Stochastic Differential Equation for TCP Window Size: Analysis and Experimental Validation.

A. Budhiraja

Department of Statistics

F. Hernández-Campos

Department of Computer Science

V.G. Kulkarni

Department of Operations Research

F. D. Smith

Department of Computer Science

University of North Carolina

Chapel Hill, N.C. 27599

March 26, 2003

1   Introduction

The dominant transport protocol for the Internet is TCP (Transmission Control Protocol). The applications using this protocol range from simple email exchanges to web browsing and live audio or video streaming. Approximately 90% of the data carried by the Internet are governed by TCP. TCP provides an end-to-end (application-to-application) service that ensures the reliable, in-order, and unduplicated delivery of bytes flowing from a sender to a receiver. The mechanisms (sequence numbers explicitly acknowledged by the receiver, delivery timers, and retransmissions) used in TCP for providing this reliable data transport also provide a basis for end-to-end flow- and congestion-control mechanisms. Flow control is necessary to ensure that the sending application does not send data at a rate exceeding the rate at which the receiving application can process it. Congestion control is necessary to ensure that the sending application does not send data at a rate exceeding the currently available transmission capacity along the network path from the sender to the receiver.

Engineering experience has shown that the vast majority of lost data occurs because finite-capacity buffers overflow and arriving data is discarded at intermediate routing nodes in the Internet. TCP, therefore, treats the detection of lost data as a signal of congestion along the Internet path from the sender to the receiver. Buffer overflows mean that the arrival rate of in-bound data at a router destined for one of its out-bound links exceeds the transmission rate of that link. The overload may be a transient condition caused by bursty traffic demands or a persistent condition caused by poor network capacity planning. In response to lost data as an indication of congestion, the TCP at the sender reduces the maximum rate at which the application can send data. When there is no indication of congestion, TCP allows the sending rate to increase so it can accommodate the nominal rate used by the application.

Both flow and congestion controls are accomplished with a window-based transmission mechanism. A "window" is simply an amount of data the sender is allowed to have in transit in the network (sent but not yet acknowledged by the receiver) at any point in time and the window can be increased or decreased to govern the sender's transmission rate. The multi-byte data transmission unit created by TCP is called a segment and window size is often expressed as a number of maximum size TCP segments. For flow control, the receiving TCP explicitly informs the sending TCP how much additional data beyond the segment being acknowledged it has buffers available to hold. This amount is the receiver-advertised window. A second window, the congestion window, is regulated in response to indications of congestion (or no congestion). At any time, the TCP sender is restricted to the minimum of the receiver-advertised window and the congestion window. The explicit increase and decrease algorithms used by TCP to adjust these windows are described in section 3.

Because TCP is the dominant mechanism controlling the flow of data in the Internet, there has been intense research interest in producing good models of its dynamic window-adjustment behaviors (and therefore its control of transmission rates and application throughputs). These

behaviors can be extremely complex depending on the specific network conditions. A brief review of this research is given below. The practical applications for a model of TCP behavior have become more numerous in the recent past. TCP congestion-control mechanisms have been successful in allowing the Internet to handle an unanticipated scale of growth without suffering collapse in periods of congestion. It has, therefore, become the de facto standard for evaluating other congestion control mechanisms. In particular, the other transport protocol for the Internet, UDP (User Datagram Protocol), does not provide any mechanisms for reliable data delivery or congestion control. It is up to applications that use UDP, notably those having multimedia data streams that require periodic real-time data delivery, to provide their own response to congestion conditions.

There is considerable concern in the Internet engineering community that UDP applications do not meet the de facto standard set by TCP for controlling congestion and that growth in their usage will lead to congestion collapse in the Internet. This has lead to various proposals (e.g., [19]) for monitoring and policing UDP flows to ensure they provide an effective level of congestion control. The basic notion is that routers can monitor their link queues to determine the rate of discarded data and can compare the data arrival rate from UDP flows with the result from a model of TCP throughput at the same loss rate. If the UDP flow persistently sends data at a higher rate, the router can subject it to a "penalty box" in which a much higher fraction of its data is discarded. A related application for a TCP model is where a UDP application monitors its own rate of dropped data (with feedback from the receiver about those that don't arrive) and adjusts its sending rate to that determined by a model of TCP at the same loss rate (a so-called "TCP-friendly" behavior called "equation-based" congestion control [20] where the equation is an analytic TCP model). Recently, it was proposed in [22] to use a model of TCP performance at a certain loss rate to write contractual service agreements between an Internet service provider and its customers.

2   Literature Review

The large number of publications on modeling TCP behavior is a statement both about the difficulty of constructing a robust model of such complex behavior and the importance of TCP congestion control to the Internet. It is impossible for space reasons to comment on all these models but we do review the most prominent examples from several approaches used. All the models described below consider the dynamic behavior of a single TCP connection as it would respond to loss events as indicators of link congestion. At a high level, the models are distinguished by three characteristics: the model of loss events (independent vs correlated), the aspects of TCP mechanisms assumed to have negligible influence on the results and not considered in the model, and whether the analysis computes expected values (e.g., the TCP window size or throughput) or produces a probability distribution of window sizes. An excellent discussion of many important issues in TCP model construction and validation can be found

3

in [11].

The important distinctions between our model and the ones reviewed below are that it avoids certain common simplifying assumptions (e.g., an unbounded window size or loss rates independent of window size) and it gives the probability distribution of window sizes.

2.1 Models with independent losses

[32] presents an early derivation of a simple stochastic model of steady-state TCP behavior in its congestion-avoidance mode (explained in more detail in section 3). This model produced the well-known result that the mean TCP window size is $C/\sqrt{p}$ where p is the probability that an arbitrary segment will be independently dropped in the network and C is a constant term that reflects details of the TCP acknowledgment algorithm ($C = 1.22$ if every segment is acknowledged and $C = 0.87$ if at least every other segment is acknowledged). Given this mean window size estimate, throughput is bounded by (MSS/RTT)*$(C/\sqrt{p})$, where MSS is the maximum segment size and RTT is the mean round-trip time of the connection. This model is often called the "periodic-drop" model because the underlying analysis assumes that a constantly sending TCP source emits $1/p$ segments between segment drops. An even earlier model derivation that is closely related can be found in [17, 18].

The form of the model is to compute the mean window size based on its evolution as a series of discrete segment-by-segment events. The model only captures TCP window adjustments associated with TCP's congestion avoidance phase and only for segment drops that can be recovered without a timeout by reacting to three duplicate acknowledgments. Key elements of TCP behavior not considered in this model are the effects of timeout periods (in which throughput is 0), the TCP connection-establishment latency, the time TCP spends in its slow-start mode (also explained in section 3), either following connection establishment or after a loss recovered by timeout, multiple losses in a single window of segments (that typically results in a timeout unless selective acknowledgment (SACK) is used), throughput limits imposed by the size of the receiver's advertised window, and delayed acknowledgments. Because these elements are not considered, the duration (latency) of the connection cannot be derived for a given size of data transfer.

