Buffer sizes for large multiplexers: TCP queueing theory and instability analysis

Gaurav Raina (Statistical Laboratory, Cambridge University, G.Raina@statslab.cam.ac.uk)
Damon Wischik (Computer Science, UCL, D.Wischik@cs.ucl.ac.uk)

***Abstract*— In large multiplexers with many TCP flows, the aggregate traffic flow behaves predictably; this is a basis for the fluid model of Misra, Gong and Towsley [1] and for a growing literature on fluid models of congestion control. In this paper we argue that different fluid models arise from different buffer-sizing regimes. We consider the large buffer regime (buffer size is bandwidth-delay product), an intermediate regime (divide the large buffer size by the square root of the number of flows), and the small buffer regime (buffer size does not depend on number of flows). Our arguments use various techniques from queueing theory.**

We study the behaviour of these fluid models (on a single bottleneck link, for a collection of identical long-lived flows). For what parameter regimes is the fluid model stable, and when it is unstable what is the size of oscillations and the impact on goodput? Our analysis uses an extension of the Poincaré-Linstedt method to delay-differential equations.

We find that large buffers with drop-tail have much the same performance as intermediate buffers with either drop-tail or AQM; that large buffers with RED are better at least for window sizes less than 20 packets; and that small buffers with either drop-tail or AQM are best over a wide range of window sizes, though the buffer size must be chosen carefully. This suggests that buffer sizes should be much much smaller than is currently recommended.

I. Introduction

In 2000 Misra, Gong and Towsley [1] published a differential equation model, also called a fluid model, for TCP. There is now a substantial literature [2] covering a variety of fluid models for Internet congestion control—this begs the question of which fluid model is most useful. One way to answer this is by simulation, and of course every proposed model has been accompanied by simulations. Another way is to look for limit theorems which say that a given fluid model is obtained asymptotically, for example as the number of flows increases, in some idealized system. Limit theorems can alert us to effects which are hard to spot with simulation—we will describe certain instabilities which we predict are only seen in systems with more than 5,000 or so flows, common in backbone routers, hard to simulate[1].

In this paper we will explore several different buffer-sizing regimes. We will work with large multiplexers, i.e. we will let the service rate be proportional to the number of flows, and we will consider what happens when this is very large. Now, there are several ways to choose buffer size. In the *large buffer* regime we let buffer size be proportional to the number of flows—this is the standard rule of thumb, which says to choose buffer equal to bandwidth-delay product. In the *small buffer* regime we choose a fixed buffer size and rely on statistical multiplexing to keep loss low. There is also an *intermediate regime* suggested in [3]. We will use various techniques from queueing theory to argue that these three regimes lead to different fluid models. The cornerstone of our argument is an account of the relative timescales of queueing phenomena and of flow-control phenomena. This approach was inspired by [4]. It has also been used in [5] to address a very similar question, though using different queueing-theoretic techniques.

We will then analyse these fluid models, using the dynamical-systems techniques described in detail in [6], [7]. We first calculate whether the system is locally stable. (Local stability is reasonably well-understood; see [2], [8]–[10] and references therein.) When the system is unstable we compute the size of the limit cycles (i.e. oscillations) in traffic rate. These limit cycles are a sign of synchronization: when the system is locally stable the flows are totally unsynchronized; when there are limit cycles there is some degree of synchronization in the TCP sawtooths; the larger the limit cycles the greater the degree of synchronization. We also study the impact of limit cycles on goodput. This lets us rank the different buffer-sizing regimes, as mentioned above in the abstract.

Here is an outline of the dynamical-systems theory. Given a delay-differential equation $dx_t/dt = f(x_t, x_{t-T})$, first find an equilibrium point: $f(x^*, x^*) =$

We are grateful to P. Giaccone, E. Leonardi, M. Handley and F. Kelly for helpful discussions. GR was funded by EPSRC grant GR/S86266/01. DJW is supported by a University Research Fellowship from the Royal Society.

[1]We are currently working on a custom large-scale simulator.

0. Next, write down a modified system by introducing an artificial gain parameter $\kappa > 0$:

$$dx_t/dt = \kappa f(x_t, x_{t-T}). \quad (1)$$

Determine local stability of the modified system, by guessing the solution $x_t = x^* + e^{\lambda t}$ and solving the first-order approximation to (1) for λ; the system is locally stable about x^* if λ has negative real part. Find the critical value κ_c, the largest gain κ such that the modified system is locally stable. (Therefore the original system is locally stable if $\kappa_c > 1$.) When $\kappa > \kappa_c$, use a power series expansion based on the Poincaré-Linstedt method to find the amplitude of the resulting limit cycles; the answer is $\sigma\sqrt{\kappa - \kappa_c} + O(\kappa - \kappa_c)$ where σ can be calculated. Therefore the limit cycles for the original system have amplitude $\approx \sigma\sqrt{1 - \kappa_c}$. Since this comes from a power series expansion, it is only accurate when it predicts small limit cycles; we suspect that when it predicts large limit cycles then the actual system also behaves badly, though not in exactly the manner we have calculated.

