TECHNICAL RESEARCH REPORT

Window Distribution of Multiple TCPs with Random Loss Queues

by Archan Misra, John S. Baras, Teunis Ott

CSHCN T.R. 99-36
(ISR T.R. 99-77)

JB 99-15

Entitled:

"Window Distribution of Multiple TCPs
with Random Loss Queues"

Authors:

A. Misra, J.S. Baras, and T. Ott

Conference:

Globecom'99
Rio de Janeiro, Brazil
December 5-9, 1999

Window Distribution of Multiple TCPs with Random Loss Queues

Archan Misra John Baras Teunis Ott

February 27, 1999

1. Introduction

In this paper, we consider the case of multiple *ideal* and *persistent* TCP flows (flows that are assumed to be performing idealized congestion avoidance) interacting with queue management algorithms that perform random drop-based buffer management. 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 to obtain the 'mean' queue occupancy and the 'mean' of the individual TCP windows. Armed with this estimate of the *means*, we then derive the window *distribution* of each individual TCP connection. Extensive simulation experiments indicate that, under a wide variety of operating conditions, our analytical method is quite accurate in predicting the 'mean' as well as the distributions. The derivation of the individual distributions is based upon a numerical analysis presented in [3], which considers the case of a *single* TCP flow subject to *variable state-dependent* packet loss.

Each TCP flow is assumed to adjust its window through the idealized congestion avoidance algorithm [1] whereby the TCP connection increments its window by 1 once every round trip time and halves it *instantaneously* on detecting congestion (through packet losses). Transient TCP behavior including phenomena like fast recovery, fast retransmit and slow start are thus ignored in this model of TCP window evolution. Mathematically speaking, the window evolution of the i^{th} TCP connection is given 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 stochastic process with the following evolutionary behavior:

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

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

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

In [3], we showed how appropriate space and time rescalings could be used to obtain the congestion window distribution for a TCP flow when the loss probability for its packets is variable but depends only on the instantaneous window size of the flow. This analysis was used to estimate the window distribution when a single TCP flow interacted with a router port performing RED [11] or ERD [10]. When multiple TCP connections are present, as in this paper, the 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. Modeling this multi-flow case exactly requires a multi-dimensional Markovian formulation whose dimension equals the number of TCP connections; such a model becomes exceedingly complex and is analytically intractable for even the simplest case of two concurrent TCP connections.

We therefore make a series of approximations to preserve the tractability of the problem. We first use a drift-based analysis (using the window evolution modeled by equations (1.1) and (1.2)) to derive the 'mean' or *center* of the queue occupancy as well as the 'mean' of each connection. We then assume that the window behavior of each individual TCP is statistically independent of the others and that the sum of the mean value of the windows of the other flows is a sufficient statistic in computing the loss function for a particular connection; we will elaborate on the implications of this assumption later. While computing the distribution for any given connection, we therefore keep the window sizes of the other connections constant at their 'mean's; this constancy reduces the occupancy of the queue to a linear dependence on the connection's window size. As a result, the loss probability has again been reduced to a deterministic function of the connection's window size; this problem can now be tackled as per the analysis in [3].

The approximation techniques are validated by the accuracy to which they can predict the true queue behavior and the true window distributions. By using simulations with a realistic TCP version (TCP New Reno), we conclude that our analysis exhibits in fairly good (within 10%) accuracy. To our knowledge, this is the first attempt to derive detailed distributions for the interaction of multiple TCP connections with random-drop based queue management mechanisms.

1.1 Related Work and Model Applicability

The window dynamics of a TCP connection in the congestion avoidance regime has been extensively analyzed under the assumption that the loss probability and round-trip times are constant. 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 is discussed in [9], [6] and [7]; a more careful analysis that provides the detailed distribution 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 when the loss probability is relatively large (above ≈ 0.05 for current TCP versions) and cause the throughput to become proportional to $\frac{1}{p}$ in this regime. In [3], we provide a detailed analysis of the window distribution when the loss probability is *not constant*; this analysis is then used to determine the distribution of the window of a single connection interacting with a RED (Random Early Detection) or ERD (Early Random Drop) queue. To our knowledge, this paper is the first attempt to derive detailed distributions by explicitly considering the interaction of multiple TCP connections with a random drop-based queue.