[27, 28] present an analysis of the mean throughput of a TCP connection that was developed concurrently with (and apparently independently of) [32] and [37]. Two cases are modeled. The first is one in which losses are not random but instead caused when the connection's window grows to a point that causes a segment to be lost because the capacity of the buffer at the bottleneck link is exceeded (assuming only the one connection is using the link). This analysis considers both slow-start and congestion-avoidance phases. The second case considers both losses from overflow at the bottleneck buffer and random losses (from other causes) where the loss events are independent with fixed per-segment probability. Only congestion-avoidance

with fast retransmission (no recovery with timeouts) is considered in this analysis. In both cases, connection establishment, receiver window limits, timeout intervals (0 throughput), and delayed acknowledgments are not modeled. [29] extends the analysis in [27] and [28] to include asymmetric delays and congestion-induced loss on the "forward" (data) and "reverse" (acknowledgment) paths.

[1] builds on the model and analysis from [27] to include loss models for IID independent losses and also a two-state (loss, no-loss) first-order Markov model for the per-segment error process to include effects of correlated loss. The model also accounts for the effects of loss recovery by timeouts (including exponential back-off of the interval timer). Otherwise, it has the same assumptions and limitations as [27]. [2] gives an earlier version of this model that considers only IID losses.

[26] describes a Markov renewal-reward process model assuming independent segment losses. The model takes into account the explicit details of several versions of TCP (Tahoe, Reno, NewReno) along with factors such as timeout values and their granularity, receiver window size, and the number of duplicate acknowledgments used to infer loss. The detailed evolution of the window size for each TCP version is examined to determine the probability that multiple losses can be recovered without a timeout and the probability of recovery by timeouts. [49] presents a modeling approach similar to [26] but using a two-state (loss, no-loss) first-order Markov model for the per-segment error process to include effects of correlated loss.

[38, 39] is the most widely cited and influential of the early TCP models. The new considerations in this model account for the effects of error recovery by retransmission after a timeout (given a mean timeout duration), the limits imposed by the maximum receiver's window (assumed to be constant at the value advertised at connection establishment) and the number of segments covered by an acknowledgment. The model of segment loss defines a probability, p, that a segment is lost given that no preceding segment from the window is lost (i.e., the probability that a given segment becomes the first loss within a window). Loss in one window is assumed independent of loss in any other window. To approximate the effects of correlated losses, if one segment from a window is lost, all following segments are considered lost also. Two assumptions also made explicit are that RTT and window size are independent, i.e., window size does not influence queueing delays (or that RTT consists only of propagation delays), and that the time to transmit all segments in a window is less than the RTT.

Construction of the [38] model proceeds from a detailed segment-by-segment analysis for the evolution of the TCP window as a stochastic process (Markov regenerative process with rewards). The result is an expression for the long-term expected value of throughput in segments/unit-time, B, expressed as $E(Y)/E(A)$ where Y is the number of segments sent in intervals of time, A, between loss events of different types using the mean RTT as a measure of time. The model equation includes a term that expresses the $C/\sqrt{p}$ relationship between mean window size and loss rate (e.g., the models of [32] and [37] are special cases of this model). They note also that for larger values of p, TCP window behavior is better approxi-

mated by C/p than by $C/\sqrt{p}$. The validity of this model is established with extensive empirical measurements of TCP connection throughputs along with the loss rates experienced.

In [22] the model of [38] is revised and extended. The primary motivation was to express the loss model in terms of independent random loss (Bernoulli) model with intensity, p, which can be determined from conventional router-maintained counters. Using this model, they derive an expression for the probability that i segments will be lost from window of size W given independent losses, replacing the assumption that all segments following the first loss in a window are also lost (an approximation to correlated loss). From this, they also derive a new estimate of the probability of a timeout. A further refinement is to model the fact that the window size following loss events depends on the number of losses occurring in a window of size W, replacing the assumption in [38] that the window is reduced to $W/2$. Other than these two refinements, the model has the same assumptions and limitations as [38].

[13] significantly extends the model in [39] to consider the effects of TCP connection establishment (including the possibility of segment loss), throughput during the initial slow-start phase (but not slow-start following a timeout for a loss event), and the timing effects of delayed acknowledgments. Because the model is an extension of [39] the same assumptions and parameters are used. Another novel aspect of this model is that it is expressed in terms of expected values for connection latency given a size of data transfer (from which a throughput rate over the entire latency interval can also be calculated). Note that this allows the model to be accurate for transfers of any size including those that are quite short (never encounter a loss or leave slow-start). This model is validated through comparison with NS simulations and live Internet measurements.

The models presented in [44, 45] are based on the same assumptions and some of the analysis and equations (such as approximations for correlated losses) from [13] and [39]. Their analysis and model differs from those earlier models because it is based on determining how many segments from an N-segment transfer are transmitted in each of the slow-start phase (both the initial one and following a loss event indicated by timeout) or in the congestion-avoidance phase as a function of the number of loss events. The approach in their analysis is somewhat similar to [26] in which the detailed evolution of the window size is examined to determine if multiple losses can be recovered without a timeout. They also develop a more refined model of the slow-start phase that accounts for effects of the delayed-acknowledgment timer. In [45] the analysis is refined to include specific details that reflect differences among TCP Tahoe, Reno, and SACK algorithms. They claim this model is more accurate for estimating total latency (and throughput) than [13] for short transfers but found little difference for transfers of more than 10,000 bytes. Results of empirical measurements (both new measurements and those in [38]) are presented to support this claim.

[47] presents a generalization of [38] model to include two additional parameters, α and β, where α is the rate of increase in the window (measured in segments) when there is no loss in an RTT and β is the fraction of the current window that exists following a loss. In all other

aspects their model is identical to [38] and required the same assumptions and has the same limitations related to elements of behavior not reflected in the model.

[34] presents a model of steady-state TCP throughput. The new result is a model of window size behavior as a Poisson-Counter driven Stochastic Differential Equation where the loss events are treated as a Poisson stream of arrivals with rate λ at the source (instead of modeling segments as being sent out but lost with a given fixed probability) and the window size is approximated as a fluid (with continuous changes). Like [38] the model accounts for timeouts, receiver window size, and number of segments covered by an acknowledgment (but not connection establishment, initial slow-start, and delays in returning acknowledgments). The result is an expression for the expected values of TCP window size and throughput as functions of the arrival rates of timeouts and duplicate-acknowledgment loss events, RTT, the mean timeout delay, and the maximum window size. There is no $C/\sqrt{p}$ term in this model but it was shown that the model includes those such as [32] as a special case. The model was also shown to produce results comparable to or better than [38] when compared to the same empirical measurements.