A limitation in this work is that we only consider a single bottleneck router, shared by a collection long-lived flows with common round trip time. Some local stability results are known for networks with heterogeneous round trip times [2], [9], [10]. Those results indicate that such networks may have limit cycles: we therefore believe that the behaviours we analyse here are not artefacts of our simple setup. We hope that our work is a useful stepping stone to a comprehensive theory.

The paper is organized thus: in Section II we review the fluid model for TCP, and explain the connection with buffer sizing regimes. In Sections III–VI we describe four regimes: small buffer, intermediate buffer, large buffer with AQM, large buffer with droptail. In Section VII we propose a rule of thumb for deciding which of these regimes is relevant. In Section VIII we summarize the different regimes.

II. Overview of limiting regimes

Consider a single bottleneck queue with service rate NC shared by a large number N of TCP flows, each with the same round trip time RTT. We will first recapitulate the differential equation for TCP [1].

A. 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. (We may as well assume the queue is located adjacent to the source, and that all the propagation delay comes after the queueing delay, since the source dynamics are the same regardless of where along the round-trip path the queue is located. Then $p(t)$ is the loss probability for packets which arrive at the queue at time t.)

Suppose now 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

$$W^N(t + \delta) - W^N(t) \approx \frac{\delta N}{RTT} - \frac{W^N}{2N}\Big(\delta \frac{W^N(t - RTT)}{RTT} p(t - RTT)\Big).$$

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

$$\frac{dw(t)}{dt} = \frac{1}{RTT} - \frac{w(t)}{2}\Big(\frac{w(t - RTT)}{RTT} p(t - RTT)\Big). \quad (2)$$

In this paper, we will find it more convenient to work with the following reparameterization: let $\rho(t) = w(t)/CRTT$ be the traffic intensity, and let $s = Ct$, giving

$$\frac{d\rho(s)}{ds} = \frac{1}{wnd^2} - \frac{\rho(s)\rho(s - wnd)p(s - wnd)}{2} \quad (3)$$

where $wnd = CRTT$.

We have made two major approximations. The first is that each flow is equally likely to receive a loss indication. There are versions of this differential equation which do not make this assumption, and which additionally are able to take account of slow start, multiple duplicate ACKs, etc., especially those in [11], [12]. The trouble is that these more refined versions involve partial differential equations, and we have not yet managed to analyse the stability of the resulting dynamical system. However, it is suggested in [1] that "this approximation does not change the fundamental nature of the multiplicative decrease mechanism, and we are able to capture TCP dynamics". This is backed up by

the observation [11] that the partial differential equation and the differential equation give essentially the same results, at least for queues with AQM.

The second major approximation is that packets are being emitted at rate $W(t)/\mathit{RTT}$ at time t, which means we are modelling a rate-based mechanism parameterized by $W(t)$ rather than a genuine window-based mechanism. The technically formidable work of Bain [13] explicitly models window-based control, and suggests an integro-differential equation instead of (2). Happily, it also indicates that (2) gives very similar results, at least in the small buffer regime.

B. Fluid model for the queue

Let the total arrival rate to the queue at time t be $X^N(t) = W^N(t)/\mathit{RTT}$, and let $x(t) = X^N(t)/\mathit{RTT}$. 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. Lindley's recursion gives us an idea of how the queue size $Q^N(t)$ will evolve:

$$Q^N(t+\delta) \approx \Big[Q^N(t) + \delta N x(t) - \delta NC\Big]_0^{B^N} \qquad (4)$$

where $[q]_0^b = \min(\max(q,0),b)$. Depending on how B^N is chosen, this can lead to different queueing 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. The details of different choices for B^N are given in the following sections.

III. Small buffers

In this section we study the choice $B^N = B$. We will treat the traffic flow from each source as a point process, where the points indicate packets. We will further assume that, over short timescales, and conditional on the mean window size $w(t)$, each flow can be treated as independent.

Note that the maximum queueing delay is B/NC. For large N, queueing delay is a negligible part of RTT.

A. Fluid model

Consider first an open-loop queueing system with N flows, in which each flow has mean rate x. The following theory for this system is taken from [14]. The aggregate arrival process converges to a Poisson process[2], 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 $\tilde{A}^N(u) = A^N(t, t+u/N)$ 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 for a queue fed by a Poisson process with arrival rate x and served at constant rate C, in an 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 open-loop 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, as in [16], which also explains how to incorporate AQM. It is also known that $Q^N(t)$ makes excursions of size $O(1)$ in timescale[3] $O(1/N)$. Therefore, in any $O(1)$ time interval, the queue size will repeatedly hit empty and full, i.e. it will 'loose its memory'.