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 TCP behavior at moderately high loss probabilities is due largely due to

1. the integration of loss recovery mechanisms with congestion control in current versions of TCP like TCP Tahoe and Reno.

2. the coarse-grained nature of timers in current implementations.

Accordingly, this analysis is accurate for current TCP versions only when the loss probabilities are small enough and the delay-bandwidth product large enough (≈ 10 and above) to ensure that timeouts are relatively rare events. As mechanisms to separate loss recovery from congestion avoidance (like TCP SACK) become commonplace and as finer-grained timers are adopted, our analysis should hold over much larger variations in performance parameters.

2. 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. Under this model of idealized behavior, the connection never times out, the data is always send in equal-sized segments (although segment sizes could vary between connections) and 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***[1], 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 segment size of M_i bytes and a nominal round trip time (excluding the queuing delay at the buffer under consideration) of RTT_i seconds. As already mentioned, 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.

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), the amount of traffic from connection i that 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:

$$p(Q) \quad = \quad 0 \qquad\qquad \forall \;\; Q < min_{th}$$

[1] 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.

$$
\begin{aligned}
&= p_{max} && \forall\ Q > max_{th} \\
&= p_{max} * \frac{Q - max_{th}}{max_{th} - min_{th}} && \forall\ min_{th} \leq Q\ lemax_{th}
\end{aligned}
$$

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 useful drop functions.

Our analysis here is really intended for pure random drop queues, where the same drop function applies to all flows. Our formulation does, however, permit a slight generalization whereby the actual drop probability of a packet may indeed be flow-dependent. 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. This implies that our model 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 the ease of the subsequent analysis.

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 $c_i^2 = M_i$) [2]. Also, for notational convenience, let $\bar{p}_i(W)$ be the dropping probability of TCP_i as a function of its window size W.

2.1 Solution Approach

An $N-$ *dimensional* continuous-state Markov model is needed to accurately describe the window evolution dynamics of N concurrent TCP connections. The state of such a process can be described by the N-ary vector

[2] Our 'scalar-multiple' model of different drop probabilities for different connections 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 [12] where the different classes have the same min_{th} and max_{th} thresholds but only different max_ps. Although the application of this model to such scenarios is straightforward, we do not explore the validation of such settings further in this paper.

$[w_1, w_2, \ldots, w_i, \ldots, w_N]^T$. State transitions would be triggered by events which correspond to the loss or transmission of *any* packet (belonging to *any* connection). Even under the simplifying assumption that a randomly chosen packet would belong to a particular connection with a probability proportional to its instantaneous window size, the transition probabilities from $[w_1 w_2 \ldots w_i \ldots w_N]^T$ to $[w_1 \ldots w_i + \frac{1}{w_i} \ldots w_N]$ or to $[w_1 \ldots \frac{w_i}{2} \ldots w_N]$ would still depend on the sum of the instantaneous windows of all the connections $\sum_{j=1}^{N} w_j$. Not only does the dimensionality of this Markovian model increase linearly with the number of flows, we have, so far, no analytical solutions for its stationary distribution even for the simplest case of 2 flows. (The problem of using re-scalings similar to the one introduced in [3] is that the loss probability is no longer dependent on the state of an individual connection but really depends on the state of each of the N connections, thereby destroying the state-dependent loss assumption.)

We are therefore motivated to find an approximation technique whereby the loss probability of a particular TCP connection can be expressed only in terms of its individual state (the state of the other connections being approximated by a constant term). **Based on our assumption of independence among connections, we postulate an approach where we abstract the sum of the windows of the other connections by the sum of their mean values.** (We shall further comment on the implications of this approach for non-linear drop functions in a later section.) To determine the means, we use a drift-based argument (similar to the simpler derivations of the square-root formula) to determine the *center* of the TCP connection windows and, by association, the *center* of the queue occupancy. This derivation and solution is presented in the next section.

Once the center of the queue occupancy is known and the other connections are represented by their computed means, the instantaneous queue occupancy (and hence the drop probability) can be related to the instantaneous window size of a particular connection through a simple linear relation. The window distribution can then be computed directly from the analysis in [3]. Details of this scheme are presented in section 4. We shall also present numerical examples that compare our analytical results with those obtained via simulation; they seem to indicate that that this methodology is robust and fairly accurate.

