The Window Distribution of Multiple TCPs with Random Loss Queues

Archan Misra
archan@research.telcordia.com

Teunis Ott
tjo@research.telcordia.com

John Baras
baras@isr.umd.edu

Abstract—**Two approximate techniques for analyzing the window size distribution of TCP flows sharing a RED-like bottleneck queue are presented. Both methods presented first use a fixed point algorithm to obtain the mean window sizes of the flows, and the mean queue length in the bottleneck buffer. The simpler of the two methods then uses the 'square root formula' for TCP; the other method is more complicated. More often than not, the simpler method is slightly more accurate; this is probably due to the fact that window sizes of the different flows are negatively correlated.**

Keywords—**TCP, multiple, distribution, RED, queues.**

I. Introduction

In this paper, we consider the case where multiple *persistent* TCP flows, which are each performing *idealized* congestion avoidance, interact with a buffer that implements randomized packet drops as a queue management algorithm. Our objective is to determine the stationary congestion window distribution of each of the TCP flows when the router port implements algorithms like RED (Random Early Detection) or ERD (Early Random Drop). We first present an analytical technique, resulting in a fixed-point iteration scheme, to obtain the 'mean' values of the queue occupancy and the individual TCP windows. Armed with this estimate of the *means*, we then evaluate *two* techniques to derive approximations to the window *distribution* of each individual TCP connection. In the simpler of the two approaches, we assume that the window evolution of a specific TCP flow is governed by a *constant* loss probability; this probability is equal to the packet dropping probability of the randomized dropping algorithm when the queue occupancy remains at its 'mean' value. The individual distributions, in this case, is computed using the analysis presented in [2]. In the other approach, we relate the window size of the flow to the queue occupancy through a simple linear relationship and hence, define a *variable* packet loss probability that is a function of the connection's window size. The individual distributions, in this case, are derived using a numerical technique presented in [3], which considers the case of a *single* TCP flow subject to *variable state-dependent* packet loss.

We have conducted a wide range of simulation experiments to verify the applicability of our approaches. One important phenomenon we have observed is that, for queues performing randomized drop without any memory, the window sizes of connections are not truly independent (or uncorrelated), but are, in fact, *negatively* correlated. As a consequence of this behavior, the queue size (and consequently the loss probability) tends to stay relatively constant over a much wider range of window size (of a particular connection) than would be expected under a truly independent model. We have therefore observed the simpler 'square-root'-based approach (which assumes a constant loss probability) to usually provide a relatively better prediction of simulated distributions than the relatively more complex 'variable probability' approach. Both approaches, however, provide reasonably accurate estimates of the simulated values, over the entire range of our simulations; the estimates are more accurate at lower loss rates. In a later paper, we propose to examine in greater detail how the presence of memory in the random dropping function (as, for example, in RED where an averaged queue occupancy is used) or modifications that change the pattern of packet drops (for example, making the inter-drop gap uniformly distributed, as in RED) affect the negative correlation and the accuracy of our estimates.

A TCP flow implementing idealized congestion avoidance [1] will increment its window by 1 segment every round trip time and will *instantaneously* halve it on detecting congestion (via packet losses). Such a model of TCP window evolution ignores transient phenomena like fast recovery [9], fast retransmit and slow start. Mathematically speaking, the window evolution of the i^{th} TCP connection is approximated by a stochastic process $(W_i^n)_{n=1}^{\infty}$, where W_i^n refers to the congestion window of connection i just after the receipt of the n^{th} *good* acknowledgement packet (one that advances the left marker of TCP's sliding window). By disregarding timeouts and fast recovery, we obtain a *discrete-time* Markovian process such that

$$P\{W_i^{n+1} = w + \frac{1}{w} | W_i^n = w\} = 1 - p_i(w) \qquad (1.1)$$

$$P\{W_i^{n+1} = \frac{w}{2} | W_i^n = w\} = p_i(w). \qquad (1.2)$$

where $p_i(w)$ is the packet loss probability when the congestion window of connection i is w [1].

[2] calculates the stationary window distribution for a single TCP flow subject to constant packet drop probability. [3]

Archan Misra and Teunis Ott are with Telcordia Technologies (formerly Bellcore), 445 South Street, Morristown, NJ 07960. John Baras is with the Center for Satellite and Hybrid Communication Networks at the University of Maryland, College Park, MD 20742.

[1] While *cwnd* in actual TCP implementations is expressed in bytes and is consequently integer-valued, we assume that, in equations (1.1) and (1.2), W is real-valued and is expressed in MSSs. The congestion window in the rest of this paper is assumed to be real-valued. We will explicitly mention the situations where the congestion window is expressed in bytes.

extends the technique to compute the congestion window distribution for a single TCP flow when the packet loss probability is variable but depends only on the flow's instantaneous window size i.e., the model of equations (1.1) and (1.2); this analysis was used to estimate the window distribution of a single TCP flow interacting with a router port performing RED [8] or ERD [12]. In this paper, when multiple TCP flows interact with a single queue, the exact loss probability for a specific connection depends not just on the window size of that connection, but also on the instantaneous window sizes of all the other connections. Since an exact description of this process involves a multi-dimensional Markovian model which is analytically intractable [2], we approximate the problem by first deriving the mean window sizes (of each TCP connection) and queue occupancy (for the random drop queue) using a zero-drift condition derived from equations (1.1) and (1.2). In the constant probability approach (which we shall call the 'square-root technique'), we assume that the loss probability for any packet is governed by a constant value determined by the sum of the mean window sizes, and hence, obtain the distribution using [2]. In the variable probability approach (which we also call the 'perturbation approach'), we assume that the sum of the instantaneous window sizes of all the other flows can be represented (with only moderate error) by the sum of their means. This assumption reduces the queue occupancy (and hence, the loss probability function) to a simple function of the flow's window size, and can be analyzed as in [3].

A. Related Work and Model Applicability

The behavior of the TCP congestion window in the congestion avoidance regime has been extensively analyzed under the assumption that the loss probability and round-trip times are constant. Derivations of the 'square-root formula', which states that the mean window of a TCP connection is inversely proportional to the square-root of the loss probability, are provided in varying degrees of rigorousness in [6], [11] and [7]. A more elaborate analysis, which derives the detailed distribution under the assumption of constant loss, is presented in [2]. More elaborate models for TCP that incorporate the effect of timeouts and fast recovery transients are presented in [5] and [4]; these essentially show that timeouts and fast recovery transients become important for current TCP versions when the loss probability is relatively large and cause the throughput to become proportional to $\frac{1}{p}$ in this regime. [3] derives the stationary distribution when the loss probability is *not constant* but a function of the window size; the technique is then applied to analyze the window distribution of a single TCP flow interacting with a RED (Random Early Detection) or ERD (Early Random Drop) queue.

[2] An accurate model of the window evolution process for N TCP connections would require an N – *dimensional* Markov model, where the state space would be a N-dimensional vector consisting of the window sizes of each individual connection. The transition probabilities between states would depend on the state of the entire system (the instantaneous windows of each connection), making the definition of useful scalings impossible.

As stated earlier, our model does not account for TCP transients like slow start, timeouts and fast recovery. We believe that the disproportionate impact of these on current versions of TCP (like Tahoe and Reno) at moderately high loss probabilities is due largely to the coarse granularity of currently implemented TCP timers and the the fact that loss recovery mechanisms (like fast retransmits and timeouts) are combined with congestion control. Accordingly, this analysis is accurate for current TCP versions when the loss probabilities are small and the delay-bandwidth product large enough ($\approx$ 10 segments and above) to ensure that timeouts are relatively rare events. As mechanisms to better separate loss recovery from congestion avoidance (such as TCP SACK) or decrease the frequency of timeouts (such as the improvements in TCP New Reno) become commonplace and as finer-grained timers are adopted, our analysis should hold over larger variations in performance parameters.

