ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2011.0047

DROP TAIL AND RED QUEUE MANAGEMENT WITH SMALL BUFFERS: STABILITY AND HOPF BIFURCATION

Ganesh Patil[1], Sally McClean[2] and Gaurav Raina[3]
[1]*Department of Computer Science and Engineering, Indian Institute of Technology Madras, India*
E-mail: ganeshp@cse.iitm.ac.in
[2]*School of Computing and Information Engineering, University of Ulster, Ireland*
E-mail: sally@infc.ulst.ac.uk
[3]*Department of Electrical Engineering, Indian Institute of Technology Madras, India*
E-mail: gaurav@ee.iitm.ac.in

Abstract

There are many factors that are important in the design of queue management schemes for routers in the Internet: for example, queuing delay, link utilization, packet loss, energy consumption and the impact of router buffer size.

By considering a fluid model for the congestion avoidance phase of Additive Increase Multiplicative Decrease (AIMD) TCP, in a small buffer regime, we argue that stability should also be a desirable feature for network performance. The queue management schemes we study are Drop Tail and Random Early Detection (RED). For Drop Tail, the analytical arguments are based on local stability and bifurcation theory. As the buffer size acts as a bifurcation parameter, variations in it can readily lead to the emergence of limit cycles. We then present NS2 simulations to study the effect of changing buffer size on queue dynamics, utilization, window size and packet loss for three different flow scenarios. The simulations corroborate the analysis which highlights that performance is coupled with the notion of stability.

Our work suggests that, in a small buffer regime, a simple Drop Tail queue management serves to enhance stability and appears preferable to the much studied RED scheme.

Keywords:
TCP, Queue Management, Small Buffers, Performance

1. INTRODUCTION

Enhancing network performance and reducing energy consumption by the network are two very important design challenges faced by network operators today. Network performance depends on the design of transport protocols in the end-systems and the choice of queue management schemes in routers. Queue management schemes serve to give feedback to end-systems, by either dropping packets or marking packets with Explicit Congestion Notification (ECN) marks [10]. In the network, routers play an important role in energy consumption, and in routers the size of the buffers used has a direct impact on it; see [1] for an extended discussion on this relationship.

The capacities of Internet routers are limited by the buffers they must use to hold packets. The challenge is that buffers need to be both large and fast. Buffers are currently sized using a rule of thumb which says that each link needs a buffer of size $B = T * C$, where T is the average round-trip time of the flows passing across the link, and C is the data rate of the link [12]. For example, a 10Gb/s router linecard needs approximately 250*ms* * 10Gb/s = 2.5Gbits of buffers, enough to hold roughly 200*k* packets. Buffers also need to be fast: a typical 10Gb/s router linecard needs to access the buffer once every 30 ns, and this access time must decrease in proportion to the link speed, so that a 40 Gb/s linecard needs to access the buffer every 7.5 ns. It is safe to say that the speed and size of buffers is the single biggest limitation to growth in router capacity today. If buffers were small enough to be held in on-chip SRAM (e.g. 32Mbits of buffers), they would remove the memory bottleneck for electronic routers.

The study of queue management has a long history among networking researchers. As yet, however, there is still no consensus on the optimal set of queue management algorithms. The lack of consensus serves to exhibit the rather difficult nature of the problem. Given the potentially promising prospect of having small buffered routers in next generation networks, we focus our attention on the impact of some commonly proposed queue management schemes in a small buffer regime. The schemes we study are Drop Tail and Random Early Detection (RED) [3].

Given the linkage between buffer size and both performance and energy, we are motivated to study queue management schemes in a small buffer regime. For our analysis, we consider a fluid model for the congestion avoidance phase of AIMD TCP with small Drop Tail buffers. The model is for long lived TCP flows. We first study this nonlinear model via a local stability and a local bifurcation analysis. We perform NS2 simulations with small Drop Tail buffers to corroborate our analysis. We also consider a mixture of TCP and UDP flows, and also a mixture of TCP with short lived flows. We observed that RED was quite sensitive to the precise choice of buffer size; thus, given the simplicity of Drop Tail it appears to be advisable to opt for Drop Tail for both performance and energy considerations.

