Simulation set-up for CUBIC and Compound TCPs

The following parameters are used for simulations: buffer size = 15, 100, $C * \overline{RTT}/\sqrt{N}$ and $C * \overline{RTT}$ packets, round-trip time (RTT) = 10 ms and 200 ms, bottleneck link capacity (C) = 100 Mbps, number of flows (N) = 60, and packet size = 1500 bytes. The currently deployed industry recommendation for $\overline{RTT}$ is 250 ms. Given the proliferation of the Windows platform, we choose a scenario where 80% of the flows are Compound TCP. This ratio between Compound and CUBIC TCP is just representative and a fuller set of experiments are left for further study. The topologies we used are depicted in Figure 1. In the figures, R's, S's and D's refer to the routers, the sources and the destination end-points, respectively. In the multi-bottleneck link topology the dotted lines represent the flows between the source and destination end-points.

The decision to choose small buffers in the range of 15 to 100 packets comes from the previous analysis of AIMD TCP Reno [11], [12]. These papers exhibit that small buffers have a stabilising effect on the end-to-end dynamics of TCP Reno. They also exhibit that even minor variations in buffer size can readily lead to the emergence of stable limit cycles. As new protocols are often designed to out-perform the standard TCP Reno, it is natural to begin an investigation of other transport protocols in similar buffer sizing regimes.

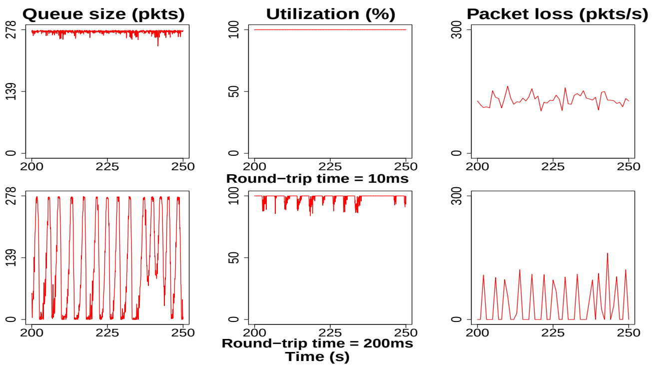


(a) Buffer size = 15 packets

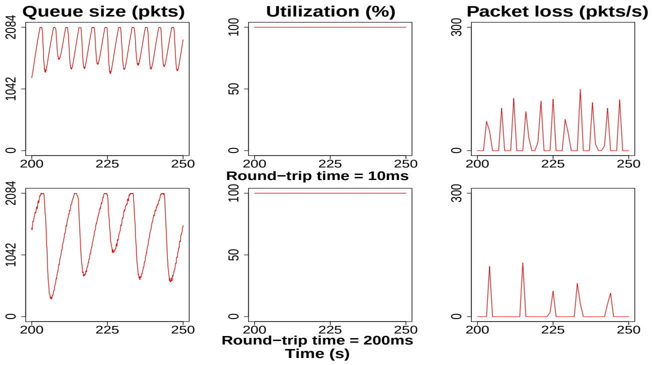


(b) Buffer size = 100 packets

Figure 2: Single bottleneck link with a capacity of 100 Mbps, 60 long-lived flows (80% Compound, 20% CUBIC) with round-trip times of $15, 100$ ms.



(a) Buffer size = $\frac{C*\overline{RTT}}{\sqrt{N}}$ packets



(b) Buffer size = $C * \overline{RTT}$ packets

Figure 3: Single bottleneck link with a capacity of 100 Mbps, 60 long-lived flows (80% Compound, 20% CUBIC) with round-trip times of $15, 100$ ms.

1) Single bottleneck, long-lived flows: In small buffers, see Figure 2, with 15 packet buffers the queue does not exhibit non-linear instabilities and there is a minor loss in utilization with larger RTTs. With 100 buffers, and with larger RTTs, the emergence of deterministic non-linear oscillations is clearly visible. With larger buffers, see Figure 3, we again witness non-linear oscillations which can also start to hurt utilization.