II. Mathematical Model and Problem Approach

The TCP connections are *persistent* (sending infinite-sized data files), with the congestion window acting as the only constraint on the injection of new packets by the sender. We assume that the connection never times out, that the data is always sent in equal-sized segments (although segment sizes could vary between connections) and that acknowledgements are never lost. For the purpose of presentation, we assume that the receiver acknowledges every received packet separately (delayed acknowledgements are not enabled); the corrections for delayed acknowledgement are listed in Appendix C. As described in [3], the stationary distribution of each connection is computed in what we call *ack time*[3], which is a positive integer valued variable that increments by 1 only when a *good* acknowledgement arrives at the source.

Let N be the number of concurrent TCP connections under consideration. The i^{th} flow of the set, denoted by TCP_i, has a maximum segment size (MSS) of M_i bytes and a nominal round trip time (excluding the queuing delay at the buffer) of RTT_i seconds. Let W_i denote the window size of the i^{th} connection; as different TCP flows have different packet sizes, this is measured in *bytes* unless explicitly stated otherwise. Note that while the process model (equations (1.1) and (1.2)) represents the window state in segments, the analysis here represents the window sizes in bytes; once the stationary distribution (in segments) has been determined, expressing the distribution in bytes is straight-forward.

The queue is assumed to perform *random* packet drops i.e., the loss probability of a packet is conditionally independent of past and future losses. The loss probability is modeled to be dependent on the *instantaneous* queue occupancy. We let the service rate of the queue be C bytes/sec. In general, let Q be the buffer occupancy of the random drop queue and Q_i (in bytes) be the amount of traffic from connection i that

[3] The 'ack time' is different from 'clock time' in that the ack time advances only when a good acknowledgement arrives at the sender. This will be linearly related to the progress of clock time only if the window sizes and the round trip times are both constant.

is buffered in the queue (so that $\sum_{i=1}^{N} Q_i = Q$). The drop function is denoted by $p(Q)$. For the experimental results in this paper, we used the *linear* drop model, so that $p(Q)$ has the following behavior:

$$\begin{aligned} p(Q) &= 0 && \forall\ Q < min_{th} \\ &= p_{max} && \forall\ Q > max_{th} \\ &= p_{max} * \frac{Q - max_{th}}{max_{th} - min_{th}} && \forall\ min_{th} \leq Q \leq max_{th} \end{aligned}$$

where, as per standard notation, max_{th} and min_{th} are the maximum and minimum drop thresholds (in bytes) and p_{max} is the maximum packet drop probability. Other forms of the drop function can also be used in the subsequent analysis; our numerical technique for determining the 'mean' only requires that $p(Q)$ be non-decreasing in Q, which is true for all sensible drop functions.

Although our analysis is primarily focussed on algorithms that do not maintain flow-specific state (and do not distinguish between flows), a slight generalization, which allows the actual packet drop probability to be flow-dependent, is possible. To that extent, we suppose that the loss probability for a packet of flow i, which arrives when the queue occupancy is Q, is given by the function $p_i(Q)$. $p_i(Q)$ is related to our afore-mentioned drop function $p(Q)$ by the expression:

$$p_i(Q) = c_i^2 p(Q) \tag{2.1}$$

where the c_i are arbitrary non-zero constants. Our model thus permits the loss function for different connections to be *scalar multiples* of one another; the scalar values are represented as c_i^2 instead of c_i for future notational convenience.

This scalar model permits us, for example, to capture the *byte-mode* of operation of RED where the probability of a *packet* drop is proportional to the size of the packet (by setting $c_i^2 = M_i$)[4]. Also, for convenience, we shall use $\bar{p}_i(W)$ to represent the (as yet unknown) relationship between the packet drop probability of TCP_i and its window size W. The reader may note that packet drops in RED, unlike our reference model, are not truly conditionally independent; a simple correction for our model in such a situation is discussed in Appendix B.

We first use a drift-based argument to determine the center of the queue occupancy, denoted by Q^*, and the centers of the *cwnd*-s of the individual connections, denoted by W_i^*, $i = \{1, \ldots, N\}$. After obtaining the center of the queue occupancy, we focus on the distribution of the individual connections and evaluate the relative merits of two different techniques. Under both approaches, when considering the distribution of the i^{th} connection, we assume that the window sizes of all the other connections are constant and equal to their 'mean' value. For the *perturbation* approach, we then relate the queue occupancy to the instantaneous window W_i and thus define a packet loss probability that is a function of W_i alone. The resulting variable-loss model is then solved using the analysis in [3]. For the *constant-loss/square-root* approach, we assume that all packets encounter a packet loss probability given by $p_i(Q^*)$. The window distribution for a process subject to this constant packet loss probability is derived using the analysis in [2]. Since the two approaches employed in this paper require the use of mathematical techniques and expressions presented in ([2]) and ([3]), we briefly discuss the principal contributions of each to make this article largely self-contained. For details, refer to the papers themselves.

The first paper [2] considers the stationary window distribution of a single persistent TCP flow $(W^n)_{n=1}^{\infty}$ (implementing idealized congestion avoidance) when each TCP segment is subject to a constant loss probability p. The derivation consists of first defining an associated process $W(t)$, such that, $W(t) = \sqrt{p} W_{\lfloor \frac{n}{p} \rfloor}$ (uniform scaling in both the time and space axes). As $p \downarrow 0$, the process $W(t)$ becomes a continuous-time, fluid-flow process with the following description: **There is a Poisson process with intensity 1, whose realizations represent the packet loss events. In between the points of the Poisson process, W evolves according to the differential equation**

$$\frac{dW}{dt} = \frac{1}{W}. \tag{2.2}$$

At a point τ of the Poisson process, the window W of the fluid process behaves as

$$W(\tau^+) = \frac{1}{2} W(\tau^-) \tag{2.3}$$

The paper then defines an associated process $Z(t)$, given by $Z(t) = \frac{W(t)^2}{2}$. It is then shown that the process $Z(t)$ also can be defined in terms of an associated constant-rate Poisson process: in between the points of this Poisson process, $Z(t)$ has a constant rate of evolution ($\frac{dZ}{dt} = 1$), and at a point of the Poisson process, $Z(t^+) = \frac{1}{4} Z(t^-)$. The random variable corresponding to the stationary distribution of $Z(t)$ can then be shown to have the form $Z = \sum_{k=0}^{\infty} (\frac{1}{4})^k E_k$ where $(E_k)_0^{\infty}$ are iid and are *exponentially distributed with mean 1*. Using this analysis and the relation between $W(t)$ and $Z(t)$, we can then show that the stationary distribution of $W(t)$ is given by

$$P\{W > w\} = \sum_{k=0}^{\infty} R_k(\frac{1}{4}) e^{-\frac{4^k w^2}{2}} \tag{2.4}$$

where $R_k(x) = \frac{(-1)^k x^{\frac{1}{2}k(k+1)}}{L(x)(1-x)(1-x^2)\ldots(1-c^k)}$ and $L(x) = \prod_{k=1}^{\infty}(1 - x^k)$. The distribution of the TCP process is then obtained by correcting for the space rescaling (using the relation $F_{TCP}(x) = F_W(\sqrt{p}x)$). The above infinite series converge very rapidly and enable us to very efficiently determine

[4] Our 'scalar-multiple' model of flow-dependent drop probabilities can capture a much richer set of random drop settings than apparent at first glance. For example, it can represent a setting of Weighted RED where the different classes have the same min_{th} and max_{th} thresholds but only different max_ps. We do not explore the validation of such settings further in this paper.

the distribution for the process W. The constant loss-based approach presented in this paper assumes that each packet of every TCP flow observes a constant packet drop probability $p_i(Q^*)$. Under this assumption, once the mean occupancy, Q^*, in the presence of multiple TCP flows is determined, we can use the above expressions to compute the distribution of each TCP flow separately.