This suggests [5] that *in the closed-loop system* if the mean arrival rate $x(t)$ doesn't change by much in a short interval, then the the loss probability is

$$p(t) = L_B(\rho(t)), \quad \rho(t) = x(t)/C. \qquad (5)$$

B. Equilibrium analysis

We have derived the dynamical system (3) & (5). As we described in the introduction, we will study stability properties in the vicinity of an equilibrium point, i.e. a point (ρ^*, p^*) such that

$$0 = \frac{1}{\mathit{wnd}^2} - \frac{(\rho^*)^2 p^*}{2} \qquad (6)$$

$$p^* = L_B(\rho^*). \qquad (7)$$

Throughout this paper, we will illustrate equilibrium points using a *load-loss graph*. For a given wnd, plot the ρ^* and p^* which solve (6); for a given B, plot the ρ^* and p^* which solve (7). Where these two curves intersect is the equilibrium point (ρ^*, p^*). This is illustrated in Figure 1.

C. Stability analysis

Given wnd and B we find the equilibrium point (ρ^*, p^*); we then calculate whether the system is locally stable around this equilibrium point, as described in the introduction. In Figure 1 we have coloured the unstable equilibrium points grey.

[2]It has been observed [15] that the packet arrival process for Internet traffic is not Poisson. However, as is pointed out in [14], the former work is concerned with long timescales, whereas for systems with buffers which are $O(1)$ it is only the short-timescale traffic characteristics that matter, and these are approximately Poisson.

[3]For intuition, consider an $M_{Nx}/M_{NC}/1$ queue, which is just an $M_x/M_C/1$ queue speeded up by a factor of N. Therefore the $M_{Nx}/M_{NC}/1$ queue hits any given size B in timescale $O(1/N)$.

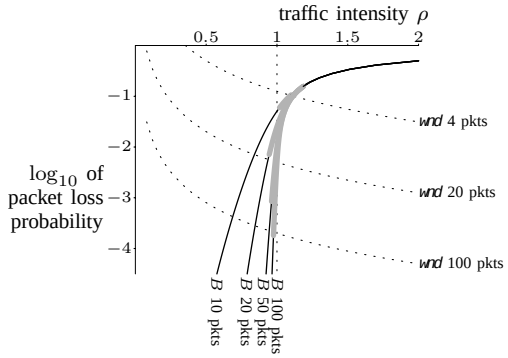

Fig. 1. The TCP throughput curve (6) for a range of window sizes *wnd*, and the loss probability curve (7) for a range of buffer sizes B. For a given *wnd* and B, the intersection of the two curves gives the equilibrium operating point (ρ^*, p^*). Equilibrium points which are locally unstable are coloured grey.

The plot shows the dilemma in choosing buffer size. If we choose a small buffer $B = 10$ packets, the system is stable for almost all *wnd*, but the utilization must be low to achieve *wnd* large. If we choose a larger buffer $B = 100$ packets we can get higher utilization, but the system is stable only when *wnd* > 100 packets (or when *wnd* is very small).

We found qualitatively the same results when we modified $L(\cdot)$ to model AQM schemes including RED.

D. Instability analysis

It is not *a priori* obvious that local instability is bad. It might be that the system is locally unstable, but that oscillations are nevertheless harmless. To investigate this we plot in Figure 2 the amplitude of the oscillations in ρ, for a range of buffer sizes, calculated using the method outlined in the introduction. (The horizontal bar about an equilibrium point (ρ, p) indicate the amplitude.) At $B = 50$ packets, for example, the system is unstable for window sizes less than around 45 packets, but the oscillations are only serious for window sizes of around 5 packets.

To judge just how serious these oscillations are, we take the four scenarios from Figure 2 and show in Figure 3 the goodput attained. At a stable equilibrium point (ρ^*, p^*) the goodput is just $\rho^*(1 - p^*)$; at an unstable point where the traffic intensity oscillates about ρ with amplitude a, the goodput is

$$\frac{1}{2\pi}\int_{\theta=0}^{2\pi} \big(\rho + a\sin\theta\big)\big[1 - L(\rho + a\sin\theta)\big]\, d\theta.$$

The large oscillations when $B = 100$ packets can cause goodput to fall by as much as 50%. We see that the buffer size must be chosen carefully—if it is too small, goodput is low because TCP backs off too much; if it is too high, goodput is low because of oscillations.