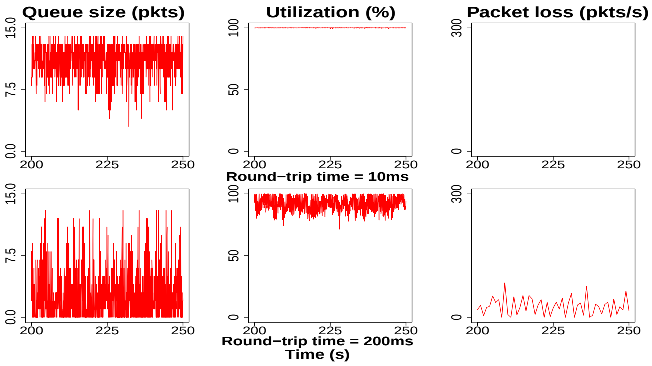
2) Single bottleneck, long-lived and short-lived flows: We observed the emergence of non-linear oscillations even with minor variations in a small buffer regime with long-lived flows. It is natural to investigate the impact a traffic mix of long-lived flows with HTTP flows in such a regime. Even with this traffic mix, see Figure 4, the non-linear instabilities prevail and there is an impact on utilization.

3) Multi-bottleneck link, Long-lived flows: It is natural to investigate the presence of multiple bottlenecks on the qualitative nature of the results observed with a single bottleneck topology. Due to space limitations, we only show results for the case of long-lived flows in a small buffer regime. Even with multiple bottlenecks, with cross traffic, the impact of small buffers on long-lived flow remains the same. From Figure 5, which shows the parameters of interest for Bottleneck Link 2, we again observe that 100 packet buffers induce non-linear oscillations whereas 15 packets have a stabilizing effect on the mix of CUBIC and Compound TCP flows. Further, flows with longer RTTs lead to a reduction in utilization.

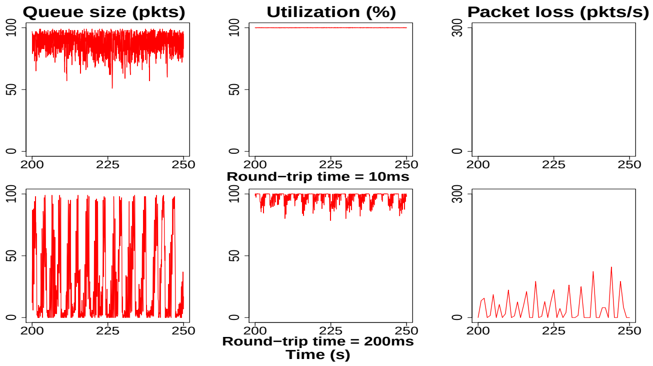
So far we focused on the impact of buffer size on the stability of the queue size, but the issue of TCP fairness is also important. The work in [1], [8] showed, using current design rules for buffers, that CUBIC TCP can be unfair to other CUBIC flows as also to other TCP variants. We now briefly comment on the issue of fairness between competing TCP flows in a small buffer regime; see Figure 6. Observe that CUBIC TCP still is unfair to other CUBIC flows, and also to Compound flows. This was despite CUBIC TCP flows being a small proportion of the overall flows.

IV. ANALYSIS OF ECN-BASED TRANSPORT PROTOCOL

In this section, we provide some analysis of the ECN-based transport protocol model that was outlined in Section II-C.