Stochastic differential equation models for TCP throughput have also been considered in [25]. The authors consider a network with multiple resources and users. The throughput rate for the r-th user increases additively in absence of loss signals and decreases multiplicatively at rate proportional to the stream of feedback signals recieved from the various resources. The stream of feedback signals is modelled either via a Poisson process (with a stochastic rate function) or a diffusion process. The rate of the Poisson process (or the governing parameters in the diffusion), for a given resource, depend on the load on the resource. In that respect the model is quite similar to the model studied in the current paper, however, the authors do not take into account the finiteness of the reciever window size or timeouts. Furthermore, they do not obtain the asymptotic distribution of the window size or throughput rate.

[36] used simulations to explore the implications of some assumptions common to a number of the analytical models (loss model, no lost acknowledgments, etc.). They examined the distribution of latencies for transfers of a given number of segments, N, and found that the shape of the distribution function was largely invariant for difference scenarios but was shifted or scaled by linear transformations. They suggest that this indicates that analytic models can be extended to compute distributions of latencies not just first and second moments.

2.2 Models with correlated losses

The emphasis in [3] is on improving the realism of the loss models. In particular, the expressions derived include a term for the correlation function of the inter-loss times in addition to the usual parameters for the loss rate, RTT, number of segments per acknowledgment, the maximum receiver window, the mean timeout, and the window increase/decrease rates resulting from

7

losses indicated by duplicate acknowledgments or timeouts. Loss processes can then be varied in the models by specifying the correlation function of inter-loss times. In terms of the elements of TCP behavior represented, this model is equivalent to [39] and [34]. It is also shown that for IID random losses the $C/\sqrt{p}$ result is a special case. An extensive set of empirical measurement is used to show that a variety of loss models are necessary to capture the effects found on real Internet paths and that the proposed model produces good results for different patterns of loss actually observed. Other aspects of their work on TCP performance with correlated losses can be found in [4, 5].

[8] uses a two-state (loss, no-loss) first-order Markov model for the per-segment error process to include effects of correlated loss. Five versions of TCP algorithms (from Tahoe to SACK) are modeled during steady-state data-transfers. The TCP elements not considered are the initial connection-establishment phase and exponential back-off of the timeout value for multiple consecutive losses. The detailed analysis considers the number of segments sent as a "train" during each interval during the evolution of the TCP window (in both slow-start and congestion avoidance phases) between loss-recovery events and produces an expression for the expected value of throughput.

2.3   Models with a distribution of window sizes

The computation of the stationary distribution of the window size has not received as much attention in the literature as the expected window size. The early model in [37] (apparently never published) is the first use of a fluid approximation to the discrete TCP window-evolution process. It assumes that segment losses are independent events with probability p. Besides the use of a fluid approximation, the other novel contribution is to give the stationary distribution function (actually the complementary CDF) for the window size process. The mean of the window-evolution process is derived from this distribution with the final result again $C/\sqrt{p}$ with a slightly different value of $C = 1.31$. The same elements of TCP behavior not considered in [32] are also not considered in this model.

[33] presents a generalization of the model from [37] to consider segment loss probability that varies as a non-decreasing function of the TCP window size instead of being a constant probability. The effects of delayed acknowledgments are also modeled but otherwise this model omits all the same elements of TCP behavior as [37] and [32], especially the limits imposed by the receiver's window and the impact of timeouts. As in [37] the result is expressed in terms of computing the cumulative distribution function of the TCP window size in number of TCP segments. No throughput results are given but can presumably be computed from the distribution of window sizes given round-trip times. The results of computing window sizes with the model are shown to compare favorably with results from NS simulations.

The models described in [14, 15] are closely related to [33] and [37] and mostly consider

only those aspects of TCP behavior modeled in [32]. Specifically they consider only idealized TCP steady-state behavior in congestion avoidance mode with independent segment losses of constant probability and no timeouts. They do, however, model the effects of the limitations imposed by the receiver's window and the increase/decrease ratios are parameterized as in [47]. The major contribution of these papers is to provide convergence results for the distribution of window sizes (and throughput) for small loss probabilities ($p \rightarrow 0$). A subset of the limit results for certain special cases is given in [14].

In both [33] and [15] the authors show that a time and space scaled version of the window size process converges in distribution to a Markov process, which can then be analyzed by the standard methods to obtain the limiting distribution. The general nature of the loss function in [33] forces them to write the limiting distribution as an infinite sum of increasing dimensional multiple integrals, which can be evaluated numerically. The analysis in [15] is closer to our analysis. They derive the limiting distribution as an infinite sum of hypo-exponential pdfs, while we get finite sums of simpler functions. Their approach seems to hide the interesting features of the limiting distribution, such as the atom at zero, and discontinuities in the pdf, that we have illustrated in this paper.

Our model is described in more detail in Section 3. It takes into account recovery from packet losses with both fast recovery and timeouts, boundary behavior at zero and maximum window size, and slow-start after timeouts. As far as we know ours is the first model that accounts for all these aspects of TCP and also provides a distribution of window sizes.

2.4   Network-level Models of TCP

In addition to research on models that yield the throughput of a single TCP connection for a particular packet loss process, there is also an active line of research on models that include more of the factors that influence TCP behavior directly in the model. The approaches that fall in this category generally consider multiple TCP connections that share network resources (link capacity, router buffers) and model their effects on each other's performance and on the utilization of the network resources. Many of these models also include queueing effects and, in particular, the behavior of the queue management algorithms at routers that ultimately determine how segments are dropped when link congestion occurs. Examples of models that consider elements of the entire network when modeling TCP connections are: [6], [7], [9], [10], [21], [23], [24], [30], [31], [35], [41], and [43]. We do not discuss these further because our work presented here falls in the category of single TCP connection models.

3 The TCP Dynamics.

We consider a single application instance sending data over the Internet using one TCP connection. We briefly describe the features of TCP that are relevant to the models developed here. TCP implements window-based protocols that control the rate at which segments are transmitted by the sender to the receiver. For each connection TCP maintains two variables: congestion window size and receiver window size. Both are specified in bytes, but since we assume that all segments have mean size B, we assume that the window sizes are in number of segments of size B. The minimum of the two window sizes is called the effective TCP window size, or just window size.