3. 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 a value for the occupancy such that the loss probability associated with that value results in congestion windows for individual TCP connections (through the square-root formula) that are consistent (with the queue occupancy value). 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.

3.1 Formulating the Fixed Point Equations

Define the ***drift*** of the congestion window of a TCP flow by the expected change in its window size. From equations (1.1) and (1.2), we see that the window size (in packets) increases by $\frac{1}{w}$ with probability $1-\bar{p}(w)$ and decreases by $\frac{w}{2}$ with probability $\bar{p}(w)$. Hence, at a window size of w, the drift is given by

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

The *center* of the window is given by the value of W that results in a drift of 0; this can be shown, by simple rearragement of the terms in equation (3.1), to be given by by the expression

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

where the approximation is quite accurate as $\bar{p}$ is usually quite small [3] (for current TCP versions, if the maximum 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

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

[3] A more accurate analysis reveals that the mean window occupancy is given by $W^a st \approx \frac{1.5269}{\sqrt{p}}$. However, this makes no difference to our computations which typically have a larger margin of error.

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

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:

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

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

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

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

3.2 Existence and Solution of Fixed Point

Having defined the equations used to obtain the 'mean' TCP windows and the 'mean' queue occupancy, we now prove that a unique solution to the above equations exist and provide a numerical technique for its rapid computation.

The existence of a unique solution can be demonstrated graphically (as in figure 1) where the two simultaneous equations are as lines 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^*.

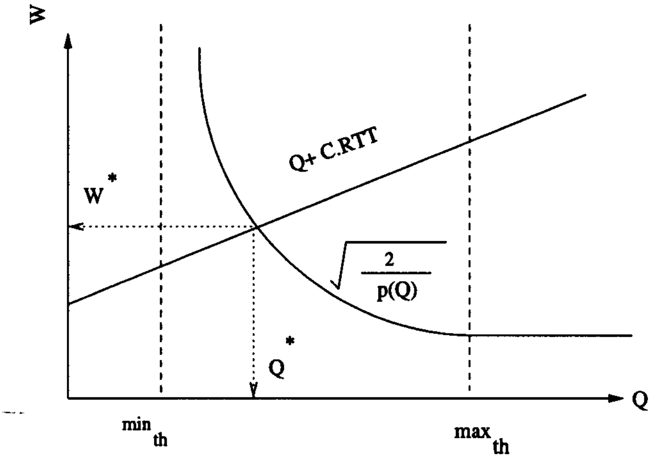

Figure 1: Typical Relationship between W and Q for Random Drop Queues

To solve the fixed point, we use the Newton gradient technique. We use this approach to find the zero of the function $f(Q)$ defined by the difference between the RHS of equations (3.8) and (3.9):

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

In Appendix A, we prove that $f(Q)$ is convex in Q. Hence we can 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 $f'(Q)$ is given by

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

3.3 Insights from Above Analysis

The drift analysis technique reveals a couple of interesting properties of the stationary behavior of TCP windows. For example, we can see that the following conclusions will hold:

- 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.

- We see from the analysis that the throughput of connections, which have identical parameters except for different round trip times, is inversely proportional to the round trip times. This unfairness towards TCP connections with larger round-trip times is well known.

3.4 Numerical Results

A wide variety of simulation experiments were performed to verify the accuracy of our prediction regarding the 'mean' TCP window sizes and the 'average' queue occupancy. All simulations are carried out on the *ns* simulator with sources implementing the New Reno version of TCP (which demonstrates all the non-ideal transients like coarse-grained timers and fast recovery). The experiments presented in this paper have the bottleneck bandwidth set to 1.5 Mbps. While the numerical analysis (including the estimation of the individual distributions) would take less than 1-2 mins on a conventioanl workstation, the simualtions would require between 20 mins to 1 hour (depending on the number of connections) to obtain results with a high degree of statistical confidence. 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 correction to our analytical model (for RED) are presented in Appendix B.