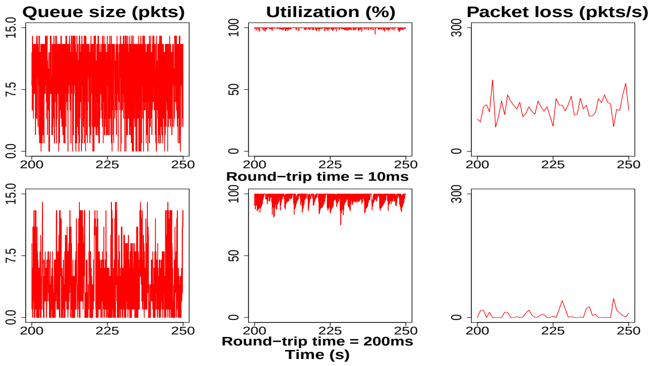


(a) Buffer size = 15 packets

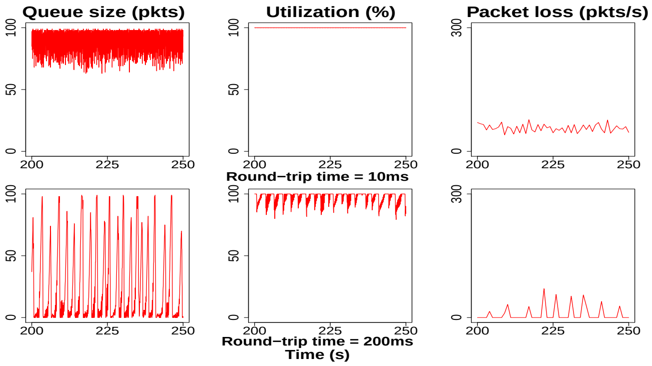


(b) Buffer size = 100 packets

Figure 4: Single bottleneck link with a capacity of 100 Mbps, 60 long-lived flows (80% Compound, 20% CUBIC), 20% short-lived HTTP flows with round-trip times of $15, 100$ ms.



(a) Buffer size = 15 packets



(b) Buffer size = 100 packets

Figure 5: Multi-bottleneck link with a capacity of 100 Mbps, 60 long-lived flows (80% Compound, 20% CUBIC), with round-trip times of $15, 100$ ms.

Consider a collection of flows all using a single resource, and that all the flows share the same gain parameter κ. Let $x(t) = \sum_r x_r(t)$ be the total flow through the link, and further let $w = \sum_r w_r$ represent the total weight. Additionally, we assume that a congestion indication signal generated at the link is returned to the source after a fixed and common RTT τ. Summing equation (9), and taking the time delay into account, we have

$$\frac{dx(t)}{dt} = \kappa\big(w - x(t-\tau)p\left(x(t-\tau)\right)\big)\cdot \tag{13}$$

Let x be the equilibrium point of equation (13), let $x(t) = x + u(t)$, and write p and p' for the values of the function $p(.)$ and $p'(.)$ at x. Then, linearising, we get

$$\frac{du(t)}{dt} = \kappa\tau(p + xp')u(t-\tau)\cdot \tag{14}$$

Using results from [10] we now state the following conditions for stability, and the onset of a Hopf type bifurcation. With the function (11), a necessary and sufficient condition for system (13) to be locally stable is $\kappa\tau p(1+B) < \pi/2$; further, the system undergoes a Hopf type bifurcation at $\kappa\tau p(1+B) = \pi/2$ producing an oscillatory solution with period 4τ. This simple example shows us that the larger the value of B, the greater the chance of the transport protocol undergoing a Hopf type bifurcation to induce limit cycles. Such deterministic periodic oscillations were also observed with the protocols we simulated, in a small buffer regime, as we varied the buffer from about 10 to 100 packets. With the

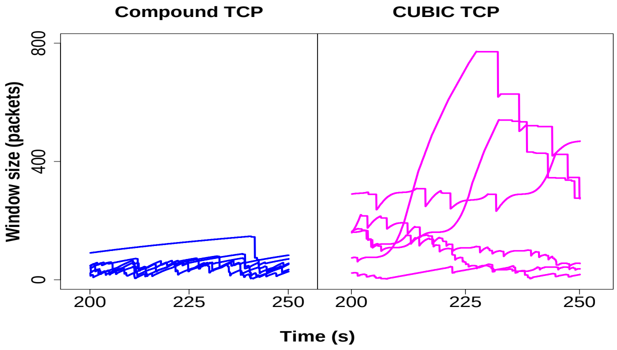


Figure 6: Sample window sizes for Compound and CUBIC. Single bottleneck, buffer 15 packets, and RTT of 200 ms.

functional form (12), a necessary and sufficient condition for system (13) to be locally stable is $\kappa\tau < \pi/2$; further, the system undergoes a Hopf type bifurcation at $\kappa\tau = \pi/2$ producing an oscillatory solution with period 4τ.