Let $W_C(t)$ be the congestion window size (in number of segments) at time t, $W_R(t)$ be the receiver's window size at time t, and RTT be the mean round trip time (in seconds) for the TCP connection. The receiver's window size is the number of segments the receiver has space for, and is typically a fixed quantity K. It changes rarely, due to the high speed at which the receiver's buffer can be emptied as compared to the speed at which the segments are received. Similarly, the round trip time and the segment sizes may vary over the duration of the connection, but we shall use their mean values in the analysis and define B as the mean segment size and RTT as the mean round-trip time over the connection duration. The TCP window size is given by

$$W(t) = \min\{W_C(t), W_R(t)\} = \min\{W_C(t), K\}.$$

TCP congestion control has two phases: slow-start and congestion avoidance. In the congestion avoidance phases, TCP behaves roughly as follows: At time t, the sender transmits $W(t)$ number of segments back to back. As soon as the receiver receives a segment, an acknowledgment for that segment is sent back to the sender. (We ignore the batching of acknowledgments that is done by some TCP versions.) Thus by time $t + RTT$ the sender receives $W(t)$ acknowledgments, if there are no problems in transmission and delivery of the segments.

If the segment is error free and is in the correct sequence, the receiver sends an acknowledgment for the highest sequence number received correctly. If a segment has unrecoverable errors or is out of sequence, the receiver continues to send the highest sequence number received correctly in its acknowledgment (sends a duplicate acknowledgment). A loss is indicated if the sender receives 3 duplicate acknowledgments or if the timeout period for a segment expires without an acknowledgment for its sequence number.

The congestion window $W_C(t)$ changes dynamically at every instant an acknowledgment is

received. In the absence of any loss indications, the window size increases by one every round trip time. If an acknowledgment is received that is a third duplicate acknowledgment (indicating a segment loss) the window is reduced to $(1-\alpha)W(t)+1$ (typically $\alpha = .5$), and the TCP protocol continues in the congestion avaidance phase. This operation is called fast recovery. We use $(1-\alpha)W(t)+1$ in place of the more customary $(1-\alpha)W(t)$ for two reasons: TCP's congestion window is never reduced below 1 segment, and increasing the window slightly is an approximation to account for the segments sent during fast recovery before the window is reduced to its final value. If a timeout occurs, the window size is reduced to one. After timeout the TCP protocol switches to the slow-start phase and rapidly increases the window size to one plus $(1-\alpha)$ fracton of the prevailing window size just before the timeout occurred. Again, there are many variations of TCP; we shall analyze this one. Note that the response to a segment loss is determined by $W(t)$ and not $W_C(t)$.

We shall study the dynamics of the window size process $\{W(t), t \geq 0\}$ in two steps. In the next section we assume that there are no timeouts and develop the probabilistic analysis of the window size process in steady state. In the follwing section we extend this analysis to include the timeouts.

4   TCP Model with No Timeouts

As a first step towards the analysis of the complete TCP dynamics, we begin with the assumption that there are no timeouts. Thus, once the TCP gets out of the slow-start phase, it stays in the congestion-avoidance phase forever. Assume that each returning acknowledgment indicates a segment loss with probability $\hat{p}$, independent of anything else. Let $X(t)$ be the number of acknowledgments received during $(t, t + RTT)$ that indicate segment losses. With the assumption of independence we see that $X(t)$ is a Binomial$(W(t), \hat{p})$ random variable. We assume that $\hat{p}$ is sufficiently small so that we can ignore the possibility that $X(t) \geq 2$, and thus $X(t)$ can be taken to be a Bernoulli random variable with

$$P(X(t) = 1) = \hat{p}W(t), \quad P(X(t) = 0) = 1 - \hat{p}W(t).$$

Thus, in effect, we assume that there is at most one segment loss per RTT. Now, as long as $0 < W_C(t) < K$, we have $W(t) = W_C(t)$, and the dynamics of the TCP protocol can be approximated as

$$W(t + RTT) = (W(t) + 1)(1 - X(t)) + (1 + (1 - \alpha)W(t))X(t).$$

Rearranging, we get

$$W(t + RTT) - W(t) = 1 - \alpha W(t)X(t). \tag{4.1}$$

Hence we get

$$W(t + RTT) - W(t) = \frac{1}{RTT} RTT - \alpha W(t) X(t)$$
$$= f * RTT - \alpha W(t) X(t) \tag{4.2}$$

where

$$f = \frac{1}{RTT}.$$

Now, we approximate the distribution of $X(t)$ by that of $N(t+RTT)-N(t)$, where $\{N(t), t \geq 0\}$ is a Poisson Process with a time varying stochastic rate function $pW(t)$ where $p = \hat{p}/RTT$. Thus Equation 4.2 can be written as

$$W(t + RTT) - W(t) = f * RTT - \alpha W(t)(N(t + RTT) - N(t)).$$

The above equation can be thought of as the finite difference version of the following stochastic differential equation for $\{W(t), t \geq 0\}$:

$$dW(t) = f dt - \alpha W(t) dN(t), \quad 0 < W(t) < K. \tag{4.3}$$

Next we modify the above equation to account for the boundary behavior at $W(t) = K$. Once $W(t)$ reaches K, $W_C(t)$ may keep changing, but $W(t)$ stays at K until a segment loss occurs. We incorporate this condition by modifying the stochastic differential equation 4.3 as follows:

$$dW(t) = f 1_{\{W(t) < K\}} dt - \alpha W(t) dN(t), \quad 0 \leq W(t) \leq K. \tag{4.4}$$

The above stochastic differential implies that the W process increases continuously at a constant rate f as long as it is less than K and there are no events in the N process. When it hits the upper limit K, it stays at K. When an event occurs in the N process, the W process jumps down by a factor $1 - \alpha$. Figure 1 shows a typical sample path of the W process.

5 Limiting Behavior of W with No Timeouts

In this section we study the limiting behavior of the TCP window size, assuming that it evolves as described in the previous section. That is, we obtain the limiting distribution of the solution to the Equation 4.4 as $t \to \infty$. Let

$$F(x) = \lim_{t \to \infty} P(W(t) \leq x), \quad 0 \leq x \leq K,$$

be the limiting distribution of the $W(\cdot)$ process. The sample path of the W process shows that the limiting distribution of W has a probability mass v_K at $x = K$ and a probability density $u(x)$ over $x \in [0, K)$. Thus