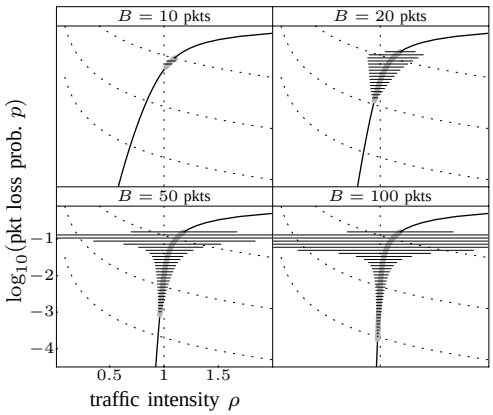

Fig. 2. When the system is unstable, there are oscillations in traffic intensity ρ. The size of the oscillations in ρ about an unstable equilibrium point (ρ^*, p^*) is indicated by a horizontal bar.

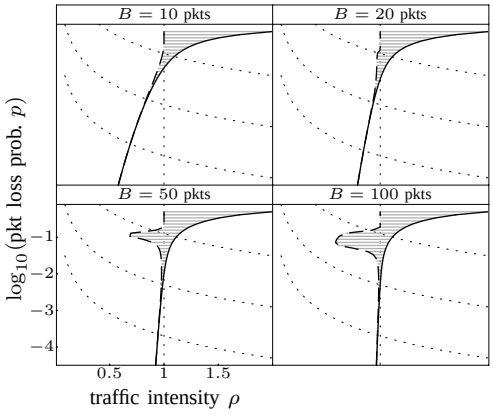

Fig. 3. If there are large oscillations in traffic intensity ρ, goodput drops. Let ρ' be the goodput corresponding to an unstable equilibrium point (ρ^*, p^*). We plot (ρ^*, p^*) in black and (ρ', p^*) dashed (and a horizontal line joining the two points).

IV. Intermediate buffers

In this section we study the choice $B^N = N^\gamma B$ where $0 < \gamma < 1$. (Recall that N is the number of flows, and NC is the service rate.) This buffer sizing regime is a consequence, for example, of the rule suggested in [3], which says that buffer size should be proportional to $NC\mathit{RTT}/\sqrt{N}$ i.e. $\gamma = 1/2$. We will make the same assumptions about the flows as in Section III.

Note that the maximum queueing delay is $N^\gamma B/NC$. For large N, queueing delay is a negligible part of *RTT*.

A. Queueing theory for underload

Consider first an open-loop queueing system with an infinite buffer, serving N flows each with mean rate $x < C$. Model each flow by a point process. According to the global approximation [17, Section 10.3], the queue size

satisfies

$$\log \mathbb{P}\big(Q^N(0) > N^\gamma B\big) \approx -\inf_{t\geq 0}\sup_{\theta\geq 0} \theta(N^\gamma B + NCt) - N\Lambda_t(\theta)$$

where $\Lambda_t(\theta) = \log \mathbb{E}\exp\theta A(t)$, and $A(t)$ is the number of packets from a typical flow in an interval of duration t. The interpretation of t is that it is the most likely timescale for the buffer to go from empty to full. Let $s = N^{(1-\gamma)}t$. Then

$$\frac{1}{N^\gamma}\log \mathbb{P}\big(Q^N(0) > N^\gamma B\big) \approx -\inf_{s\geq 0}\sup_{\theta\geq 0} \theta(B + Ct) - \frac{s}{M}\Lambda_M(\theta)$$

where $M = N^{-(1-\gamma)}s$. Suppose that packets from a single flow cannot be closer together than ε for some $\varepsilon > 0$. Then, whenever $M < \varepsilon$, $A(M) = 0$ with probability xM and 1 with probability $1 - xM$, from which we can deduce that $\Lambda_M(\theta)/M \to x(e^\theta - 1)$ as $N \to \infty$. Thus

$$\begin{aligned}\frac{1}{N^\gamma}\log \mathbb{P}\big(Q^N(0) > N^\gamma B\big) &\approx -\inf_{s\geq 0}\sup_{\theta\geq 0}\Big[\theta(B + Ct) - sx(e^\theta - 1)\Big] \\ &= -B\sup\big\{\theta > 0 : x(e^\theta - 1) < \theta C\big\}. \qquad (8)\end{aligned}$$

The last equality is from [17, Lemma 1.7], which also shows that the optimal s satisfies $0 < \hat{s} < \infty$, and the optimal θ satisfies $0 < \hat{\theta} < \infty$. Furthermore, it is reasonable to believe that (8) also gives the asymptotic probability of overflow for a finite buffer system; see [17, Chapter 6] for an outline of why.

That theory also says that the timescale of overflow for this open-loop system is $t = N^{-(1-\gamma)}\hat{s}$. Since this becomes smaller and smaller as N increases, and since (2) suggests that TCP's arrival rate only changes by a small amount $\delta x'(t)$ in a short interval $(t, t+\delta)$, it is reasonable to believe that *in the closed-loop system* the loss probability experienced by a packet arriving at time t is roughly $\exp(-N^\gamma B\hat{\theta})$. In particular, whenever $x(t) < C$, the packet loss probability tends to zero as $N \to \infty$. An AQM scheme would not change this.