The rest of the paper is organized as follows. In Section 2, we outline some queue management schemes, analyze a fluid model for AIMD TCP, and perform some NS2 simulations to corroborate our analysis. In Section 3, we conclude and outline avenues for further research.

2. QUEUE MANAGEMENT

Queue management schemes can broadly be divided into two groups: schemes that use the instantaneous queue size, like Drop Tail, and schemes that advocate an element of averaging of the queue size, like RED, before dropping or marking decisions are made. We focus on Drop Tail and RED.

Drop Tail

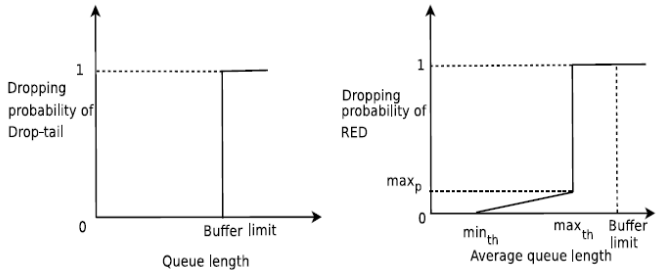
Drop Tail is perhaps the simplest queue management policy; it drops all incoming packets after the buffer is full.

Random Early Detection (RED)

The goals of the RED algorithm [3], as per RFC 2309 [2], are to reduce queuing delay and packet loss, to maintain high link utilization, to better accommodate bursty sources, and to provide a low-delay environment for interactive services by maintaining a small queue size. In this paper we use drops, instead of ECN marks, as the feedback signal to the end-systems. After the arrival of each packet, the RED algorithm calculates the average queue size *avg* as follows:

$$avg = (1 - w_q) * \overline{avg} + w_q * q \ ,$$

where w_q is the queue weight, q is the instantaneous queue size, and $\overline{avg}$ is the previous average queue size. If the *avg* is less than min_{th} then packets are enqueued. If the *avg* is more than max_{th} then all the incoming packets are dropped. If the *avg* is in between min_{th} and max_{th}, then packets are dropped with a probability p_{a}. The drop functions for Drop Tail and RED are shown in Fig.1.



(a) Drop Tail

(b) RED

Fig.1. Drop functions of Drop Tail and RED

Many variants of the RED algorithm have been proposed; they either modify the calculation of the drop function or propose different parameter settings.

Fluid Model for TCP

Consider a single TCP flow, whose window size at time t is $W(t)$. When there are no loss indications, W increases by one packet every RTT; when there is a loss indication, W is cut in half. The rate at which packets are emitted at time t is roughly $W(t)/RTT$, so the rate at which acknowledgements or loss indications are received at time t is $W(t - RTT)/RTT$. Let $p(t)$ be the packet loss probability for packets emitted at time t.

Suppose there are N flows, and let $W^N(t)$ be the sum of all the window sizes. In the interval $(t, t+ \delta)$, $W^N(t)$ changes in two ways. First, there is a decrement due to window halving: the total number of flows which receive loss indications is roughly,

$$\delta \frac{W^N(t - RTT)}{RTT} p(t - RTT))$$

and (assuming each flow is equally likely to receive a loss indication) the average reduction in window size for each of these flows is $W^N(t)/2N$. Second, there is an increment of $\delta(N/RTT - O(\delta))$, since each flow increases its window size by δ/RTT, except for those which receive loss indications. The net change in window size is

$$W^N(t+\delta) - W^N(t) \approx \frac{\delta N}{RTT} - \frac{W^N(t)}{2N}\left[\delta \frac{W^N(t-RTT)}{RTT} p(t-RTT)\right].$$

This suggests that the average window size $w(t) = W^N(t)/N$ should not depend on N, and should obey a differential equation

$$\frac{dw(t)}{dt} = \frac{1}{RTT} - \frac{w(t)}{2}\left[x(t-RTT)p(t-RTT)\right].$$

An approximation used is that packets are being emitted at rate $W(t)/RTT$ at time t, which means we are modeling a rate-based mechanism parameterized by $W(t)$ rather than a window-based mechanism.