The second paper ([3]) computes the window distribution of a *single* persistent TCP flow when the packet drop probability is *not constant, but is rather a function of the instantaneous window size.* In this case, the TCP process $(W^n)_{n=1}^{\infty}$ is subject to packet losses that can be represented as $p(W)$ (as opposed to a constant value p). As with [2], a re-scaled process $W(t)$ is derived by scaling both the time and space axes of the process W^n. If we proceed as in the constant loss case and perform a uniform scaling of the time axis, we will see that the rate of the associated Poisson process, in this case, will not be constant but will become state-dependent, making the analysis much more difficult. Accordingly, we had to resort to a time-dependent rescaling of the time axis, such that the associated Poisson process would again have a constant rate. The time axis t (which we call *subjective time* in this case) is obtained through an invertible (but state-dependent) mapping,

$$\Delta t = p(W^n)\Delta n \tag{2.5}$$

such that subjective time increases at a state-dependent rate: an increase of 1 in the ack time index ($\Delta n = 1$) corresponds to an increase in $p(W^n)$ in the subjective time index. The space scaling is as before and is given by $W(t) = \sqrt{p_{max}}W^n$. We can then prove, as in [3], that as $p_{max} \downarrow 0$, the limiting process $W(t)$ becomes one that can be characterized by a differential equation-based evolution (in W) between the instants of realization of a Poisson process of intensity 1. Unlike the case of a constant loss model, the differential equation has a slightly more complex denominator,

$$\frac{dW}{dt} = \frac{p_{max}}{p(\frac{W}{\sqrt{p_{max}}})W}. \tag{2.6}$$

As before, at the instants of events of the Poisson process, $W(t_+) = \frac{W(t_-)}{2}$. Due to this relatively complex nature, we are unable to derive an analytical expression for the stationary distribution of this limiting continuous-time process $W(t)$, and hence, need to solve it using numerical techniques. As a first step, we show, by relating the possible ways in which the process state can evolve over infinitesimal time intervals, that the stationary cumulative distribution (in subjective time), F_s, satisfies the differential equation:

$$\frac{dF_s(x)}{dx} = q(x)\{F_s(2x) - F_s(x)\} \tag{2.7}$$

where $q(x) = \frac{p(\frac{x}{\sqrt{p_{max}}})x}{p_{max}}$. We first transform equation (2.7) into an equivalent equation for the function $H(x)$, defined by $H(x) = (1 - F_s(x))e^{-\int_0^x q(x)dx}$. The resulting equation for $H(x)$ is then numerically solved by an iterative technique that we prove to be stable and rapidly convergent. Once $H(x)$ and, subsequently, $F_s(x)$ have been computed, the distribution for (W^n) is obtained by essentially reversing the space and time scalings employed. Of course, in this case, we have to be careful to account for the state-dependent nature of the time-scaling used. This technique permits us to evaluate the window distribution of any TCP process whose packet loss rate can be shown to be a function only of its instantaneous window size. We shall later see how our perturbation-based approach results in precisely such a loss model: by essentially treating the window sizes of the other connections as constants, we can relate the buffer occupancy (and hence the loss probability) directly to the window size of a specified individual flow. The scaling techniques and numerical procedure outlined here can then be applied to obtain the window distribution for each connection individually.

III. Estimating the Mean Queue Occupancy

To estimate the *center* of the queue occupancy, we use a set of fixed point mappings. The basic idea is to find values for the average window sizes, such that the average queue size given by those set of values is consistent with the average loss probability that is implied by the window sizes. The derivation of the 'square-root' formula via the drift-based technique is borrowed from [2]. As noted earlier, let Q^* be this mean or center value of the queue occupancy and let W_i^*, $i \epsilon \{1, 2, \ldots, N\}$ be the center of the i^{th} TCP flow.

A. Formulating the Fixed Point Equations

Define the *drift* of the congestion window of a TCP flow by the expected change, ΔW, in its window size. Since, for a window size of w, the window size (in packets) increases by $\frac{1}{w}$ with probability $1 - \bar{p}(w)$ and decreases by $\frac{w}{2}$ with probability $\bar{p}(w)$, we have:

$$\Delta W = (1 - \bar{p}(w))\frac{1}{w} - \bar{p}(w)\frac{w}{2}. \tag{3.1}$$

From the above equation, the *center* or '0-drift' value of W, called W^*, is seen to be

$$W^* \approx \sqrt{2\frac{1}{\bar{p}(W^*)}} \tag{3.2}$$

where the approximation is quite accurate as $\bar{p}$ is usually quite small [5] (for current TCP versions, if the drop probability exceeds 0.05, timeouts and slow start phenomena begin to dominate TCP behavior.)

For the case of multiple TCPs, the zero-drift analysis gives the window size (in packets) for flow i, as

$$W_i'(packets) = \sqrt{\frac{2}{p_i(Q^*)}} \tag{3.3}$$

[5] A more accurate analysis [2] reveals that the mean window occupancy, in ack time, is given by $W^* \approx \frac{1.5269}{\sqrt{\bar{p}}}$. It is this value that we used in all our experimental results ; for notational ease, however, we shall continue using the $\sqrt{2}$ approximation in our exposition.

By incorporating expression (2.1) in the above equation and noting that each packet is of size M_i bytes, we get the mean window size (in bytes) as

$$W_i^* = \frac{M_i}{c_i}\sqrt{\frac{2}{p(Q^*)}} \tag{3.4}$$

Now, let C_i be the bandwidth obtained by TCP i. Assuming that there is no significant buffer underflow and that the link is fully utilized, we get the relation $\sum_{i=1}^{N} C_i = C$. C_i can also be computed by a different method: by noting that a TCP connection sends one window worth of data in one effective round trip time. Since a queue of size Q will contribute a buffering delay of $\frac{Q}{C}$, the effective round trip time of connection i is $RTT_i + \frac{Q}{C}$; whence, we can related C_i to W_i by the expression

$$C_i = \frac{W_i}{RTT_i + \frac{Q}{C}} \tag{3.5}$$

On summing the C_is from the above equation and equating them to C, we get

$$C = W\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{RTT_i + \frac{Q}{C}} \tag{3.6}$$

or, upon simplification,

$$W = \frac{1}{\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{Q + C.RTT_i}} \tag{3.7}$$

where $W = \sqrt{\frac{2}{p(Q)}}$. For notational convenience, let the RHS of equation (3.7) be denoted by the function $g(Q)$ so that $g(Q) = (\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{Q+C.RTT_i})^{-1}$.

The *fixed point* solutions for the 'average' TCP window sizes and the queue occupancy is then given by the set of values that provide a solution to the following simultaneous equations:

$$W = \sqrt{\frac{2}{p(Q)}} \tag{3.8}$$

$$W = (\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{Q + C.RTT_i})^{-1} = g(Q) \tag{3.9}$$

The individual 'average' TCP windows are then computed from W^* by the relationship

$$W_i^* = \frac{M_i}{c_i} W^* \tag{3.10}$$

B. Existence and Solution of Fixed Point

We now prove the existence of a unique solution to the above simultaneous equations and also provide a numerical technique for its rapid computation.

The existence of a unique solution can be demonstrated graphically (as in figure 1) by plotting each equation as a line on the (Q, W) axes. Since $p(Q)$ is assumed non-decreasing in Q, we have W in equation (3.8) to be a non-increasing function of Q, while in equation (3.9), $g(Q)$ can be seen to an increasing function of Q. The two plots will therefore intersect at a single point, which is our 'zero-drift' solution for W^* and Q^*.