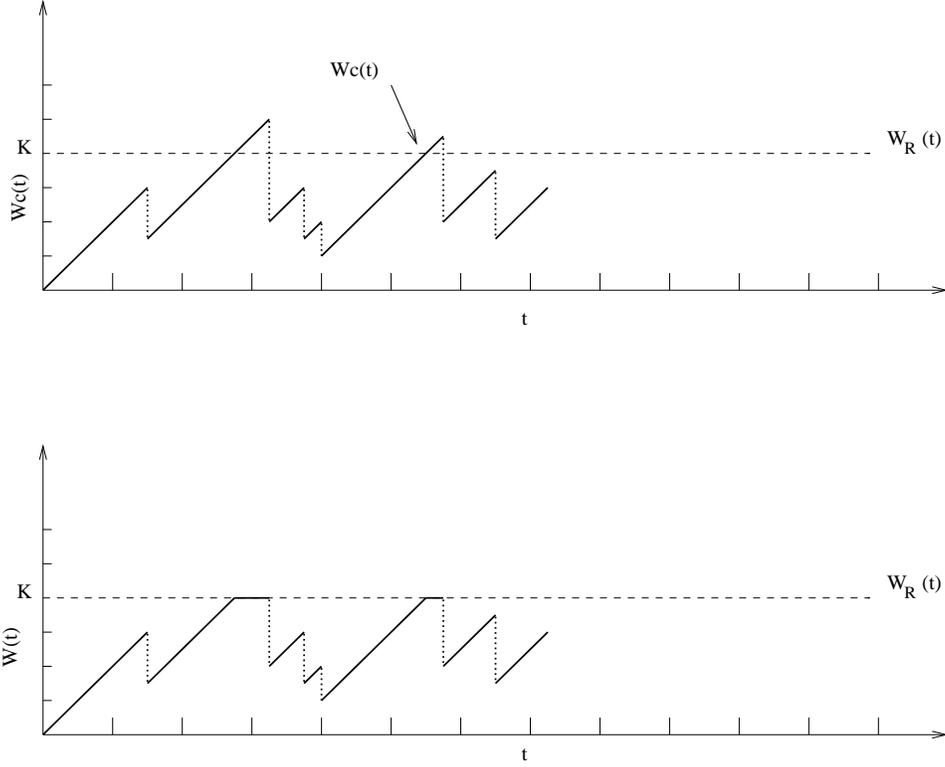$$F(x) = \int_0^x u(y) dy, \quad 0 \leq x < K,$$

12

Figure 1: Sample paths of W and W_C processes in the absence of timeouts.

$$F(K) = \int_0^K u(y)dy + v_K = 1.$$

The next theorem gives v_K and $u(\cdot)$.

Theorem 5.1 *The density $u(x), 0 < x < K$ is given by,*

$$u(x) = a_0 \sum_{r=0}^{n} \alpha_{n-r}\beta_r g_r(x), \quad x \in (K(1-\alpha)^{n+1}, K(1-\alpha)^n), \quad n \geq 0 \qquad (5.5)$$

where a_0 is a normalizing constant, and

$$g_r(x) = \exp\{\frac{-px^2}{2f(1-\alpha)^{2r}}\}, \quad x \geq 0, \ r \geq 0, \qquad (5.6)$$

and the constants $\{\alpha_i\}$, $\{\beta_i\}$ are as below:

$$\beta_0 = 1, \quad \beta_{r+1} = \frac{(1-\alpha)^{2r}}{(1-\alpha)^{2r+2} - 1}\beta_r; \quad r \geq 0; \qquad (5.7)$$

13

$$\alpha_0 = 1,$$

$$\alpha_1 = \frac{1}{g_0(K(1-\alpha))}\left(g_0(K(1-\alpha)) - (1+\beta_1)g_1(K(1-\alpha))\right),$$

$$\alpha_{r+1} = \frac{1}{g_0(K(1-\alpha)^r)}\sum_{j=0}^{r}\alpha_j\left(\beta_{r-j}g_{r-j}(K(1-\alpha)^{r+1}) - \beta_{r+1-j}g_{r+1-j}(K(1-\alpha)^{r+1})\right) \tag{5.8}$$

for $r > 0$. The probability mass v_K is given by

$$v_K = \frac{fu(K-)}{pK}. \tag{5.9}$$

Finally, the normalizing constant a_0 is chosen so that

$$\int_0^K u(y)dy + v_K = 1.$$

Proof: Since W is an irreducible Markov process on $[0, K]$, it has a stationary distribution F. Let $W(0)$ have distribution F. Then $W(t)$ has distribution F for all $t \geq 0$. Then, for $0 < x < K$,

$$\begin{aligned}
F(x) &= P(W(h) \leq x)\\
&= \int_{[0,K]} P(W(h) \leq x \mid W(0) = y)dF(y)\\
&= ph\int_{[0,\min(\frac{x}{(1-\alpha)},K)]} ydF(y) + \int_{[0,x-fh]}(1-phy)dF(y) + o(h)\\
&= ph\int_{[0,\min(\frac{x}{(1-\alpha)},K)]} ydF(y) + F(x-fh) - +ph\int_{[0,x-fh]}ydF(y) + o(h).
\end{aligned}$$

Thus we have that

$$\frac{F(x) - F(x-fh)}{h} = p\int_{[0,\min(\frac{x}{(1-\alpha)},K)]} ydF(y) - p\int_{[0,x-fh]}ydF(y) + o(1).$$

Taking limit as $h \to 0$ we have that

$$fu(x) = p\int_{[0,\min(\frac{x}{(1-\alpha)},K)]} ydF(y) - p\int_{[0,x)}ydF(y); \quad x \in (0, K]. \tag{5.10}$$

In particular, we have that

$$fu(K-) = pKv_K$$

which yields equation 5.9.

Consider now the case when $x \in I_0 \equiv (K(1-\alpha), K)$. Then from (5.10) it follows that

$$fu(x) = p \int_{[0,K]} y dF(y) - p \int_{[0,x)} y dF(y).$$

Differentiating we have that

$$fu'(x) = -pxu(x),$$

and solving the above equation we get

$$u(x) = a_0 g_0(x),$$

where a_0 is some constant. Thus, we have shown (5.5) for the case $n = 0$. We will prove (5.5) for a general n via induction. Now suppose that (5.5) holds for $n = 0, 1, \cdots, j$. We begin by observing that from (5.10) it follows that for $x \in [0, K(1-\alpha))$

$$u'(x) = p \frac{px}{f(1-\alpha)^2} u(\frac{x}{1-\alpha}) - \frac{px}{f} u(x). \tag{5.11}$$

Thus from the induction hypothesis it follows that for $x \in (K(1-\alpha)^{j+2}, K(1-\alpha)^{j+1})$

$$fu'(x) = p \frac{x}{(1-\alpha)^2} \left\{ a_0 \sum_{r=0}^{j} \alpha_{j-r} \beta_r g_r(\frac{x}{1-\alpha}) \right\} - pxu(x).$$

We can solve the above differential equation to get

$$
\begin{aligned}
u(x) &= \frac{pa_0}{(1-\alpha)^2} e^{-\frac{px^2}{2f}} \sum_{r=0}^{j} \alpha_{j-r} \beta_r \int x g_r(\frac{x}{(1-\alpha)}) e^{\frac{px^2}{2f}} dx + a_{j+1} e^{-\frac{px^2}{2f}} \\
&= \frac{a_0}{(1-\alpha)^2} \sum_{r=0}^{j} \alpha_{j-r} \beta_r g_{r+1}(x) \left(\frac{(1-\alpha)^{2r+2}}{(1-\alpha)^{2r+2} - 1} \right) + a_{j+1} e^{-\frac{px^2}{2f}} \\
&= a_0 \sum_{r=0}^{j} \alpha_{j-r} \beta_{r+1} g_{r+1}(x) + a_{j+1} \beta_0 g_0(x), \tag{5.12}
\end{aligned}
$$

where a_{j+1} is some constant.