Fluid Model for the Queue

Let the total arrival rate to the queue at time t be $X^N(t) = W^N(t)/RTT$, and let $x(t) = X^N(t)/RTT$. In the interval $(t, t + \delta)$, the total arrival rate changes by $N\delta x'(t)$ and a total of $N\delta x(t)$ packets arrive. Suppose the queue has service rate NC and buffer size B^N. Lindleys' recursion gives us an idea of how the queue size $Q^N(t)$ will evolve:

$$Q^N(t+\delta) \approx \left[Q^N(t) + \delta N_x(t) - \delta NC\right]_0^{B^N}$$

where $[q]_0^b = min(max(q,0),b)$. Depending on how B^N is chosen, this can lead to different queuing models. For example, if $B^N = \sqrt{N}B$, then it is entirely possible for the queue to go from empty to full in a short interval $(t, t + \delta)$, if N is large enough; if $B^N = NB$ this is not possible. Consider the case of small buffers, i.e. $B^N = N^{\gamma}B$ where $\gamma = 0$. Note first that the maximum possible queuing delay is B/NC which is negligible for large N.

Consider an open-loop queuing system with N flows, in which each flow has mean rate x. As $N \rightarrow \infty$, the aggregate arrival process will converge to a Poisson process, assuming that the packet inter arrival time is bounded away from zero, in the following sense: if $A^N(t,u)$ is the total number of packets arriving in the interval (t, u), then the random process $\tilde{A}^N = A^N(t, t+u/N)$ converges to a Poisson process with rate x. This result carries through to queue size: if $Q^N(t)$ is the queue size at time t, then the distribution of $Q^N(t)$ converges to that of a queue fed by a Poisson process with arrival rate x and served at constant rate C, in a infinite-buffer system, assuming $x < C$. We expect that this result can be extended to a system with a finite buffer B, and thence to $x \geq C$. The loss probability for a finite-buffer openloop queue is thus $p = L_B(x/C)$, where $L_B(\cdot)$ can be calculated by finding the equilibrium distribution of a suitable Markov Chain. Now $Q^N(t)$ makes excursions of size $O(1)$ in timescale $O(1/N)$. For intuition, consider an $M_{N_x}/M_{NC}/1$ queue, which is just an $M_x/MC/1$ queue speeded up by a factor of N. Therefore the $M_x/MC/1$ queue hits any given size B in timescale $O(1/N)$.

Since the timescale of queuing phenomena is $O(1)$, one can claim that *in the closed-loop system* if the mean arrival rate $x(t)$ doesn't change by much in a short interval, then the loss probability is $p = L_B(\rho(t))$, $\rho(t) = x(t)/C$. This is because, over a short enough interval, the queue can't tell if it is being fed by open-loop or by closed-loop traffic; it sees an input process which is a near-constant-rate Poisson flow. For a more detailed exposition, on the queuing theoretic arguments, see [4], [9].

2.1 AIMD TCP WITH DROP – TAIL

The fluid model for the congestion avoidance phase of AIMD TCP is,

$$\frac{dw(t)}{dt}=\frac{1}{RTT}-\frac{w(t)}{2}\left[x(t-RTT)p(t-RTT)\right]. \qquad (1)$$

Now, let $x(t)$ be the total rate at which packets arrive at the queue, and let C be the service rate. Let $L_B(x)$ be the packet loss probability for a queue with buffer size B, service rate C and Poisson arrivals of rate x. It was argued in [8], [9] that Poisson arrivals are a good approximation when buffers are small. Thus $p(t) = L_B(x(t))$.

It was also argued in [8], [9] that, for large number of flows, the blocking probability of an $M/M/1$ queue is a reasonable model for the packet loss incurred by a small buffer Drop Tail router. Thus, for the model, the routers will be assumed to have the following packet loss model:

$$p(t) = (x/C)^B \qquad (2)$$

where C is the service rate and B is the buffer size. The packet drop probability is a function of rate x, and the average window size at time t is $w(t) = x(t)RTT$. Thus Eq.(1), outlined above, becomes

$$\frac{dx(t)}{dt}=\frac{1}{RTT^2}-\frac{x(t)x(t-RTT)p(x(t-RTT))}{2} \qquad (3)$$

with equilibrium, $x=\frac{1}{RTT}\sqrt{\frac{2}{p}}$.