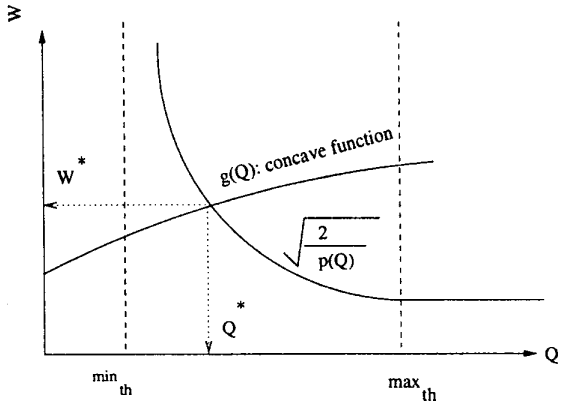

Figure 1: Typical Relationship between W and Q for Random Drop Queues

In Appendix A, we prove that the function $g(Q)$ is concave; accordingly we can see that the function $f(Q)$, defined by the difference between the RHS of equations (3.8) and (3.9), is convex in Q.

$$f(Q) = \sqrt{\frac{2}{p(Q)}} - \frac{1}{\sum_{i=1}^{N} \frac{M_i/c_i}{Q+C.RTT_i}} \tag{3.11}$$

Hence, we use the Newton gradient technique, which is guaranteed to converge and provide a solution to the equation $f(Q) = 0$, to solve for the fixed point. We start with an initial estimate of $Q_0 = min_{th} + \delta$ (an initial value to the left of Q^*) and proceed with repeated iteration. In this particular setting, the derivative $f'(Q_j)$ at the j^{th} iteration is given by

$$\frac{p'(Q_j)}{\sqrt{2}p(Q_j)^{\frac{3}{2}}} - \frac{\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{(Q_j+C.RTT_i)^2}}{(\sum_{i=1}^{N} \frac{\frac{M_i}{c_i}}{Q_j+C.RTT_i})^2} \tag{3.12}$$

C. Insights from Above Analysis

The drift analysis technique provides some insights for predicting or controlling the stationary behavior of persistent TCP connections and for understanding the accuracy of our approximation technique. For example, our analysis shows that:

- TCP connections with the same round trip time but different packet sizes will see the same 'average' window size (in bytes) if $c_i = \alpha M_i \ \forall i$, where α is an arbitrary constant. In other words, to ensure fair sharing of throughput among TCP connections with different packet sizes, the packet dropping probability should be

proportional to the *square of the packet size.* Contrast this with current byte-mode drop schemes where the packet drop probability is normally proportional to the packet size.

- TCP connections which are identical, except for different round trip times, will observe relative throughput that is inversely proportional to the round trip times. This unfairness towards TCP connections with larger round-trip times is well known.
- Since W^* (the 'fixed point' that satisfies both equations (3.8) and (3.9)) is identical for all flows, it should be clear from equation (3.10) that the mean value of the window size (in packets) for all TCP flows, which have the same drop function (same p_is), will be the same, irrespective of their round-trip times and segment sizes. The point is more subtle than apparent at first glance: the means are identical only when expressed in MSSs and when the distribution is taken with respect to *ack time.* When sampled in *clock time,* the distribution of the window size and even the mean value of each TCP connection will indeed depend on its round-trip delay (which influences the rate of progress of the connection). We can, however, easily compute the distribution in clock time from that in ack time, if the round-trip delay for a specific connection is non-varying (through the relation $dF_{ack}(x) = \frac{x dF_{clock}(x)}{\int_0^\infty y dF_{clock} y}$). As the number of flows increases, we shall later see that the buffer occupancy (and hence, the queuing delay) shows relatively smaller variation; estimates of clock-time distributions from our ack-time calculations can then be expected to be more accurate.
- Since the mean analysis technique is based upon an ideal model where each TCP window stays around its 'average' value and the loss rate is constant, the accuracy of our predictions should be higher when the buffer occupancy (and hence the loss probability) does not change appreciably. We shall later see that the TCP window sizes, when interacting with an ERD-based queue, are not independent but reveal negative correlation; accordingly, the queue occupancy shows less variance {mathematically speaking, *the coefficient of variation,* $\frac{\mathbf{Std.Dev(Q)}}{\mathbf{Mean(Q)}}$, *decreases* } with an increase in N, the number of connections. Accordingly, the queue occupancy shows less variation with increasing N, making the estimates via the mean value approximation technique progressively more accurate. This explanation and prediction is corroborated by results presented later, in figures 4 and 5. Furthermore, we also note in passing, that for our model of a TCP flow subject to packet drops with a constant drop probability, $Std.Dev(W) = 0.38E[W]$, i.e., the coefficient of variation is around .4.

D. Simulation Results for The Mean Window Sizes

A wide variety of simulation experiments, with various combinations of segment sizes and round trip times, were performed to verify the accuracy of our fixed point-based prediction technique. All simulations are carried out on the *ns* [10] simulator with sources implementing the New Reno version of TCP. The queue service rate equals 1.5 Mbps throughout the results in this paper. While the numerical analysis (including the estimation of the distributions of the individual congestion windows) takes less than 1-2 mins on a conventional workstation, the simulations would require 20 mins (and higher, depending on the number of connections) before results with an acceptable degree of statistical confidence could be obtained. To study the accuracy of our drift analysis, we simulated both RED (Random Early Detection) and ERD (Early Random Drop) queues. The differences between these algorithms and the necessary corrections to our model (for RED) are presented in Appendix B.

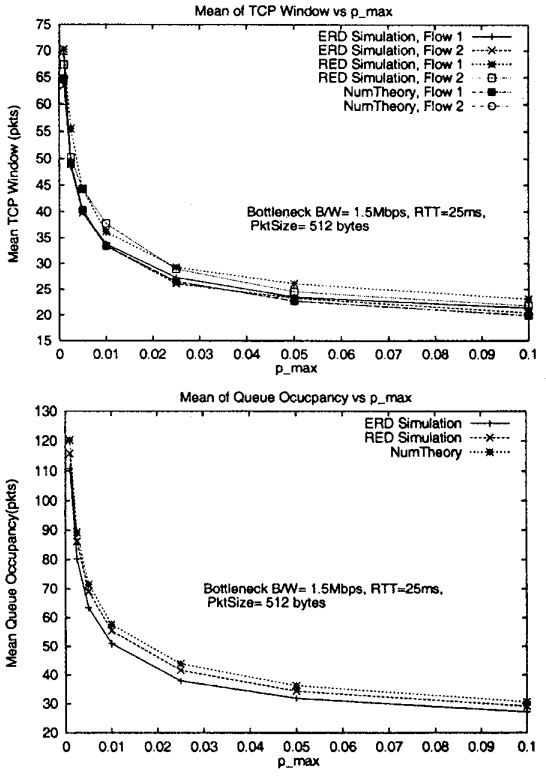

Figure 2: Mean Behavior with 2 Identical Connections

A set of illustrative examples are presented in figures 2 and 3. In these simulations, we had two concurrent TCP connections, with 512 byte packets, interacting with a single bottleneck queue. The queue parameters were kept as follows: $min_{th} = 10240$ bytes, $max_{th} = 102400$ bytes and the buffer size was kept at 256000 bytes. p_{max} was varied between the values outlined in the plots. Figure 2 considers two TCP connections with identical parameters while in Figure 3, we have two connections with the nominal RTT of the second con-

nection double that of the first connection's RTT (called the BaseRTT in the figure). By varying p_{max}, we change the slope of the drop function and hence, the 'zero-drift' point of the queue occupancy. In general, the accuracy of our predictions would slightly degrade for larger RTT values, although in all cases the agreement was within $10 - 15\%$ of the predicted values. This is expected because a larger RTT essentially increases the chance of buffer underflow (which invalidates our model) by increasing the feedback time of the TCP control loop. Since our model does not account for phenomena like fast recovery (during which the queue size reduces), we tend to predict larger queue occupancies than those obtained via simulation. Also, as expected, the quality of the prediction increases with the number of flows N (as long as $p(Q^*)$ did not become large enough to cause timeouts).