In order to determine a_{j+1}, we now consider the case $j = 0$ and $j \geq 1$ separately.
Case 1: $j = 0$: In this case we have from (5.10) that

$$u(K(1-\alpha)+) = u(K(1-\alpha)-) + \frac{pKv_K}{f}.$$

Using this observation, the induction hypothesis and (5.12), we have now that

$$a_0 \beta_1 g_1(K(1-\alpha)) + a_1 g_0(K(1-\alpha)) = a_0 g_0(K(1-\alpha)) - \frac{pKv_K}{f}.$$

15

Thus solving for a_1, we get

$$a_1 = \frac{a_0}{g_0(K(1-\alpha))} \left(g_0(K(1-\alpha)) - (\beta_1 + 1)g_1(K(1-\alpha))\right)$$
$$= a_0\alpha_1.$$

Case 2: $j \geq 1$**:** Next note that from (5.10) it follows that $u(\cdot)$ is continuous on $(0, K(1-\alpha))$, thus if $j \geq 1$ then $u(K(1-\alpha)^{j+1}+) = u(K(1-\alpha)^{j+1}-)$. Using this observation, (5.12) and the induction hypothesis we have that

$$a_0 \sum_{r=0}^{j} \alpha_{j-r}\beta_{r+1}g_{r+1}(K(1-\alpha)^{j+1}) + a_{j+1}\beta_0 g_0(K(1-\alpha)^{j+1}) = a_0 \sum_{r=0}^{j} \alpha_{j-r}\beta_r g_r(K(1-\alpha)^{j+1}).$$

Solving for a_{j+1} and rearranging the terms we have now that

$$a_{j+1} = \frac{a_0}{g_0(K(1-\alpha)^{j+1})} \sum_{r=0}^{j} \alpha_r \left\{\beta_{j-r}g_{j-r}(K(1-\alpha)^{j+1}) - \beta_{j+1-r}g_{j+1-r}(K(1-\alpha)^{j+1})\right\}$$
$$= a_0\alpha_{j+1}.$$

Substituting the above observations in (5.12), we have (5.5) for the case $n = j+1$. This proves the result.

The typical form of the limiting distribution is shown by the upper graph in figure 2. Note that $u(\cdot)$ has a jump discontinuity at $x = K(1-\alpha)$, and its derivative has a jump discontinuity at $x = K(1-\alpha)^2$. The lower graph shows u^T, the limiting density of of the window size in the presence of timeouts, which is studied in the next sectiuon.

6 TCP Model with Timeouts

The above analysis does not account for the response of TCP for lost segments that will be detected by timeouts. Typically such events are accompanied by intervals of zero throughput for a random amount of time. The reason is that the sender at some point reaches the limit of its current effective window but the lost segment's timeout interval (which depends on an estimated RTT and thus behaves like a random variable) has not occurred to initiate recovery and start sending again. We make the assumption that a fixed fraction c of the events in the N process are timeouts; while the remaining $(1-c)$ fracton of the events are fast recovery events. We assume that the effective window size jumps down to zero in the event of a timeout and it stays zero for an exponential duration with mean $1/\mu$, the expected duration of the timeout interval. We assume that after this zero window-size interval, the window size jumps instantaneously to $(1-\alpha)W^-$, where W^- is the window size when the timeout occurred. In other words, we approximate the rapid exponential increase in the window size during the slow-start phase by
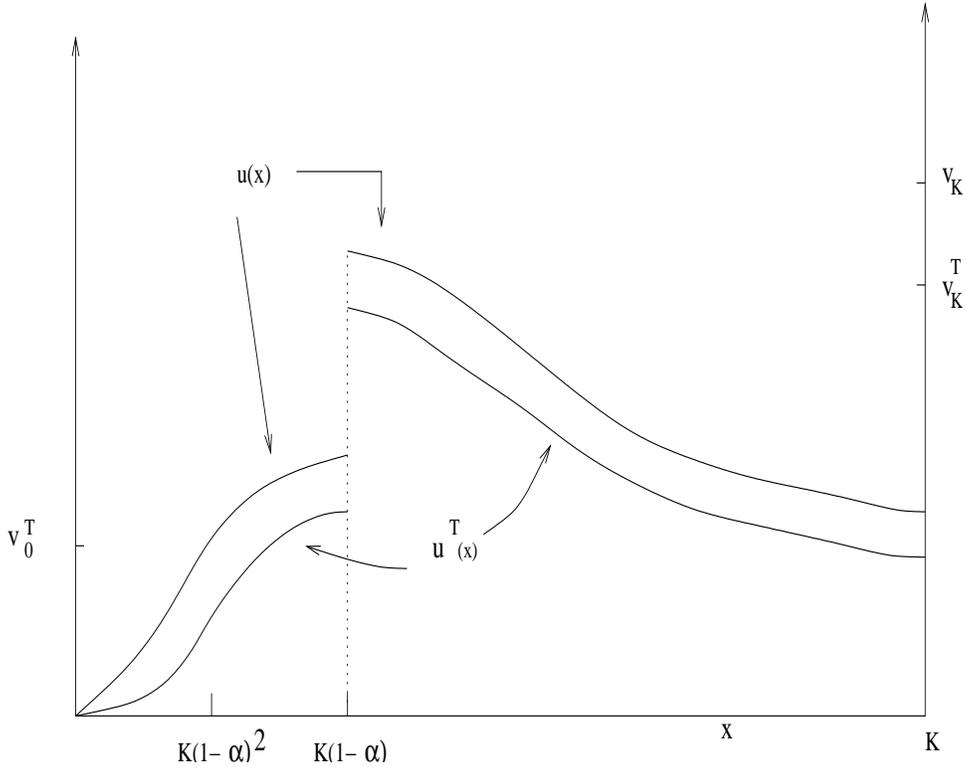
Figure 2: Typical graph of the density u and u^T of the window size.

an instantaneous jump at the end of the timeout period. The window size then begins to evolve according to Equation 4.4 until the next loss event is encountered. We denote the window size process in the presence of timeouts by $\{W^T(t), t \geq 0\}$, to distinguish it from the window size process $\{W(t), t \geq 0\}$ without the timeouts. Figure 3 shows the sample paths of $W^T(t)$ and $W_C(t)$ and their relation to each other, taking into account the fast recovery and the timeout events, and the boundary behavior at K and 0. It is clear the sample paths of the W^T process can be thought of as the sample paths of the W process where periods of zero window size are inserted after a fraction c of the downward jumps in the W process in a completely random fashion. This observation implies that there is a very close relation between the limiting cdf of the W process and that of the W^T process. This relation is made clear in the following theorem.