Local Stability and Local Hopf Bifurcation Analysis

Let x^* be the equilibrium point of the system (3), let $x(t) = x^*+u(t)$, and linearize about x^*, we get the equation,

$$\frac{du(t)}{dt}=-au(t)-bu(t-RTT) \qquad (4)$$

where $a=\frac{1}{2}x^*p(x^*),\ b=\frac{1}{2}x^*\left(p(x^*)+x^*p'(x^*)\right).$

We now recall some results about Eq.(4) [7], where $a \geq 0$, $b > 0$, $b > a$, and $RTT > 0$. A sufficient condition for stability is

$$bRTT<\frac{\pi}{2} \qquad (5)$$

and the system undergoes a Hopf bifurcation at

$$RTT\sqrt{b^2-a^2}=cos^{-1}(-a/b) \qquad (6)$$

with period $2\pi RTT/\cos^{-1}(-a/b)$.

Now in terms of network parameters, we may state the following about Eq.(3). A sufficient condition for local stability, using the drop function (2), is

$$\frac{1}{w^*}(1+B)<\pi/2\,. \qquad (7)$$

The two parameters which feature in the condition are the equilibrium window and the buffer size. The condition will be harder to satisfy as buffers get larger. Further, the system (3) will undergo a Hopf bifurcation at

$$\frac{1}{w^*}\sqrt{B(B+2)}=cos^{-1}\left(\frac{-1}{1+B}\right) \qquad (8)$$

with period $(2\pi RTT)/\left(cos^{-1}\left(\frac{-1}{1+B}\right)\right)$. Observe that the Hopf condition also has the equilibrium window size, and the buffer size, and the period depends on the round-trip time and the buffer size.

A Brief on the Hopf Bifurcation

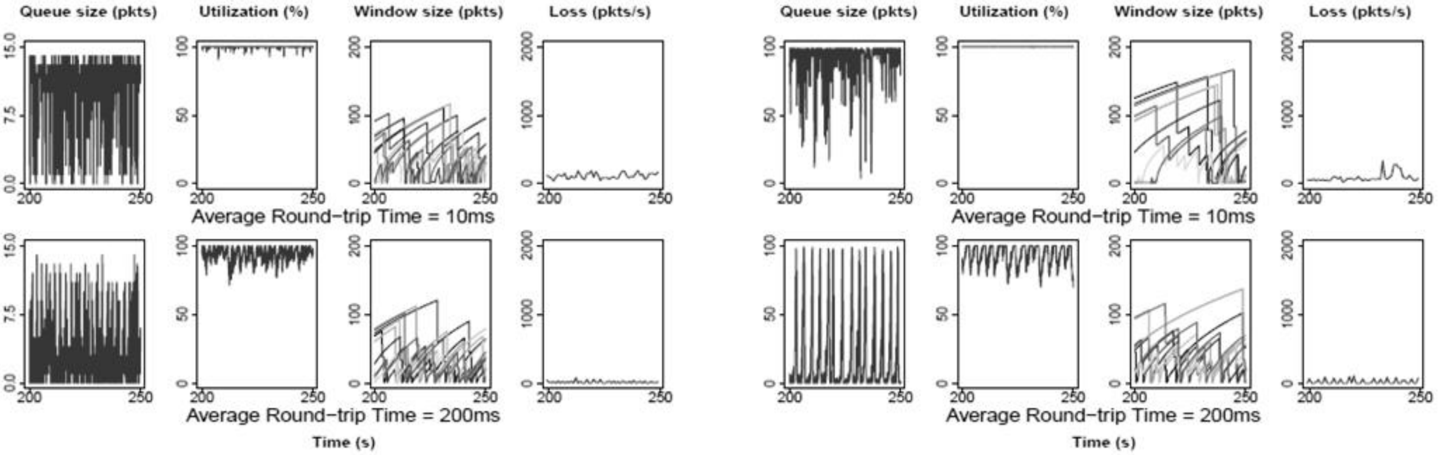
As conditions obtained by local stability analysis get violated, bifurcations may occur. A very common behavior of nonlinear systems (apart from convergence to a stable equilibrium) is the emergence of limit cycles which, like equilibria, may also be stable or unstable. The Hopf bifurcation is a way to analyze the emergence and stability of limit cycles bifurcating from a stable equilibrium. As a brief introduction to Hopf bifurcation theory [5], let us assume that we have a system of differential equations $dx/dt = f_\eta(x)$ on $\mathbb{R}^n$, with a locally unique equilibrium x^* that is stable for $\eta < \eta_c$ and unstable for $\eta > \eta_c$. Further assume that $Df(x^*)$ and the characteristic exponents at x^* are continuous in η and the stability changes when one pair of complex conjugate characteristic exponents crosses the imaginary axis. Now let ζ, $\bar{\zeta}$ be the corresponding eigenvectors of $Df(x^*)$, then at η_c the linearized system has periodic solutions lying in the plane of Re(ζ) and Im(ζ). A geometric approach (based on the central manifold theorem) shows that for η near η_c, there is a 2-manifold invariant under the flow tangent to Re(ζ) and Im(ζ). This is where a lot of the interesting dynamics take place. It is indeed possible to analyze the motion on this central manifold, and one way to do it is by parameterizing the central manifold by a single complex variable and then essentially using the method of averaging [5], [7].