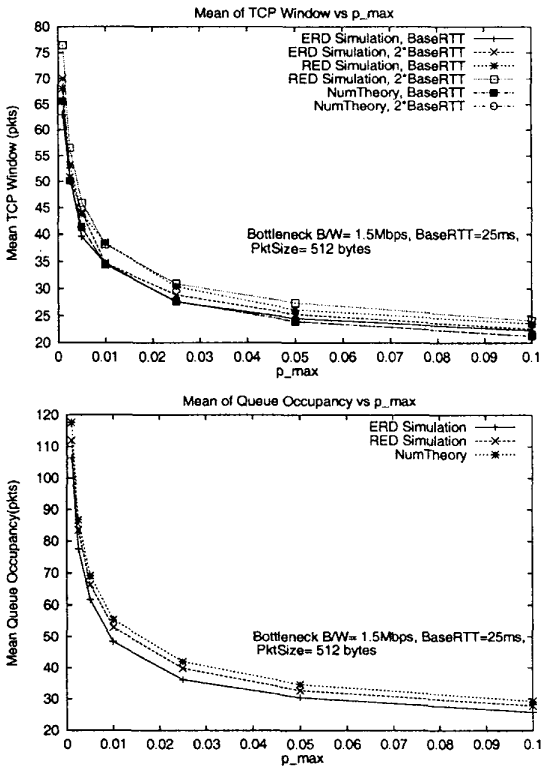

Figure 3: Mean Behavior with 2 Dissimilar Connections

Our simulations hence support our analysis, which states that the means of the TCP windows (in segments) should be identical (in ack time), even though the round-trip times of the various flows and the segment sizes are different. It should also be noted that the negative correlation among window sizes (discussed a little later) helps to reduce the variation in packet loss probability and improves the accuracy of our technique.

IV. Computation and Accuracy of Individual Distributions

A. Computation of Individual Distributions

Having seen how to compute the 'mean' of the individual distributions and the queue occupancy, we now proceed to determine the detailed distribution of the individual connections. Since the approach is identical for all the connections, we consider, in general, the i^{th} connection with a calculated mean of W_i^*, a segment size of M_i, a drop function $p_i(Q)$; as before, the computed mean of the queue occupancy is Q^*.

We use our independence assumption to postulate that the other connections always have their window size equal to their computed means [6]. For the square-root approximation, we assume that the loss probability of a packet for the flow is constant and does not change with the window size. This constant packet loss probability is given by the value of $p(Q)$ when $Q = Q^*$. The window distribution (in ack time) is then computed using the rescalings and the analytical formula presented earlier and in [2].

For the perturbation-based analysis, if W_i is the window size (in packets) of the connection under consideration, the buffer occupancy, Q, corresponding to this window size is given (in bytes) by the relation

$$Q = [Q^* + \frac{(W_i * M_i - W_i^*) * Q^*}{Q^* + C * RTT_i}]^+ \qquad (4.1)$$

where the $[\,]^+$ reflects the fact that the queue occupancy cannot be negative. Accordingly, we now have a state-dependent loss probability for the TCP connection where the packet loss probability is a function of the window size W and is given by

$$\tilde{p}_i(W) = p(Q) = p([Q^* + \frac{(W_i * M_i - W_i^*) * Q^*}{Q^* + C * RTT_i}]^+) \qquad (4.2)$$

We can then use the technique outlined in this paper earlier (and detailed in [3]) to determine the window distribution.

B. Simulation Results for 'Distribution' of TCP Windows

We now present the result of comparing the distributions predicted by our analytical techniques with that obtained via simulation. As before, the simulations were carried out on the *ns* simulator with the TCP New Reno model. Several sets of experiments were carried out with the number of connections

[6] A few words are in order about our assumption that the queue occupancy of the other connections can be represented by their mean. In general, the loss probability, for a particular value of W_i, is a *random variable*, say X, whose value will depend on the instantaneous values of the other windows; let us denote this dependence by $X = p(\sum_{j \neq i} W_j + W_i)$. Now the expected value of X, conditioned only on the window W_i of the flow under consideration, is denoted by $E[X]$ and equals $E[p(\sum_{j \neq i} W_j + W_i)]$. This conditional expectation equals $p(\sum_{j \neq i} E[W_j] + W_i)$ only if the loss function p is linear. Accordingly, for linear loss functions, our formulation is equivalent to assuming that the loss probability for a given window size is replaced by the expected loss probability for that size; this explanation does not hold when the loss function is non-linear.

varying from $2 - 20$ and with wide variations in the round trip times and segment sizes. For all the plots presented here, $min_{th} = 10240$ bytes, $max_{th} = 102400$ bytes and $p_{max} = 0.05$.

B.1 Negative Window-size Correlation and its Consequences

Before presenting the simulation results themselves, we discuss an important relationship that we have observed between the window sizes of multiple TCP connections. We noticed this relationship only when trying to analyze the simulation results; for lucidity of presentation, we present this empirical result upfront.

The perturbation-based approach assumes that the window sizes of the other flows are *uncorrelated* to the window size of the flow under consideration; the queue occupancy consequently increases and decreases in tandem with the window size of the flow. If this were true (the windows were truly uncorrelated), the window size probability distribution would indeed have less spread (be more concentrated around the mean): any increase beyond the mean would result in a larger drop probability (and more aggressive drops) while any decrease below the mean would be immediately compensated for by a less aggressive drop probability. Consider now what would happen if the flows were *negatively* correlated; we use the case of 2 flows for the ease of presentation. A negative correlation implies that when the window size of one flow is large, the other one has a smaller window size, and vice versa; the queue occupancy thus exhibits lower variability and tends to be less dependent on the variations in the window size of a single flow. In such a *negatively correlated* environment, the square-root technique would perform better than the perturbation technique, since it (correctly) assumes that the queue size (and the loss probability) is largely independent of the flow's window size. On the other hand, if the flows were *positively* correlated (flows tended to increase and decrease in tandem), the perturbation technique should provide a better fit than the square-root model, although both models could indeed exhibit lower accuracy.

The experiments and results reported here use the ERD algorithm where the drop behavior is memoryless and is based on the *instantaneous* queue occupancy. To investigate the correlation for two flows ($N = 2$), we **sampled** *in clock time* the window size of each flow and determined the individual and joint moments of their distributions. The resulting correlation coefficient turned out to be $-.4$, indicating a not insignificant degree of negative correlation. For the general case of N flows, where individual correlation indices are somewhat harder to comprehend, we use the sampling technique to plot the variance of the sum of the window sizes $Var(\sum_{i=1}^{N} W_i)$ against the sum of the individual variances $\sum_{i=1}^{N} Var(W_i)$. We know that the two should be equal if the flows are ideally uncorrelated; for negative correlation, the sum should exhibit lower variance ($Var(\sum_{i=1}^{N} W_i) < \sum_{i=1}^{N} Var(W_i)$), while for positive correlation, the sum should exhibit larger variance ($Var(\sum_{i=1}^{N} W_i) > \sum_{i=1}^{N} Var(W_i)$). (This follows from the general relationship

$$Var(\sum_{i=1}^{N} W_i) = \sum_{i=1}^{N} Var(W_i) + \sum_{i \neq j} Cov(W_i, W_j) \quad (4.3)$$

Hence, if the covariance terms are negative, then the LHS of equation (4.3) is less than the RHS.)

A graph showing the observed behavior for N identical flows (flows with identical operating parameters), for different values of N, is shown in figure 4. The figure shows that $Var(\sum_{i=1}^{N} W_i)$ is always less than $\sum_{i=1}^{N} Var(W_i)$ (and, in fact, $Var(Q)$ is even lower than $Var(\sum_{i=1}^{N} W_i)$). This indicates the presence of 'negative correlation' among the TCP flows. This observation will help explain our later results which show that, for a majority of the simulated cases, the square-root approach provides a better estimate than the variable-probability perturbation technique.