Let us explore another simple functional form for the resource. Let us suppose that the workload arriving at the resource over a time period δ is Gaussian with mean $x\tau$ and variance $x\delta\sigma^2$. Further, suppose that an incoming packet is marked, with an Explicit Congestion Notification bit, if when it arrives the workload that is already present in the queue is larger than the threshold level B. From the stationary distribution for a reflected Brownian motion [4]

$$p(x) = \exp\left(\frac{-2B(C-x)}{x\sigma^2}\right). \tag{15}$$

With the aforementioned resource design function, the condition for the first Hopf bifurcation becomes

$$\kappa\tau(1 + 2BC/(x\sigma^2))p(x) = \pi/2, \tag{16}$$

with period 4τ. Noting that the left-hand side of the above relation is increasing in $w(= xp(x))$, thus for any $w < C$, a condition for local stability is

$$\kappa\tau(1 + 2B/(\sigma^2)) < \pi/2. \tag{17}$$

These conditions clearly serve to highlight the destabilising impact of the threshold B. The threshold may be motivated in terms of buffer size, or in terms of thresholds for marking packets in active queue management schemes. So even if we had a largish buffer size, these models suggest that threshold for marking, or dropping packets may have a destabilising effect on queue dynamics. Now let us explore the design considerations and trade-off that arises for stability. An increase in the factor B causes p' to increase, causes an increased sensitivity in the resource's load. To counter the potentially destabilising effect of this increased sensitivity, there will have to be a reduction in the factor $\kappa\tau$ which represents the sensitivity of the response of the end-systems to the congestion indication signals. Thus, with ECN based transport protocols the form of the resource design again plays an important role for performance.

V. CONCLUSION AND FUTURE WORK

Today, the Internet has CUBIC TCP, Compound TCP and large buffers. With growth in communications capacity, router design with large buffers will not be scalable. Using simulations, for CUBIC and Compound, and the analysis of an illustrative ECN-based protocol, we reveal the rather subtle influence that small buffers could have on performance. A key phenomena which arises with even minor variations in buffer size is the emergence of limit cycles. These periodic cycles exhibit the loss of control theoretic stability, they induce periodic oscillations in the queue size and in the losses, and can also reduce link utilization.

Utilization is an important metric, but the network should not strive for a 100% utilization at the cost of large queue sizes which contribute to extra queuing delay. A small reduction in link utilization could well be acceptable if next-generation routers could be made faster or cheaper.

Queuing delay is a key concern for real-time services and is an added justification for having small buffered routers as an architectural consideration for a future Internet. *Packet loss* is important, but only within reason. In fact, loss is the primary feedback signal that is used in the Internet today and TCP has mechanisms to cope with loss. Packet loss can be handled; say, by forward error correction for real time traffic and by appropriate retransmission algorithms for other traffic. On the other hand it is rather difficult to compensate for queuing delay. *Non-linear oscillations* are observable in large bandwidth-delay product environments, when buffer sizes are not dimensioned appropriately. Deterministic, and periodic, queue size fluctuations will lead to bursty losses, they will induce jitter and can hurt link utilization. One really cannot predict their influence on quality of service for end-users. For example, they may prompt time-outs for web transfers, and may also defeat the purpose of forward error correction. We recommend the dimensioning of router buffer sizes to avoid such non-linear oscillations.

Our work shows that to develop a comprehensive understanding of next-generation network performance, we will have to investigate jointly the design of transport layer protocols, feedback from queues, and router buffer sizing.

ACKNOWLEDGMENTS

The authors would like to acknowledge the UK EPSRC Digital Economy Programme and the Government of India Department of Science and Technology (DST) for funding given to the IU-ATC. The authors would like to thank Professor V. Kamakoti for extending the facilities of the Reconfigurable Intelligent Systems Engineering (RISE) Lab at IIT-Madras.

REFERENCES

[1] I. Abdeljaouad, H. Rachidi, S. Fernandes, and A. Karmouch, "Performance analysis of modern TCP variants: A comparison of Cubic, Compound and New Reno", Symposium on Communications, May 2010, pp. 80–83, doi: 10.1109/BSC.2010.5472999.

[2] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance", IEEE/ACM Transactions on Networking, vol. 1, no. 4, August 1993, pp. 397–413, doi: 10.1109/90.251892.

[3] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant", ACM SIGOPS Operating Systems Review, vol. 42, no. 5, July 2008, pp. 64–74, doi: 10.1145/1400097.1400105.