B. Queueing theory for overload

Consider now an open-loop queueing system with a finite buffer $N^\gamma B$, serving N flows each with mean rate $x > C$. Let $R^N(t) = N^\gamma B - Q^N(t)$ be the amount of free space in the queue. This evolves as follows: it increases at a constant rate NC (up to a maximum of $N^\gamma B$), and it decreases by 1 whenever there is a packet arrival (as long as it can do so and remain non-negative), and these arrivals occur at rate Nx. So R^N is like a queue with traffic intensity $NC/Nx < 1$. The theory in Section IV-A suggests that the probability that R^N overflows decays like $\exp(-N^\gamma B\hat{\varphi})$ for some $\hat{\varphi} > 0$, so it is reasonable to approximate $R^N(t)$ by $\bar{R}^N(t)$ which has the same dynamics but no upper limit. We wish to know the packet loss probability for the original queue Q^N, i.e. the probability that $\bar{R}^N(t) < 1$. For this purpose we can replace $\bar{R}^N(t)$ by $\tilde{R}^N(t)$, in which a complete unit of work arrives every $1/(NC)$ time units, rather than have work arrive continuously at rate NC. Now apply Little's law to the head of the queue $\tilde{R}^N(t)$: the expected number of units of work present at the head of the queue is NC/Nx. Since the head of the queue can hold at most one complete unit of work, the probability that there is one complete unit of work there is C/x. But this is exactly the probability that a service event finds some work to serve, which in the original system is the probability that an incoming packet is not dropped. Therefore the packet loss probability for the original system is $1 - C/x$.

For this open loop queueing system, fluctuations of size $O(1)$ in the queue $R^N(t)$ occur over timescales which are $O(1/N)$, just as in Section III. Therefore it is reasonable to use this same formula $p(t) = 1 - C/x(t)$ for the packet loss probability in a closed-loop system.

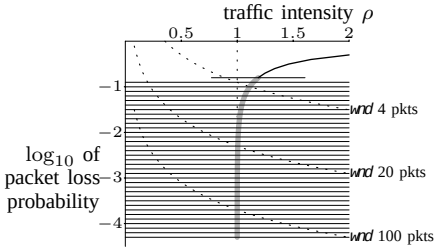

Fig. 4. Oscillations in traffic intensity ρ about an unstable equilibrium point (ρ^*, p^*), indicated by horizontal bars, for the intermediate buffer regime.

C. Stability/instability analysis

We have derived a fluid model of exactly the same form as that in Section III, but where the loss function is $p(t) = L(\rho(t)) = [1 - 1/\rho(t)]^+$. This loss function was introduced in [18], though with different reasoning. The fluid model can be analysed just as before—Figure 4 shows that things go seriously wrong for window sizes larger than a few packets. We might have guessed this from Figure 2, which shows that larger buffers result in greater instability.

Interestingly, the simulations in [3] show a different picture. They show small oscillations in traffic rate,

synchronized with large oscillations in queue size. As we will see in Section V, this is a symptom of a system with a reasonable amount of buffer space *per flow*, not the negligible amount of buffer space per flow we have analysed in this section. The simulations described in that work did not have enough flows for the problems we have described here to become apparent. They had several hundred flows; our rule of thumb in Section VII suggests the problems will start to appear at 5,000 flows or so.

V. Large buffers with AQM

In this section we study the choice $B^N = NB$, and we assume there is an AQM scheme in operation.

A. The fluid model

Lindley's recursion (4) suggests that the scaled queue size $q(t) = Q^N(t)/N$ satisfies

$$\frac{dq(t)}{dt} = \Big[x(t)\big(1-p(t)\big) - C\Big]^{+[q(t)=0],-[q(t)=B]} \tag{9}$$

where the notation for dq/dt means that if $q(t) = 0$ we take the positive part of the term in brackets $[\cdot]$, and if $q(t) = B$ we take the negative part. This is the fluid model suggested by [1].

The AQM scheme specifies how $p(t)$ depends on $q(t)$. We will look at GentleRED, choosing parameters according to the guidelines in [19]: $p_{max} = 10\%$, and $min_{th} = O(1)$ packets which translates to 0 in the $1/N$ scaling. RED also keeps an exponentially-weighted moving average of queue size, in order that it should detect persistent changes in rate rather than momentary bursts. Since momentary bursts lead to $O(1)$ fluctuations in queue size (see Section III-A), whereas $q(t)$ only reflects $O(N)$ changes in queue size, there is no need for this sort of smoothing in large multiplexers. Therefore we will take the loss probability to be an function of the instantaneous queue size. Thus we might as well set $max_{th} = B/2$. This leads to $p(t) = L_B(q(t))$ where

$$L_B(q) = \begin{cases} 2qp_{max}/B & \text{if } q \le B/2 \\ p_{max} + (1-p_{max})(2q/B - 1) & \text{else.} \end{cases}$$