Another interesting observation can be made by observing the graphs in figure 5, where we plot the coefficient of variation ($\frac{Std.Dev}{Mean}$) of the queue size, the coefficient of variation of the sum of the window sizes ($\frac{\sqrt{Var(\sum_{i=1}^{N} W_i)}}{Mean(\sum_{i=1}^{N} W_i}$) and the mean of the coefficient of variation of the N TCP flows ($\frac{\sum_{i=1}^{N} CoeffVar(TCP_i)}{N}$). As the figure shows, the coefficient of variation of the queue (as well as the sum of the window sizes) decreases with increasing N, indicating that the queue becomes smoother (the variance of the queue occupancy increases more slowly than the average queue occupancy). This corroborates our observation in section III.C, which predicted a decrease in the coefficient of variation for the queue with increasing N and used this to explain why our 'mean-value analysis' gets progressively more accurate with increasing N. Also, note that the mean of the coefficient of variation of the TCP flows stays around .4, indicating a reasonable validation of our constant drop probability assumption.

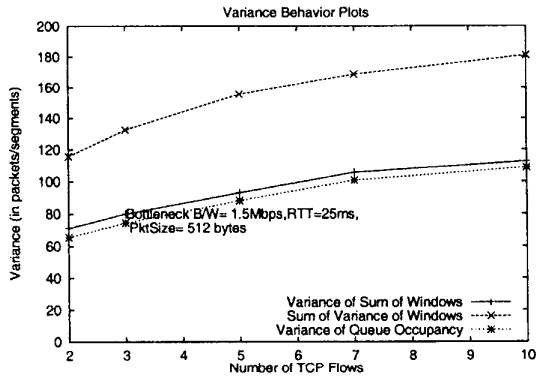

Figure 4: Variance Plots for TCP flows over an ERD Queue

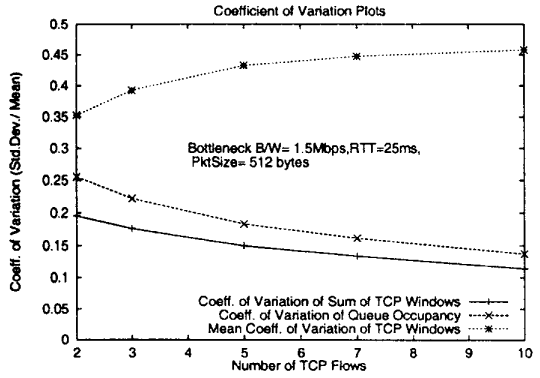

Figure 5: Coefficient of Variation Behavior

B.2 Illustrative Results

We performed an extensive set of simulations to understand the behavior and accuracy of our estimates. The results seem to indicate that, across a wide range of operating conditions, the analytical techniques offer a reasonably accurate estimate of the distributions of the different flows. In particular, it comes as no surprise to observe that the predictions improve in accuracy when the number of flows increases (until the loss probabilities become large and transient TCP phenomena like timeouts become significant): as the number of flows increases, the dependency of the queue occupancy on a single connection, as well as the coefficient of variation of the queue occupancy, decreases; consequently, the assumptions behind both the perturbation approach and the square-root technique become progressively more accurate. It should also be noted that, not only is the square-root approach usually more accurate than the perturbation approach, it is always computationally cheaper and simpler than the perturbation technique.

The simulations in figure 6 compare the results when 2 or 5 concurrent TCP connections, all having the same parameters, share the ERD queue. The packet sizes are 512 bytes and the round trip times are $25ms$. As we can see, the agreement is fairly close; the square-root approximation, in fact, gives a very good fit.

In figure 7, we present results for simulations involving 2 or 5 flows, all of which have the same packet size (512 bytes) but different round trip times (the distributions should be the same). The RTT of the first flow is $25ms$ while each subsequent connection has a RTT double that of the previous flow. For conciseness and clarity, in each case, we present the comparison of the results for 2 flows, those with the smallest and largest RTT respectively; the agreement is observed to be fairly good; the square-root approach again proves superior to the perturbation approach.

Figure 8 shows the result of experiments similar to those of figure 7, except that now we keep the round trip time constant at $25ms$ but vary the segment sizes; each flow should now have a different distribution. As before, the segment size of a connection is twice that of the previous connection. The smallest segment size is mentioned in the plots which, as before, are shown only for the flows with the smallest and largest segment sizes. Fairly good agreement is observed again. In this case, the square-root approach gives an identical distribution for all the connections, while the perturbation approach gives different estimates for each flow. We can see that, for the flow with the largest segment size, the perturbation technique provides a better estimate than the square-root technique; this is because the square-root technique is unable to capture the fact that the queue occupancy changes with a change in the window size (which is more acute for flows with larger MSS).

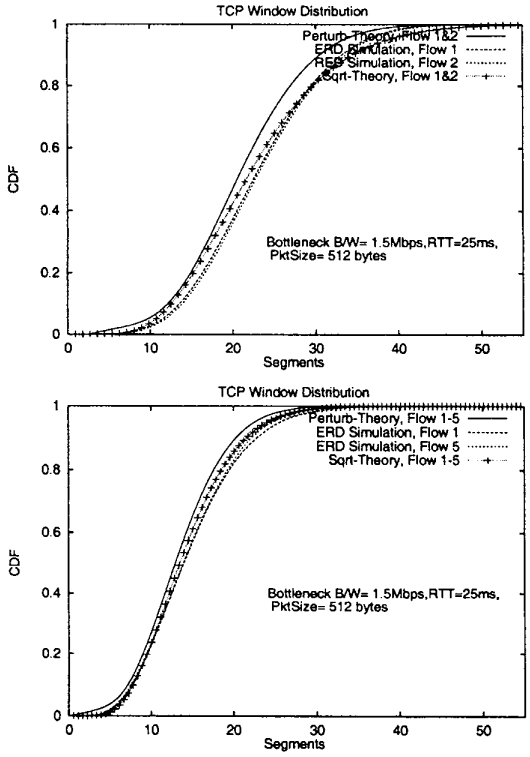

Figure 6: 2/5 Identical TCP Connections

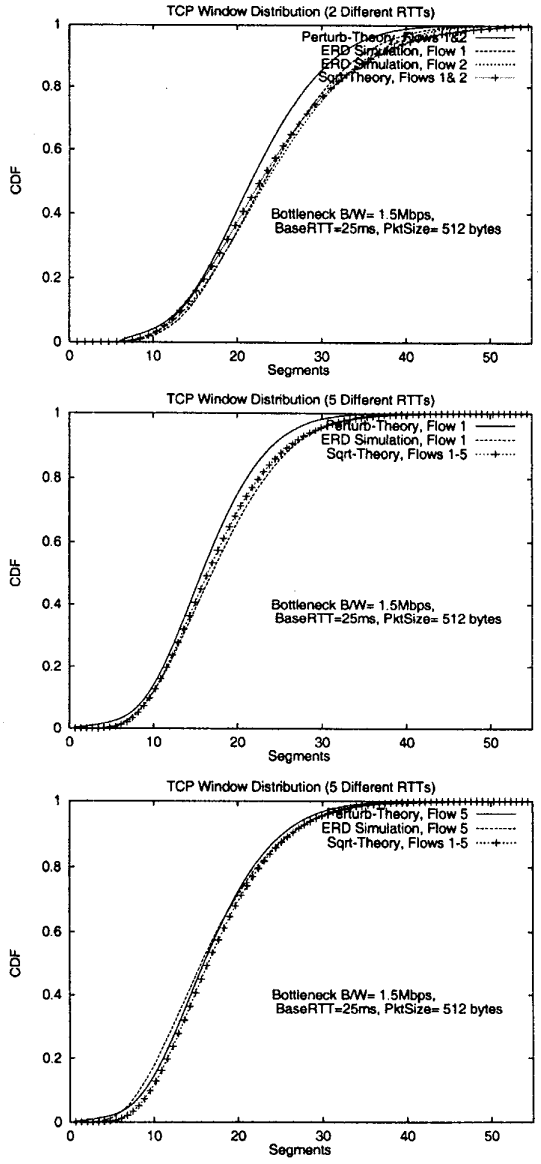

Figure 7: 2/5 Connections with Different RTT

To further illustrate the effect of negative window correlation and the consequent accuracy of the simpler square-root approach, we carried out a series of experiments where we simply varied the number of concurrent flows. Each TCP flow had identical parameters like segment sizes and round trip times and the drop function was constant across all simulations. The results for 2, 5, 10 and 15 flows are presented in figure 9. The graphs with the 'Theoretical Distn.' label refer to the plots for the perturbation-based predictions. As we can see from the graphs, the square root-based predictions outperform the perturbation-based predictions, with increasing accuracy at larger N.