A set of representative figures are presented in figures 2 and 3 to illustrate this technique. Figure 2 considers two ***identical*** TCP connections while in Figure 3, we have two connections with the nominal RTT of the second connection double that of the first connection's RTT (called the BaseRTT in the figure). The graphs are plotted for various values of p_{max}, which essentially changes the slope of the drop function and the 'zero-drift' point of the queue distribution. We also experimented by changing the value of the nominal round-trip time. In general, the agreement between computed and simulated values would slightly degrade for larger RTT values, although in all cases the agreement was within 5-10predicted values. This degradation is expected because a larger RTT increases the chance of buffer underflow (which invalidates our model) owing to an increase in the feedback time of the TCP control loop. The analytical technique tends to overestimate the queue occupancy because we discount phenomena like fast recovery (during which the queue size reduces as TCP attempts to adjust its sending rate) and timeouts. However, the accuracy of the predictions seems to be quite creditable, given the simplicity of the analysis and the various approximations involved. The other noteworthy point was that closer agreement was obtained when the number of connections increased (as long as $p(Q^*)$ did not become large enough to cause timeouts); with a larger number of flows, the sensitivity of the queue size to the variations of a single connection decreases and hence the queue occupancy exhibits slightly lower variance.

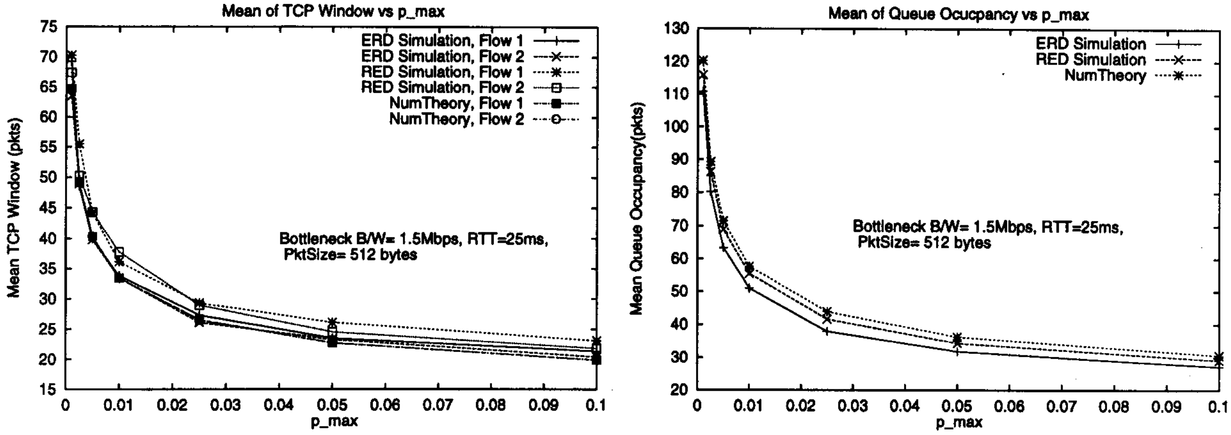

Figure 2: Mean Behavior with 2 Identical Connections

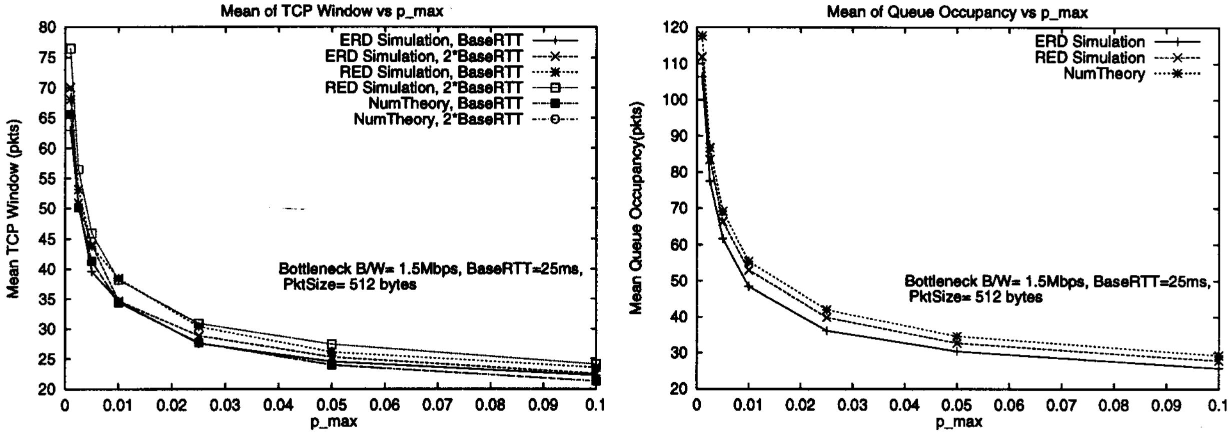

Figure 3: Mean Behavior with 2 Dissimilar Connections

4. Deriving the 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 queue occupancy is Q^*. For notational convenience, we drop the i from the subscript for the rest of this section.

We use our independence assumption to postulate that the other connections always have their window size equal to their computed means. Accordingly, if W is the window size of the connection under consideration, the buffer occupancy, Q, corresponding to this window size is given (in bytes) by the relation

$$Q = [Q^* - (W * M - W^*)]^+ \tag{4.1}$$

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

$$\bar{p}(W) = p(Q) = p([Q^* - (W * M - W^*)]^+) \tag{4.2}$$

We have now managed to reduce the window evolution process of the connection to one governed by a variable but state-dependent loss rate, which is exactly the model considered in [3]. Since the numerical-analytical technique to evaluate the window distribution in such a case was presented in that paper, we shall only outline the analytical approach behind the solution to the problem. [4]