We still need to determine the type of the Hopf bifurcation in Eq.(3), i.e. if it is super-critical or sub-critical: which is beyond the scope of this article due to space limitations. In the next section, we conduct simulations, with Drop Tail and the RED algorithm with variations in the buffer size, which serve to exhibit the onset of limit cycle dynamics.

2.2 SIMULATIONS AND DISCUSSION

We now simulate, using NS2 [13], Drop Tail and RED in a small buffered environment. The network set-up is a single bottleneck dumbbell topology, and the bottleneck capacity used is 100Mbps. The simulations are conducted over smaller round-trip times (10ms) as well as larger round-trip times (200ms).

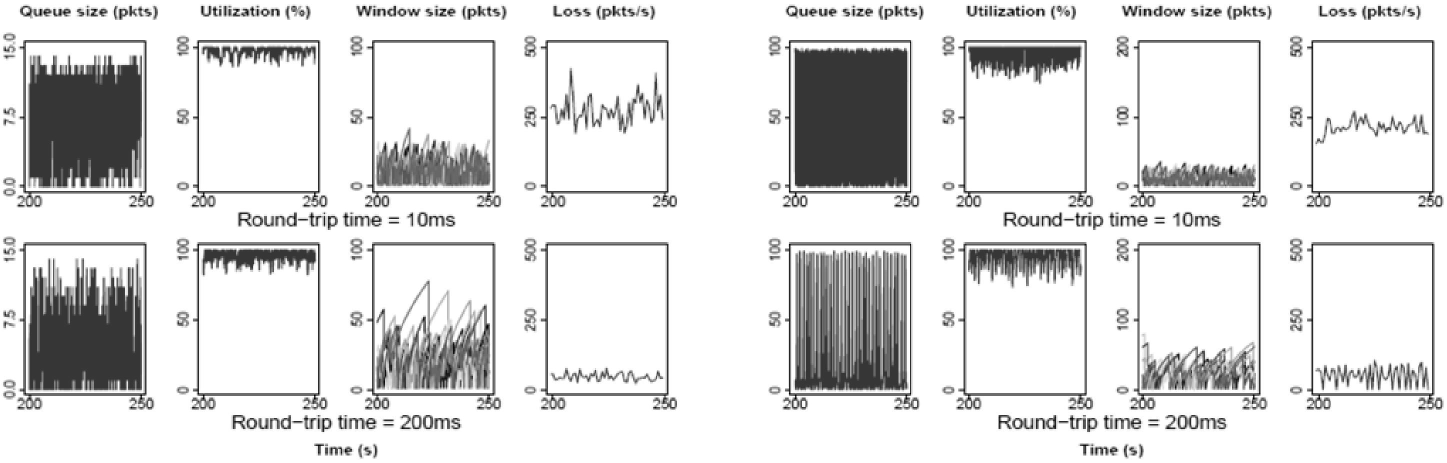
We consider three types of traffic. The first set of traffic has only long lived TCP flows. The second type of traffic has long lived flows mixed with UDP flows. And finally, we also consider the mix of long lived and HTTP flows. The packet size is set to 1500 bytes. We monitor the following quantities: the queue size (in packets), the link utilization (in percentage), the evolutions of the window size for 10 randomly chosen TCP flows (in packets), and loss (in packets per second).