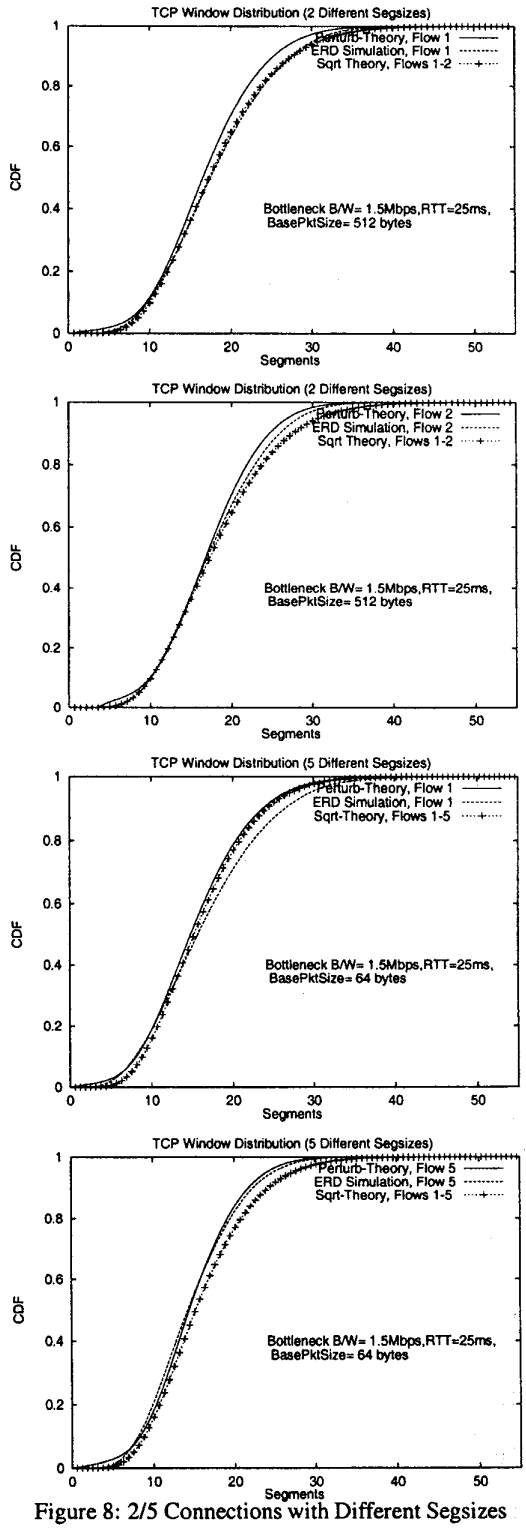

Figure 8: 2/5 Connections with Different Segsizes

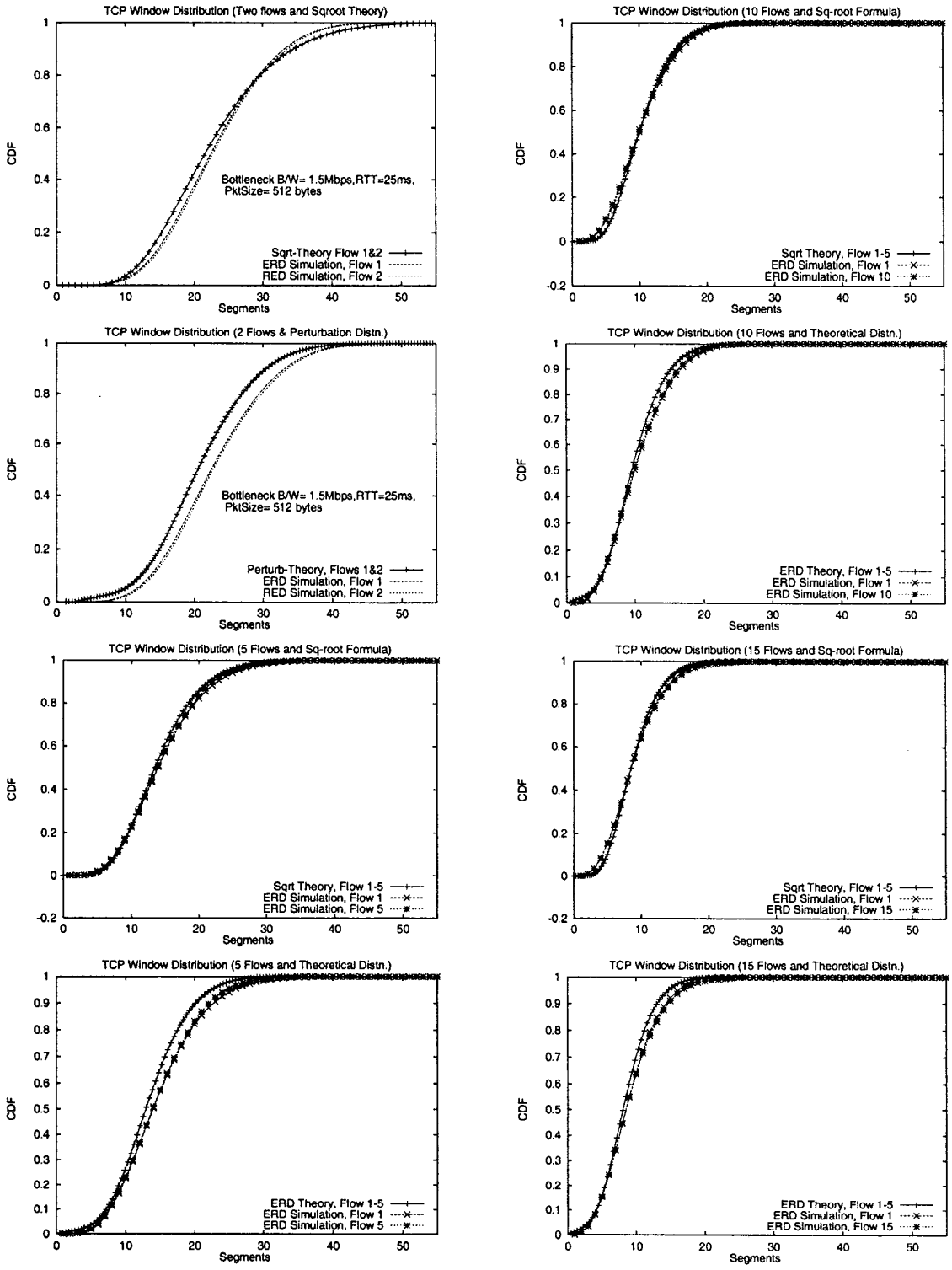

Figure 9: 2/5/10/15 Connections
(Square-root vs. Perturbation Approach)

V. Conclusions

In this paper, we have demonstrated an analytical and numerical technique to obtain the centers of the TCP window sizes and the associated queue occupancy when multiple persistent TCP flows share a bottleneck buffer performing random packet drops. We then evaluated two competing techniques to determine the window distribution of each of the individual TCP flows. One technique derives the distribution assuming a constant loss probability whose value is determined by the center of the queue occupancy. The other technique assumes a variable loss probability model and uses a perturbation-type approximation to relate the packet loss probability for a given connection to the window size of that connection alone. Simulation experiments indicate that both techniques are fairly robust (accurate to within $\approx$ 10%); the numerical predictions are much more accurate at lower values of the average drop probability. For ERD queues, the observed negative correlation between the window sizes of different flows causes the predictions of the constant loss model to outperform the predictions of the variable loss model, in a majority of the experiments.