In fact, to make our calculations more realistic when the oscillations are moderate-sized, we prefer to approximate $L_B(\cdot)$ by a smoother loss function

$$\tilde{L}_B(q) = \frac{a^{q/B} - 1}{a - 1}$$

where a is chosen so that $\tilde{L}_B(B/2) = L_B(B/2) = p_{max}$. This is useful because our theory is based on third-order expansions of $L_B(q^* + \delta)$ about an equilibrium point q^*, and for a piecewise-linear function like L_B such an expansion doesn't reveal what happens for moderately large δ.

Note that the round trip time may vary, depending on the queue size: the round trip time experienced by packets/ACKs which arrive at the source at time t is $RTT_t = t - \tilde{t}$ where $t = \tilde{t} + q(\tilde{t}) + PT$ and PT is the propagation delay. So $\tilde{t}$ is the time that a packet must leave the source if an ACK is to return at time t.

A very similar model was studied by [20]. They considered a rate model of TCP in which the window size $W(t)$ for each connection increases steadily at rate $1/RTT_t$ and decreases according to a Poisson process of intensity $W(t - RTT_t)p(t - RTT_t)/RTT_t$. They prove that, as $N \to \infty$, this stochastic system converges to a fluid model, similar to (2) but involving partial differential equations. As remarked in Section II-A, the PDE model has very similar behaviour to (2).

B. Equilibrium analysis

An equilibrium point of (2) & (9) satisfies $w^* = \sqrt{2/p^*}$, $x^*(1-p^*) = C$, $p^* = L_B(q^*)$, $RTT^* = PT + q^*/C$. As usual we prefer to express this in terms of $\rho^* = x^*/C$ and p^*: so the equilibrium point is

$$\rho^* = \sqrt{2/p^*}/CRTT^* \tag{10}$$

$$p^* = 1 - 1/\rho^* \tag{11}$$

and the subsidiary quantities are

$$q^* = L_B^{-1}(p^*), \quad CPT = CRTT^* - q^*. \tag{12}$$

Note that (10) is the usual equation for TCP equilibrium (6), and (11) is the loss function we used in Section IV-C.

As usual we will plot load-loss graphs of ρ against p, although now that $CRTT^*$ depends on the equilibrium point we prefer to illustrate (10) differently: we assume the buffer sizing rule $B = \beta CPT$ (a generalization of the bandwidth-delay product rule), we use (10) & (12) to obtain a relationship between ρ^* and p^* parameterized by $flt = CPT$, and we plot several dotted lines corresponding to $flt = 4, 20, 100$ packets. See Figure 5 for an illustration.

C. Stability/instability analysis

It seems very difficult to analyse the stability of (3) & (9) taking into account the time-dependence of RTT_t. We are able to analyse delay-differential equations in which the delay is constant, and so we will approximate (3) & (9) by replacing $wnd_t = CRTT_t$ by $wnd^* = CRTT^*$, where RTT^* is the equilibrium round trip time, as calculated in

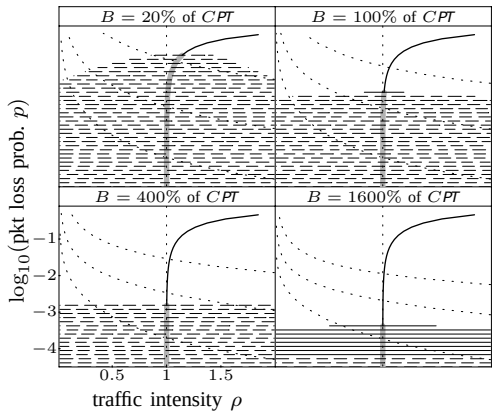

Fig. 5. Oscillations in ρ about an unstable equilibrium point (ρ^*, p^*) are indicated by horizontal bars. The bars are dashed if the oscillations in q hit both $q = 0$ and $q = B$, i.e. if the buffer size fluctuates wildly. In order to prevent such oscillations for $\mathit{flt} = 20$ packets or higher, one needs at least a buffer at least four times the bandwidth-delay product.

Section V-B. We will also reparameterize (9) in terms of $s = Ct$ to give

$$\frac{dq(s)}{ds} = \Big[\rho(s)\big(1 - p(s)\big) - 1\Big]^{+[q(s)=0],-[q(s)=B]}.$$

Figure 5 illustrates the stability of this system, assuming the buffer sizing rule $B = \beta C\mathit{PT}$, for a range of values of β. The black/grey line illustrates (11), with points coloured grey if the system is locally unstable. At such points, we calculate the oscillations in $\rho(t)$ and $q(t)$, and we indicate the amplitude of the former by means of a horizontal bar; the bar is dashed if the oscillations in $q(t)$ hit both $q = 0$ and $q = B$ (by which point one should not place too much faith in our estimates, based as they are on third-order expansions about equilibrium; nonetheless this is a sign of worryingly large fluctuations in queue size).