[4] J. M. Harrison, "Brownian motion and stochastic flow systems", John Wiley & Sons, New York, 1985, ISBN: 0471819395.

[5] F. P. Kelly, “Mathematical modelling of the Internet”, Mathematics Unlimited - 2001 and Beyond, Springer-Verlag, 2001, pp. 685–702, ISBN: 9783540669135.

[6] F. P. Kelly, G. Raina, and T. Voice, “Stability and fairness of explicit congestion control with small buffers”, Computer Communication Review, vol. 38, no. 3, July 2008, pp. 51–62, doi: 10.1145/1384609.1384615.

[7] D. Leith, R.N. Shorten, and G. McCullagh, “Experimental evaluation of Cubic-TCP”, Sixth International Workshop on Protocols for Fast Long-Distance Networks, March 2008, pp. 1–9, doi: 10.1.1.64.9364.

[8] K. Munir, M. Welzl, and D. Damjanovic, “Linux beats windows!–or the worrying evolution of TCP in common operating systems”, Sixth International Workshop on Protocols for Fast Long-Distance Networks, 2007, pp. 43–48, doi: 10.1.1.69.905.

[9] G. Patil, S. McClean, and G. Raina, “Drop tail and RED queue management with small buffers: stability and Hopf bifurcation”, ICTACT Journal on Communication Technology, vol. 2, no. 2, June 2011, pp. 339–344, ISSN: 2229-6948(ONLINE).

[10] G. Raina, “Local bifurcation analysis of some dual congestion control algorithms”, IEEE Transactions on Automatic Control, vol. 50, no. 8, August 2005, pp. 1135–1146, doi: 10.1109/TAC.2005.852566.

[11] G. Raina, D. Towsley, and D. Wischik, “Part II: Control theory for buffer sizing”, ACM SIGCOMM Computer Communication Review, vol. 35, no. 3, July 2005, pp. 79–82, doi: 10.1145/1070873.1070885.

[12] G. Raina and D. Wischik, “Buffer sizes for large multiplexers: TCP queuing theory and instability analysis”, Proceedings of Euro-NGI Conference on Next Generation Internet, April 2005, pp. 173–180, doi: 10.1109/NGI.2005.1431663.

[13] P. Raja and G. Raina, “Delay based feedback, transport protocols and small buffers”, ACM/IEEE International Workshop on Quality of Service (IWQoS), San Jose, June 2011, pp. 1–3, doi: 10.1109/IWQOS.2011.5931330.

[14] K. T. J. Song, Q. Zhang, and M. Sridharan, “Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks”, 4th International Workshop on Protocols for Fast Long-Distance Networks, 2006, pp. 1–8, doi: 10.1.1.130.1595.

[15] A. Vishwanath, V. Sivaraman, and G. N. Rouskas, “Anomalous loss performance for mixed real-time and TCP traffic in routers with very small buffers”, IEEE/ACM Transactions on Networking, vol. 19, no. 4, August 2011, pp. 933–946, doi: 10.1109/TNET.2010.2091721.

[16] A. Vishwanath, V. Sivaraman, and M. Thottan, “Perspectives on router buffer sizing: recent results and open problems”, ACM SIGCOMM Computer Communication Review, vol. 39, no. 2, April 2009, pp. 34–39, doi: 10.1145/1517480.1517487.

[17] T. Voice and G. Raina, “Stability analysis of a max-min fair rate control protocol (rcp) in a small buffer regime”, IEEE Transaction on Automatic Control, vol. 54, no. 8, August 2009, pp. 1908–1913, doi: 10.1109/TAC.2009.2022115.

[18] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, “FAST TCP: motivation, architecture, algorithms, performance”, IEEE/ACM Transactions on Networking, vol. 14, no. 6, December 2006, pp. 1246–1259, doi: 10.1109/TNET.2006.886335.

[19] D. Wischik and N. McKeown, “Part I: Buffer sizes for core routers”, ACM SIGCOMM Computer Communication Review, vol. 35, no. 3, July 2005, pp. 75–78, doi: 10.1145/1070873.1070884.

[20] NS2, [Online]. Available: *http://www.isi.edu/nsnam/ns/*, [Accessed: 20th December 2011].