(a) Buffer size = 15 packets

(b) Buffer size = 100 packets

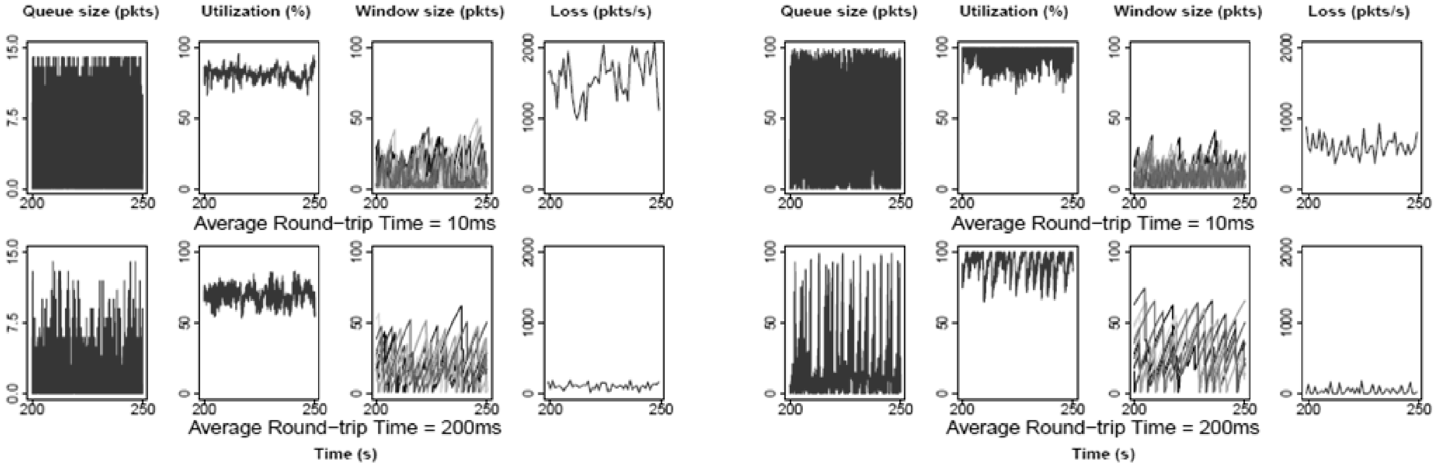
Fig.2. *Drop Tail with TCP flows*. Bottleneck capacity = 100Mbps, Number of flows = 60, each with a 2Mbps link



(a) Buffer size = 15 packets

(b) Buffer size = 100 packets

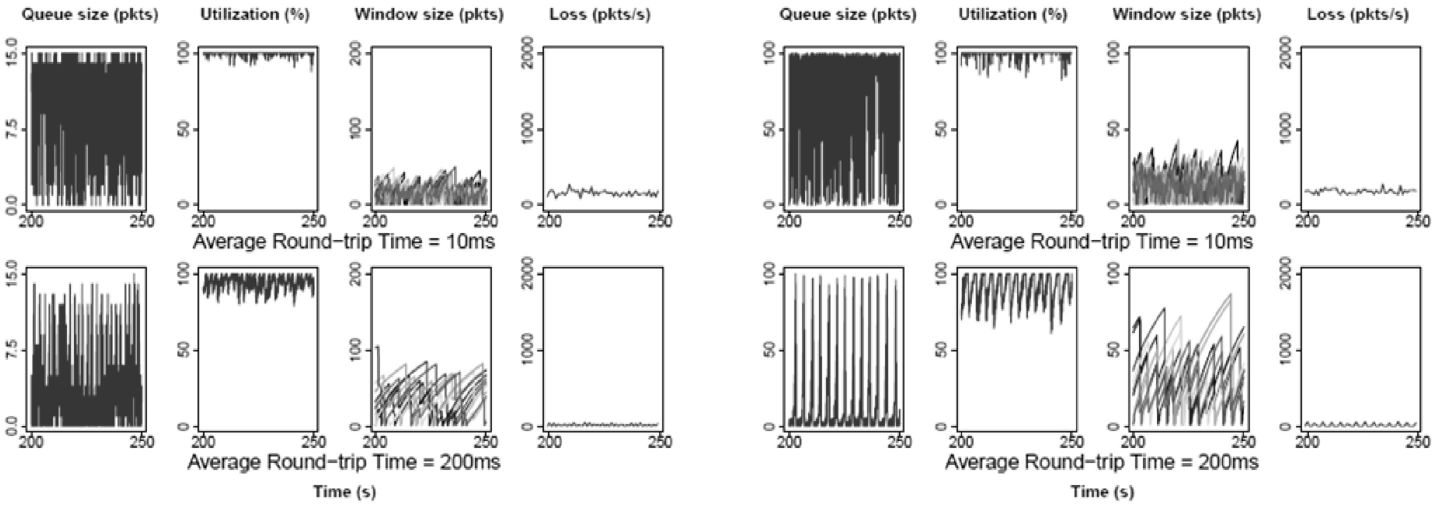
Fig.3. *Drop Tail with TCP and UDP flows*. Bottleneck capacity = 100Mbps, Number of TCP flows = 50, each with a 2Mbps link, Number of UDP flows = 20 each with a 1Mbps link