VI. Large buffers with droptail

In this section we consider the choice $B^N = NB$, for a queue with no AQM.

A. Fluid model

At an equilibrium point there must be some non-zero loss probability, which requires $\rho^* > 1$. Suppose there is a small fluctuation in ρ, with $\rho(t) > 1$, about the equilibrium. As in Section IV-B, the free-space process makes excursions from 0 of size $O(1)$ over timescales which are $O(1/N)$, and so the loss probability is $p(t) = 1 - 1/\rho(t)$. These excursions are negligible compared to the total buffer size, so the scaled queue size remains $q(t) = B$.

Suppose now that $\rho(t)$ drops < 1. Then the queue size starts to decrease, and (4) suggests

$$\frac{dq(t)}{dt} = \Big[C\big(\rho(t) - 1\big)\Big]^{+[q(t)=0]}.$$

Indeed this will remain true until $q(t)$ hits B again, when $\rho(t) \geq 1$, bringing us back to the former situation.

Reparameterizing in terms of $s = Ct$, we obtain

$$\frac{dq(s)}{ds} = \big[\rho(t) - 1\big]^{+[q(s)=0],-[q(s)=B]}$$
$$p(s) = \big(1 - 1/\rho(s)\big)1_{q(s)=B}.$$

B. Stability/instability analysis

When $\rho(t) > 1$ and $q(t) = B$, this is exactly like the intermediate-buffer system in Section IV; if the analysis there indicates either stability or small oscillations with $\rho(t) > 1$ then we will see exactly the same in the large-buffer system. Note though from Figure 4 that the intermediate-buffer system hardly ever has small oscillations, that it is either stable or wildly unstable.

When the intermediate-buffer system is wildly unstable, it has oscillations in which $\rho(t)$ drops < 1. The impact of large oscillations on the large-buffer system is to lag the response, thereby making the instability more pronounced. On the other hand goodput is improved with large buffers[4], since there is idleness only when $q(t) = 0$, and this requires that $\rho(t) < 1$ for an extended period.

VII. Limit philosophy

We have presented three different buffer-sizing regimes, and analysed their performance in the limit as the number of flows increases. We now describe a rule of thumb for judging how to apply those results to a given system with N flows, total service rate C, and total buffer size B. Also let $\mathit{wnd} = C\mathit{RTT}/N$. The idea is to calculate the *tipping time*, the time it takes to tip from underloaded to overloaded or vice versa.

First, calculate the equilibrium traffic intensity ρ^*, using (6) & (7). (The latter equation reduces to (11) when B is large.) Consider first $\rho^* < 1$. Suppose that there have not been any drops for a short time, so that mean transmission rate is increasing at rate $1/\mathit{RTT}^2$ and thus traffic intensity is increasing at rate $N/C\mathit{RTT}^2$. It takes time $(1 - \rho^*)\mathit{wnd}\mathit{RTT}$ until $\rho = 1$. Consider next $\rho^* > 1$. On the grounds that in equilibrium the increase rate and decrease rate balance out, suppose that traffic intensity is decreasing at rate $N/C\mathit{RTT}^2$. Then it takes

[4]To estimate goodput, one needs to estimate how long it takes for q to hit 0, based on the oscillations in ρ. This goes beyond the scope of the dynamical systems theory in this paper.

time $(\rho^* - 1)\mathit{wndRTT}$ until $\rho = 1$. In both cases, the first component of tipping time is

$$\tau_1 = |1 - \rho^*|\mathit{wndRTT}.$$

Now, for the case $\rho^* < 1$, we calculate the further time it takes for the buffer to fill completely (and for the queue size distribution to shift from bouncing around empty to bouncing around full). Suppose that the mean transmission rate continues to increase at the same rate as before. After a short time t of this, the total arrival rate is $C + tN/\mathit{RTT}^2$, so the time it takes to fill the buffer is $\mathit{RTT}\sqrt{2B/N}$. For the case $\rho^* > 1$ we want instead the time it takes for the buffer to drain completely, which turns out to be exactly the same. In terms of the maximum queueing delay $D = B/C$, the second component of tipping time is in both cases

$$\tau_2 = \sqrt{2D\mathit{wndRTT}}.$$

If the total tipping time $\tau_1 + \tau_2$ is much less than RTT then this counts as a drastic 'bang-bang' response, and so the system is likely to be unstable. Stability can come either from making τ_1 large (i.e. keeping equilibrium utilization ρ^* low) or from making τ_2 large (i.e. making sure the queueing delay D is large).