The above state-dependent process is first converted to a continuous-time, continuous-state process by appropriately rescaling both the time and the state-space axes. The time-rescaling is state-dependent and results in a time frame called ***subjective time*** which increases in a non-linear manner compared to *ack* time. This time-rescaling is the critical element in defining a modified process which is characterized by the following path dynamics: **There is a Poisson process with intensity 1. In between the points of this process, the window W evolves according to a differential equation**

$$\frac{dW}{dt} = q(W) \tag{4.3}$$

[4] 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, is a ***random variable***, say X, whose value will depend on the instantaneous values of the other windows; let us denote this dependence by $X = p(\sum_{j \neq i} W_j + W)$. Now the expected value of X, conditioned only on the window W of the flow under consideration, is denoted by $E[X]$ and equals $E[p(\sum_{j \neq i} W_j + W)]$. This conditional expectation equals $p(\sum_{j \neq i} E[W_i] + W)$ 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.

(where q is an appropriate function). At a point τ of this process, the window behaves as $W(\tau^+) = W(\tau^-)$.
In [3], it is shown how the stationary Kolmogorov equation for the above process is obtained and how that equation can be solved through a rapidly converging iterative technique. Once the distribution of the process in *subjective time* has been numerically computed, we correct for the time and space scalings by essentially reverting the mappings. The technique also incorporates the window evolution over lossless regions of the state space (where the loss probability is 0 and the window evolves deterministically). For the case at hand, it should be mentioned that the lossless region will occur only if $Q^* > W^* + min_{th}$, a condition that may or may not occur for one or more flows, depending on the system parameters. It bears repetition that a reader wishing to understand the details of the stationary distribution computation must be familiar with the content of [3].

4.1 Simulation Results

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 varying from $2 - 20$ and with wide variations in the round trip times and segment sizes. The results seem to indicate that, across a wide range of operating conditions, the analytical technique offers a reasonably accurate estimate of the distributions of the different flows. n particular, it comes as no surprise to observe that the predictions improve in accuracy when the number of flows increases. As the numebr of flows increases, the dependency of the queue occupancy on a single connection decreases; consequently our approach of treating an individual connection as a perturbation on the overall queue occupancy becomes progressively more accurate.

The simulations in figure 4 compare the results when 2 or 5 identical concurrent connections 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.

In figure 5, we present results for a 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.

Figure 6 shows the result of experiments similar to those of figure 5, except that now we keep the round trip time constant at 25*ms* 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. Failry good agreement is observed again.