(a) Buffer size = 15 packets

(b) Buffer size = 100 packets

Fig.4. *Drop Tail with TCP and HTTP flows*. Bottleneck capacity = 100Mbps, Number of TCP flows = 50, each with a 2Mbps link, Number of HTTP flows = 180 contributing 10Mbps



(a) Buffer size = 15 packets, min_{th} = 5, max_{th} = 15

(b) Buffer size = 100 packets, min_{th} = 30, max_{th} = 100

Fig.5. *RED with TCP flows*. Bottleneck capacity = 100Mbps, max_p = 0.02, w_q = 0.001, Number of flows = 60, each with a 2Mbps link

Drop Tail: For Drop Tail, we show the plots with buffer sizes of 15 and 100 packets with average RTTs of 10ms and 200ms. Fig.2(a) and Fig.2(b) show the plots of Drop Tail with TCP flows. Fig.3(a) and Fig.3(b) show the plots of Drop Tail with TCP and UDP flows. Fig.4(a) and Fig.4(b) show the plots of Drop Tail with TCP and HTTP flows.

At the buffer size of 15 packets, the queue is stochastic and stable. As we vary the buffer size from 15 to 100 packets, as per the local stability and bifurcation theory, we observe the emergence of deterministic oscillations in the form of (stable) limit cycles. This phenomenon confirms that stability should indeed be a metric for network performance. This qualitative change in the system dynamics leads the TCP flows to get synchronized. Thus it would be prudent to choose buffer size and AQM schemes to ensure that the system is stable.

Indeed, as can be seen from Fig.2(a) and Fig.2(b) the formation of oscillatory dynamics is visible in the traces of the queue size. With slightly larger buffers, i.e. 100 packets, and with larger round-trip times (see 200ms of Fig.2(b)), the oscillations in the queue size are more prominently visible. From the window sizes as in Fig.2(b), the TCP flows clearly seem to get synchronised. The same phenomena can be seen when we have TCP and UDP flows together; see Fig.3(a) and Fig.3(b), and also when we have TCP and HTTP flows together; see Fig.4(a) and Fig.4(b). However, utilization drops in the case of TCP mixed with HTTP and UDP flows.

RED: For the simulations with the RED algorithm (which was configured for dropping packets), we used the same network parameters as outlined above. The maximum dropping probability max_p was set to 0.02, which is the value used in [3]. The weight parameter w_q was set to 0.001, which is close to the value used in [3]. Here also, we focus on two buffer sizes: 15 and 100. For a buffer of 15 packets, we used min_{th} as 5, max_{th} as 15, and for a buffer of 100 packets, we took min_{th} as 30, max_{th} as 100.

As we move from a buffer of 15 to 100 packets, we again observe distinct determinate oscillations in the queue size; see Fig.5(a) and Fig.5(b). In this respect the results are qualitatively similar to those obtained from Drop Tail. The results were qualitatively similar to the results obtained for Drop Tail for other types of traffic also and therefore we do not show plots for these.