The sample path of the W^T process shows that the limiting distribution of W^T has a probability mass v_0^T at $x = 0$, v_K^T at $x = K$ and a probability density $u^T(x)$, $0 < x < K$. Let

$$F^T(x) = \lim_{t \to \infty} P(W^T(t) \leq x), \quad 0 \leq x \leq K,$$

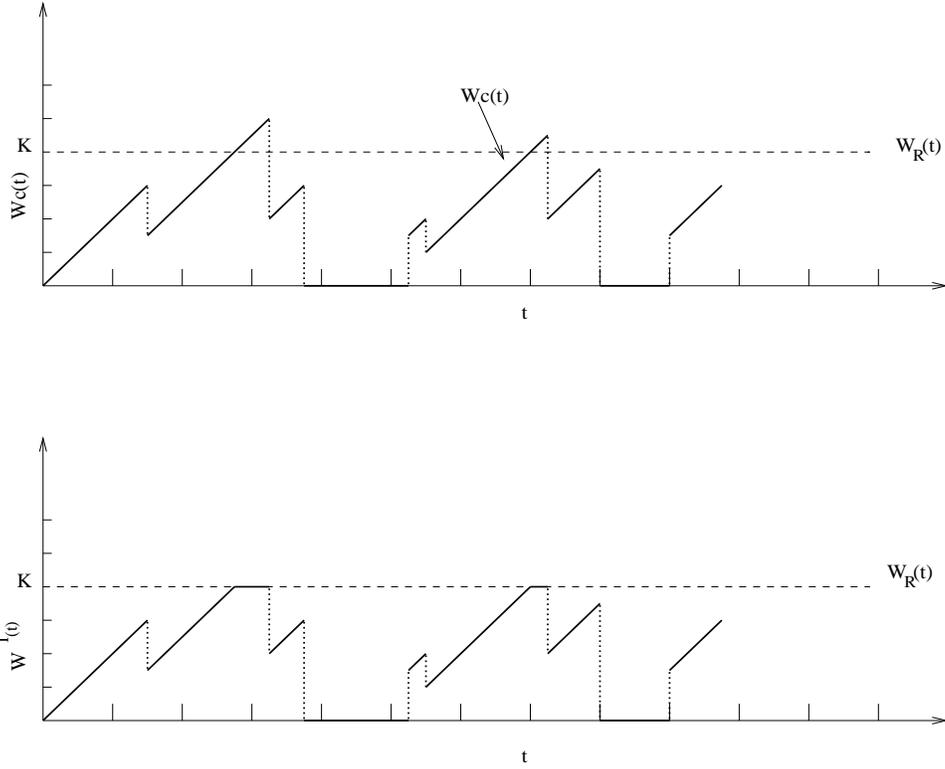be the limiting distribution of the $W^T(\cdot)$ process.

Figure 3: Sample paths of $W_c(t)$ and $W(t)$ in the presence of timeouts.

The next theorem gives v_0^T, v_K^T and $u^T(\cdot)$ in terms of the quantitiies v_K, $u(\cdot)$ of Theorem 5.1 and

$$m = \int_0^K yu(y)dy + Kv_K.$$

Theorem 6.2 *We have*

$$v_0^T = \frac{mpc}{\mu + mpc}, \tag{6.13}$$

$$u^T(x) = (1 - v_0^T)u(x), \quad 0 < x < K, \tag{6.14}$$

$$v_K^T = (1 - v_0^T)v_K. \tag{6.15}$$

Proof: Let

$$F^T(x) = \lim_{t \to \infty} P(W^T(t) \le x), \quad 0 \le x \le K.$$

18

Since the window size is zero when the timeout is in progress, we see that

$$\lim_{t\to\infty} P(W^T(t) \le x| \text{ Timeout is in progress at time } t) = 1, \quad 0 \le x \le K.$$

Also,

$$\lim_{t\to\infty} P(W^T(t) \le x| \text{ Timeout is not in progress at time } t) = F(x), \quad 0 \le x \le K,$$

where $F(x)$ is from Theorem 5.1.

It is clear that v_0^T is the fraction of the time the W^T process is in timeout period. Hence, conditioning on whether the timeout is in progress or not, we get

$$F^T(x) = v_0^T + (1 - v_0^T)F(x).$$

Equations 6.14 and 6.15 immediately follow from this. We thus need to derive Equation 6.13 to complete the proof. Note that the rate at which the W^T process leaves state 0 in steady state is $v_0^T \mu$. The rate at which it enters state 0 is given by

$$\int_0^K y dF^T(y) pc = (1 - v_0^T) \int_0^K dF(y) pc = (1 - v_0^T) mpc.$$

Equating these two we get Equation 6.13. This completes the proof.

Figure 2 shows the typical shape of the density function u^T. It also shows the two probability masses v_0^T and v_K^T. The fact that $u^T(0) = 0$ is the result of our approximation of the exponential growth of window size in the slow-start phase by an instantaneous jump. In the next section we validate the analytical results of this section by simulation using NS. As expected, we find that the empirical pdf of the window size near zero is indeed not zero. An extension of the analytical technique to better account for the slow-start growth is complicated, but is under consideration.

7   Validation by Simulation.

To evaluate the effectiveness of this model at representing TCP behavior, we compared the results from the analytic model with the results from simulations using the NS simulator [12]. This simulator is widely used in networking research and has a very detailed model of TCP error recovery and congestion control mechanisms. It has been subjected to careful validations and is regarded as a standard tool for TCP research. The simulated environment we chose for evaluating the model emphasizes the role of link- level congestion as the dominant factor that causes segment losses. For our experiments we constructed a model of an enterprise or campus

network having a single wide-area link to an upstream Internet service provider (ISP). The high level view of the simulations is that a single, continuous TCP flow sending at the maximum rate it can achieve shares a bottleneck link with a large number of TCP flows used for Web-like traffic. The motivation for choosing web-like traffic to share the link was the assumption that this traffic would exhibit highly variable and bursty demands on the link (specifically, a self-similar segment arrival process). The empirical data used to generate our web traffic [16] had heavy-tailed distributions for both "think" (OFF) times and response size (ON) times, which would lead to self-similar traffic from the aggregation of sources [46]. This segment arrival process entering the fixed capacity queue in the router serving the congested link provides a realistic simulation of the loss processes found in the Internet when router buffers overflow. Thus the simulation does not assume a particular loss model (independent, correlated, etc.) but instead the loss process is a direct result of running the simulation with different levels of load from self-similar web traffic.