Several interesting questions remain to be answered in the future. It is not immediately clear how various techniques, such as basing the drop probability on a subset of the past history of packet drops (as in RED's use of 'average queue occupancy') or controlling the distribution of the drop pattern (as in RED's 'drop-biasing' technique), affect the negative correlation observed here and alter the variability of the queue occupancy. Preliminary results (which we hope to report in later publications) suggest that regulating the pattern of random loss behavior can appreciably alter the buffer occupancy dynamics. Modifications to the TCP algorithm that reduce its window variance can also improve the stability of the queue occupancy and lead to better prediction of the window distributions. Accordingly, we propose to investigate TCP modifications (such as a better response to Explicit Congestion Notification) that reduce this variance. We also hope to study the applicability of this model to cases where different flows have different c_is (different drop probabilities), which would be the case in settings similar to those proposed in Weighted RED.

VI. Appendix

A. Proof that f(Q) is convex

We prove here that the function $f(Q)$ defined in equation (3.11) is convex. First, some notation: let $\frac{M_i}{c_i}$ be denoted by b_i and $C.RTT_i$ be denoted by d_i. The function $g(Q)$ is then given by $g(Q) = (\sum_i \frac{b_i}{Q+d_i})^{-1}$. On differentiating this function we obtain

$$g'(Q) = g(Q)^2 \sum_i \frac{b_i}{(Q+d_i)^2} \tag{6.1}$$

Since from above, $g'(Q) > 0 \, \forall Q$, $g(Q)$ is an increasing function of Q. Differentiating again, we have the second derivative given by

$$g''(Q) = 2g(Q)g'(Q) \sum_i \frac{b_i}{(Q+d_i)^2}$$

$$-2(g(Q))^2 \sum_i \frac{b_i}{(Q+d_i)^3}$$

or on rearranging

$$g''(Q) = 2(g(Q))^3 \{(\sum_i \frac{b_i}{(Q+d_i)^2})^2$$

$$-(\sum_i \frac{b_i}{(Q+d_i)^3})(\sum_i \frac{b_i}{Q+d_i})\} \tag{6.2}$$

We now prove that the term in the curly braces in equation (6.2) is negative. To see this, let $\beta = \sum_i b_i$ and let $a_i = (Q+d_i) \, \forall \, i \, \epsilon \, \{1, 2, \ldots, N\}$ (note that a_i is always positive). Consider a random variable A which takes on the value a_i with probability $\pi_i = \frac{b_i}{\beta}$. Then, the second derivative can also be written (with $E[\,]$ denoting the expectation operation) as

$$g''(Q) = 2\beta^2(g(Q))^3\{E^2[A^2] - E[A^3]E[A]\} \tag{6.3}$$

Now, we know if A is a random variable that has $Prob(A > 0) = 1$, then $\log E[A^m]$ is convex in $m \; \forall m \geq 0$. Thus, we have $\log E[A^2] \leq \frac{\log E[A] + \log E[A^3]}{2}$, so that $E^2[A^2] - E[A^3]E[A] \leq 0$. Applying this result to expression (6.3), we see that $g''(Q)$ is negative and hence, $g(Q)$ is a *concave* function of Q.

As the term $\sqrt{\frac{2}{p(Q)}}$ is easily seen to be convex (its second derivative is positive), we can conclude that $f(Q)$ is a convex function of Q.

B. Modeling RED behavior

In this appendix, we discuss differences between Early Random Drop (ERD) and the Random Early Detection (RED) that affect the applicability of our model. The important points of difference are:

- RED operates on the average (and not the instantaneous) queue length. The drop probability, p,is thus a function of the weighted average Q_{avg} of the queue occupancy i.e., p is a function not just of Q_n but of $(Q_n, Q_{n-1}, Q_{n-2}, \ldots)$ with an exponential decay.
- To avoid unbounded inter-drop gaps, RED increases the drop probability for every accepted packet. (This property, which we call *drop biasing*, is achieved by using a variable, *cnt*, which increments with every successive accepted packet; the true dropping probability is then given by $\frac{p(Q)}{1 - cnt.p(Q)}$. This results in a inter-drop period that is uniformly distributed between $(1, \ldots, \lfloor \frac{1}{p(Q)} \rfloor)$ as opposed to the geometrically distributed inter-drop gap caused by an independent packet drop model.

- RED has a sharp discontinuity in drop probability: when Q_{avg} exceeds max_{th}, $p(Q) = 1$ so that all incoming packets are dropped. This contrasts with our assumption of random drop throughout the entire range of the buffer occupancy. This is however not a problem as long as the queue occupancy almost never exceeds max_{th}.

While the effects of averaging cannot be incorporated in our model, a simple change works quite well in capturing the effect of drop biasing. We essentially change $p(Q)$ in our random drop model such that the average inter-drop gap $\frac{1}{p}$ becomes equal to the average inter-drop gap $\frac{1}{2p}$ of RED. All we have to do is to make our model p_{max} double that of the p_{max} used in the actual RED queue. It is interesting to speculate that the averaging of the queue occupancy in RED could have one interesting effect: depending on the memory of the averaging process, the correlations between the window sizes of different flows could change appreciably. We hope to investigate this possibility in greater detail in the future.

C. Correction for Delayed Acknowledgements

Delayed acknowledgements essentially imply that the TCP process increments its window only once for every K (K is usually 2) acknowledgements. A simple way to capture this effect is to alter equation (1.1) to

$$P\{W_i^{n+1} = w + \frac{1}{K.w} | W_i^n = w\} = 1 - p_i(w) \qquad (6.4)$$

i.e., approximate it by a process that increments its window by $\frac{1}{K}$ for every acknowledgement.

We point out the main changes to our technique for the case $K = 2$ (for other values, refer to the concerned publications):

- The square-root relationship now becomes $W^* = \sqrt{\frac{1}{p(W^*)}}$ instead of equation (3.2). This affects the first equation (3.8) in the set of simultaneous quations that define the fixed point.
- During the time re-scaling required to obtain the individual distributions, the function $q(X)$ mentioned in equation (2.2) is modified slightly (see [3] for details).

References

[1] V Jacobson, "Congestion Avoidance and Control", SIGCOMM 1988.

[2] T Ott, M Matthis and J Kemperman, "The Stationary Behavior of Idealized Congestion Avoidance", ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps, August 1996.

[3] A Misra and T Ott, "The Window Distribution of Idealized TCP Congestion Avoidance with Variable Packet Loss", Proceedings of Infocom '99, March 1999.

[4] J Padhye, V Firoiu, D Towsley and J Kurose, "Modeling TCP Throughput: a Simple Model and its Empirical Validation", Proceedings of Sigcomm '98, September 1998.

[5] A Kumar, "Comparative Performance Analysis of Versions of TCP in a Local Network with a Lossy Link", IEEE/ACM Transactions on Networking, August 1998.

[6] T V Lakshman, U Madhow and B Suter, "Window-based Error Recovery and Flow Control with a Slow Acknowledgement Channel: a Study of TCP/IP Performance", Proceedings of Infocom '97, April 1997.

[7] S Floyd, "Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic", Computer Communication Review, Vol.21, No. 5, October 1991.

[8] S Floyd and V Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993.

[9] V Jacobson, "Modified TCP congestion avoidance algorithm", April 30, 1990, end2end-interest mailing list.

[10] The ns-2 network simulator, http://www-mash.CS.Berkeley.EDU/ns.

[11] M Mathis, J Semke, J Mahdavi and T Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm", Computer Communications Review, July 1997.

[12] E Hashem, "Analysis of Random Drop for Gateway Congestion Control", MIT-LCS-TR-506.