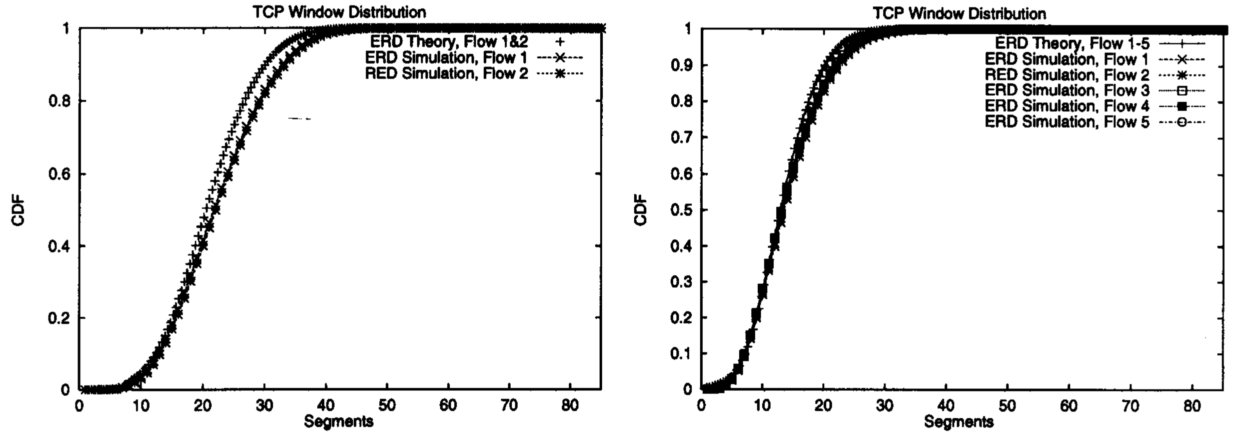

Figure 4: 2/5 Identical TCP Connections

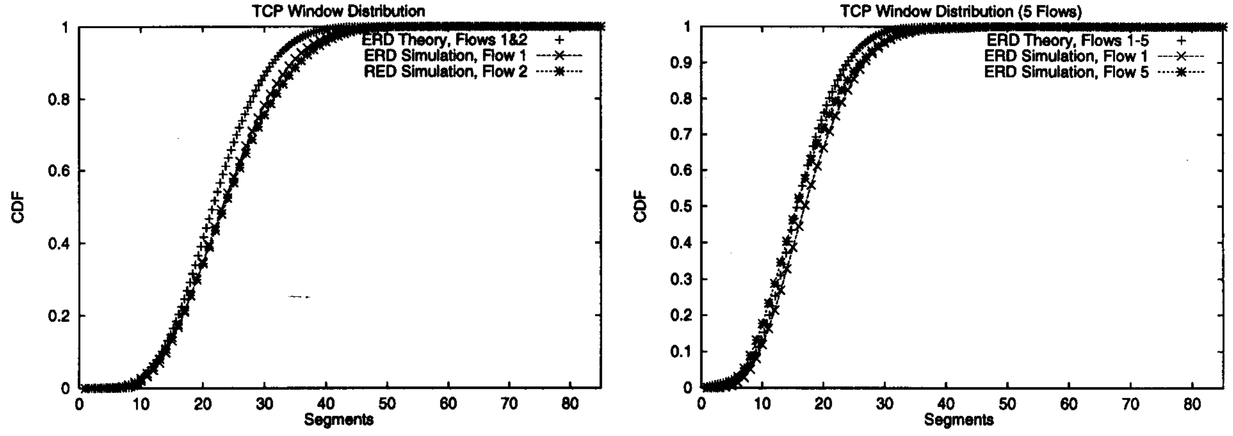

Figure 5: 2/5 Connections with Different RTT

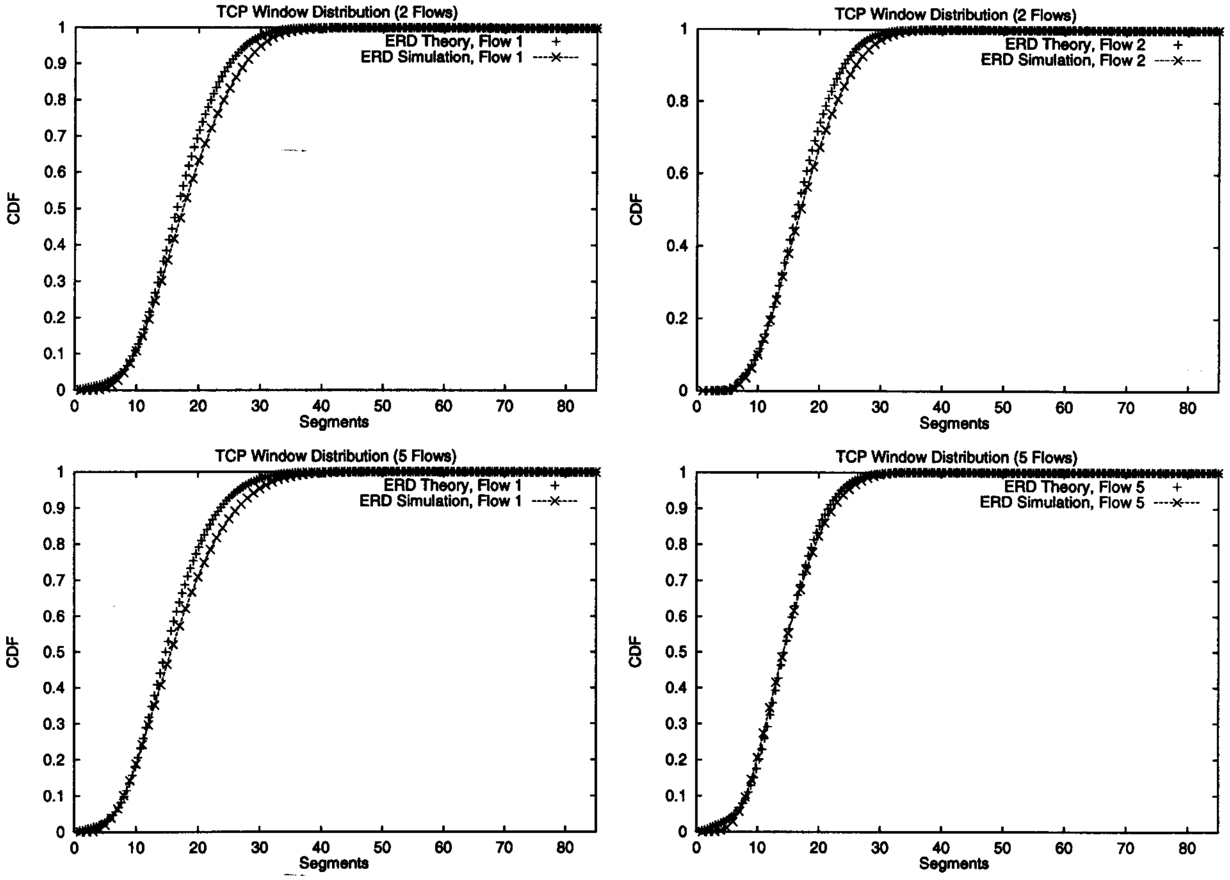

Figure 6: 2/5 Connections with Different Segsizes

5. Conclusions

In this paper, we have demonstrated an analytical and numerical technique to obtain the window distribution of individual TCP connections when multiple persistent TCP flows share a bottleneck buffer which performs random packet drops. This objective is achieved by first using drift analysis to obtain an estimate of the mean window sizes as well as the mean queue occupancy, and then using a perturbation-type approximation to relate the loss probability for packets of a given connection to the window size of that connection alone. Simulation experiments seem to indicate that our analytical technique is fairly robust and accurate to within $\approx 10\%$.

To our knowledge, this is the first attempt to predict detailed TCP distributions by explicitly considering the interaction with a random drop queue. In future we hope to use these individual distributions to determine the occupancy distribution of the queue itself; it will be interesting to see how accurate our analysis will prove in that case and also to see whether the analysis improves in accuracy as the number of flows increases.

6. Appendix

6.1 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 $\pi_i = \frac{b_i}{\beta}$ (check that this is a legitimate probability distribution). 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.

6.2 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.

6.3 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 $q(W)$ mentioned in equation (4.3) 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 Avodiance 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] V Jacobson, "Modified TCP congestion avoidance algorithm", April 30, 1990, end2end-interest mailing list.

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

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

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

[12] Need to get this reference