3. OUTLOOK

The study of queue management schemes continues to be an area of active research. The question of sizing buffers, and in particular the prospect of networks with small buffers, now provides us with a new platform under which queue management schemes could be evaluated.

We studied a fluid model of AIMD TCP coupled with a fluid model of Drop Tail in a small buffer regime. The nonlinear model was amenable to analysis using control and bifurcation theory. We provided sufficient conditions for local stability, and also conditions to ensure the existence of a local Hopf bifurcation. In essence, the larger the buffer size the greater the possibility of violating the Hopf bifurcation condition. NS2 simulations served to verify the analysis, and exhibited the onset of stable limit cycles.

We also experimented with the commonly proposed RED algorithm with the same choice of network parameters. Again we noticed the emergence of stable limit cycles induced by changes in buffer size. Given that buffer size has an impact on performance and energy consumption, our current work suggests that the simple Drop Tail policy would be favorable over the more involved RED algorithm.

Avenues for Further Research

First, we need to develop models for the interaction of TCP and HTTP flows, and also for a mixture of TCP and UDP flows. We also need to analyse and simulate networks which have multiple and diverse round trip times, and also networks with multiple bottlenecks. It would also be useful to understand better the impact of any additional averaging that may be performed at the queues, as is currently advocated by queue management schemes like RED. A potential starting point for the development of light weight schemes could be the proposals outlined in [6].

Currently there is a lot of interest in revising the protocols and the architectural principles for the current Internet. An outline of a range of architectural issues and proposals is given in [11]. The relationship between network performance, energy consumption and buffer size is becoming apparent. It would be worthwhile to see if any other future architectural issues, as outlined in [11], are also impacted by the buffer sizing question.

ACKNOWLEDGEMENTS

This work was carried out under the IU-ATC project funded by the Department of Science and Technology (DST), Government of India, and the UK EPSRC Digital Economy Programme.

REFERENCES

[1] G. Appenzeller, “Sizing Router Buffers”, Ph.D. diss., Department of Computer Science, Stanford Univ, 2004.

[2] B. Braden, *et al.*, “Recommendations on queue management and congestion avoidance in the Internet”, RFC 2309, IETF, 1998.

[3] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance”, *IEEE/ACM Trans. on Networking*, Vol.1, No. 4, pp. 397–413, 1993.

[4] Ganesh, N. O’Connell, and D. Wischik, “*Big Queues*,” Lecture Notes in Mathematics, Springer, 2004.

[5] B.D. Hassard, N.D. Kazarinoff, and Y. Wan, “*Theory and applications of Hopf bifurcation,” Cambridge*: Cambridge University Press, 1981.

[6] P.G. Kulkarni, S.I. McClean, G.P. Parr, and M.M. Black, “Lightweight proactive queue management,” *IEEE Trans. on Network and Service Management*, Vol. 3, No. 2, pp. 1-11, 2006.

[7] G. Raina, “Local bifurcation analysis of some dual congestion control algorithms,” *IEEE Trans. on Automatic Control*, Vol. 50, No. 8, pp. 1135–1146, 2005.

[8] G. Raina, D. Towsley, and D. Wischik, “Part II: Control theory for buffer sizing,” *ACM SIGCOMM Trans. Computer Communication Review*, Vol. 35, No. 3, pp.79–82, 2005.

[9] G. Raina and D. Wischik, “Buffer sizes for large multiplexers: TCP queueing theory and instability analysis,” in *Proc. of EuroNGI Conference on Next Generation Internet*, 2005.

[10] K.K. Ramakrishnan, S. Floyd, and D. Black, “The addition of Explicit Congestion Notification (ECN) to IP,” RFC 3168, Proposed Standard, 2001.

[11] Ramamurthy, G.N. Rouskas and K. Sivalingam, Eds., “*Next-Generation Internet: Architectures and Protocols*,” New York: Cambridge University Press, 2011.

[12] Wischik and N. McKeown, “Part I: Buffer sizes for core routers,” *ACM SIGCOMM Trans. Computer Communication Review*, Vol. 35, No. 2, pp. 75–78, 2005.

[13] NS2, “The Network Simulator NS2 homepage.” Available at: http://www.isi.edu/nsnam/ns/. Accessed 1 April 2011.