The simulations in [3] had N ranging from 50 to 500 flows, propagation delays PT ranging from 25ms to 300ms, buffer sizes from 50 to 350 packets, and service rate around 100 pkts/ms. Using midpoints of these ranges, $\mathit{RTT} \approx 165$ms, $\mathit{wnd} \approx 60$ packets, $\rho^* \approx 99.4\%$, $\tau 1 \approx 63$ms, $\tau_2 \approx 196$ms. Thus N is not large enough to see the effects we have predicted. If we scale the system up to have M flows, scaling the service rate in proportion to M and the buffer size in proportion to $\sqrt{M}$, then τ_2 scales like $M^{-1/4}$; at $M = 5000$ flows we get $\tau_2 = 60$ms, which we suspect is small enough to cause serious instability.

VIII. Conclusion

In Section VI we studied the bandwidth-delay product rule for sizing buffers (which we have called the large buffer regime), without any AQM. It is the most unstable of all the regimes we studied.

The intermediate buffer regime, as suggested by [3] and studied in Section IV, has marginally better stability (see Figure 4) though the goodput may be slightly lower. Adding AQM has marginal impact.

To improve things, use large buffers with AQM. This has much better stability, at least for window sizes up to around 20 packets (see Figure 5).

Switching to small buffers (with or without AQM) leads to improved stability even for very large window sizes, though the buffer size and/or AQM must be chosen carefully to avoid instability[5] at window sizes of around 5 packets (see Figure 2).

If the bandwidth-delay product rule becomes technologically impractical, then buffers should be slashed ruthlessly. Indeed, we recommend slashing buffers ruthlessly right now, since this should improve goodput; it would also cut costs and reduce delay and jitter.

References

[1] V. Misra, W.-B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," *ACM SIGCOMM CCR*, 2000.

[2] R. Srikant, *The Mathematics of Internet Congestion Control*, Birkhäuser, 2004.

[3] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," in *Proc. ACM SIGCOMM*, 2004.

[4] F. P. Kelly, "Models for a self-managed Internet," *Phil. trans. Royal Society*, vol. A358, 2000.

[5] S. Deb and R. Srikant, "Rate-based versus queue-based models of congestion control," in *ACM Sigmetrics*, 2004.

[6] H. C. Morris, "A perturbative approach to periodic solutions of delay-differential equations," *J. Inst. Maths Applics*, 1976.

[7] G. Raina, *Congestion control for the Internet: stability and bifurcation analysis*, Ph.D. thesis, Cambridge University, 2004.

[8] F. Kelly, "Fairness and stability of end-to-end congestion control," *European Journal of Control*, 2003.

[9] G. Vinnicombe, "On the stability of networks operating TCP-like congestion control," in *Proc. IFAC*, 2002.

[10] H. Han, C. V. Hollot, Y. Chait, and V. Misra, "TCP networks stabilized by buffer-based AQMs," in *IEEE Infocom*, 2004.

[11] M. Ajmone Marsan, M. Gatetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello, "Using partial differential equations to model TCP mice and elephants in large IP networks," in *Proc. IEEE Infocom*, 2004.

[12] F. Baccelli, D. R. McDonald, and J. Reynier, "A mean-field model for multiple TCP connections through a buffer implementing RED," *Performance Evaluation*, 2002.

[13] A. Bain, *Fluid limits for congestion control in networks*, Ph.D. thesis, University of Cambridge, 2003.

[14] J. Cao and K. Ramanan, "A Poisson limit for buffer overflow probabilities," in *Proc. IEEE Infocom*, 2002.

[15] V. Paxson and S. Floyd, "Wide-area traffic: the failure of Poisson modeling," *ACM SIGCOMM CCR*, 1994.

[16] P. Kuusela, P. Lassila, J. Virtamo, and P. Key, "Modeling RED with idealized TCP sources," in *Proc. IFIP ATM & IP*, 2001.

[17] A. J. Ganesh, N. O'Connell, and D. J. Wischik, *Big Queues*, Lecture Notes in Mathematics. Springer, 2004.

[18] S. Kunniyur and R. Srikant, "End-to-end congestion control schemes: utility functions, random losses and ECN marks," in *IEEE Infocom*, 2000.

[19] Sally Floyd, "RED: Discussions of setting parameters," Oct 2004, www.icir.org/floyd/REDparameters.txt.

[20] D. R. McDonald and J. Reynier, "Mean field convergence of a rate model of multiple TCP connections through a buffer implementing RED," in press, 2004.

[21] T. Kelly, "On engineering a stable and scalable TCP variant," Tech. Rep. CUED/F-INFENG/TR.435, Cambridge University Engineering Department, 2002.

[5] We experimented with Scalable TCP [9], [21] and found that it eliminates the instabilities.