We study the dynamics of the single continuously-sending TCP flow when it shares this congested link with many other TCP flows carrying web-like traffic. The simulated web clients (browsers) are all located on the enterprise or campus network and all their requests are satisfied by Web servers located somewhere on the Internet beyond the ISP link. The traffic model that drives the simulation is based on a contemporary model of web browsing reported in [16]. This model is an application-level description of the critical elements that characterize how HTTP/1.0 protocol is used. It is based on empirical data and is intended for use in generating synthetic Web workloads. Because response sizes are much larger than requests in this traffic model, the load on the network link that carries traffic from the servers to the browsers will be much greater than on the link carrying traffic in the opposite direction that is basically uncongested. The continuously-sending TCP flow also sends its segments along the congested path carrying web traffic from servers to clients.

The load generated by the web-like flows at the bottleneck link was varied in different runs of the simulation from 40% to 98% of a link capacity that was fixed at 10,000,000 bits per second. The continuous TCP flow attempted to send at the maximum rate it could sustain given the competing load from the web traffic on the shared link. For example, at a 40% load for web traffic, the continuous flow was able to send at a rate equivalent to nearly 60% of the link capacity. Note that because the continuous TCP flow can use all link bandwidth the web traffic is not using, the link is congested in all cases. As the load from the web traffic is increased, the continuously-sending TCP flow will experience greater segment loss and lower throughput.

The version of TCP simulated is TCP Reno and the delayed-acknowledgments option was turned off. The continuous TCP flow had a maximum receiver's window size of 45 segments of 1460 bytes. The buffer size in the router was 60 of these maximum size segments.

Each simulation was run for 60 minutes of simulated time and data from the first 10 minutes was discarded to eliminate transient start-up effects. Scripts for running the NS simulations

Load	40%	50%	60%	70%	80%	90%	98%
p	0.0007	0.0010	0.0026	0.0042	0.0090	0.0135	0.0195
c	0.3675	0.3316	0.3023	0.2760	0.4338	0.5332	0.6350
RTT	74.7	76.1	77.8	79.5	81.0	0.0843	88.2

Table 1: Estimates of p, c, and RTT for the seven runs.

which also show various parameter settings can be found at [50]. In each simulation run we recorded for the continuous TCP flow the number of segments transmitted, the number of 3-duplicate-acknowledgment loss events, the number of timeout loss events, the mean round-trip time, the mean throughput, and the effective size of the TCP window sampled every 10 milliseconds. The frequencies of loss events experienced by the continuous TCP flow in the simulations were used as loss probabilities in the analytic TCP model for a comparison of results. The mean round-trip time (including transmission, propagation and queueing delays) for the continuous TCP flow in the simulations was used as the mean RTT in the analytic model. The estimates of p, c and RTT (in milliseconds) are given in the table below for the seven different load conditions. They are also shown in Figures 5 and 6. Figure 5 shows the values of $p1 = p(1 - c)$ and $p2 = pc$, while Figure 6 shows the plot of the mean RTT (in milliseconds) for the seven runs.

In the NS simulation of TCP, the effective size of the TCP window is not reset to its minimum size (1 segment) until the end of a timeout period while the analytic model considers the effective window to be minimum (0 segments) for a random interval during the timeout period. The TCP window sizes sampled at 10 millisecond intervals during the simulations were adjusted to match this definition by setting the sample value to zero if the sample was taken during a timeout period. If the sample was not taken during a timeout period, the sample value was recorded without change.

8 Comparison of Analytic and Simulation Results

Figure 4 compares the throughput for the continuous TCP flow observed in the simulation with the throughput predicted by the analytic model (the line labeled "theory" in the plot) for several link loads generated by background web traffic. The loss probabilities used in the analytic model for each load were those values observed in the corresponding simulation as given in Figure 5. For example, at a load of 70% web traffic, the simulation results for the continuous TCP flow showed a per-packet probability of a loss indicated by 3 duplicate
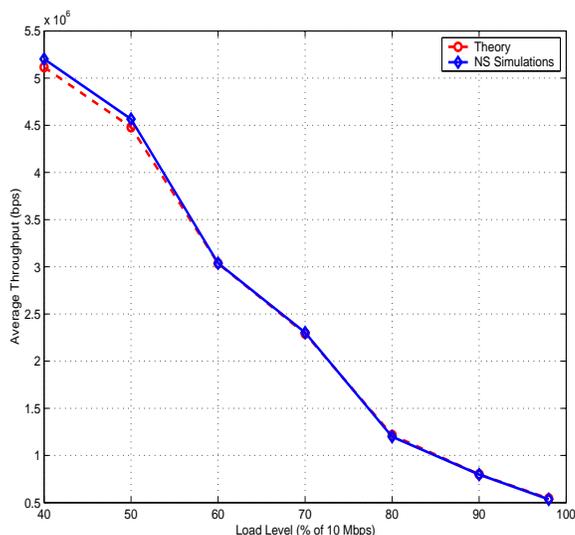
Figure 4: Throughput Comparison

acknowledgments of 0.003 and per-packet probability of a loss indicated by a timeout of 0.0012. The mean RTT used in the analytic model for each load were those values observed in the corresponding simulation as given in Figure 6. For example, at a 70% load the mean RTT for the continuous TCP flow was 79 milliseconds.

The analytic model predicts slightly lower throughput only at the lowest loads. At all higher loads where the TCP connection experienced more loss and queueing delays, the simulation and analytic models produce essentially the same results. Thus for a single TCP connection, the model presented here can be expected to give an excellent prediction of throughput for given probabilities of fast recovery and timeout events and a mean RTT.

Figures 7, 8, and 9 compare the probability density function of effective TCP window size computed with the analytic model ("theory") with histograms of window size observed in the NS simulations for different loads of background web traffic. The plots on the left of each figure show a full range of probability density values on the y axis while the plots on the right show more detail for densities in the range [0, 0.2]. All related plots are shown with a uniform scale for ease of comparison. For the analytic results, the computed probability mass at window sizes of zero and K are plotted with special symbols.
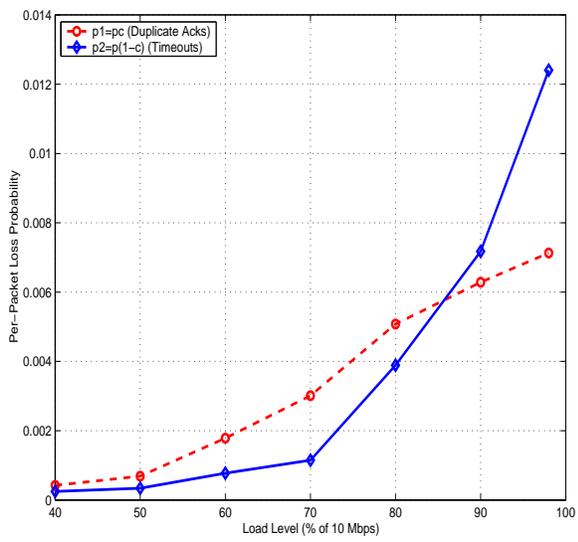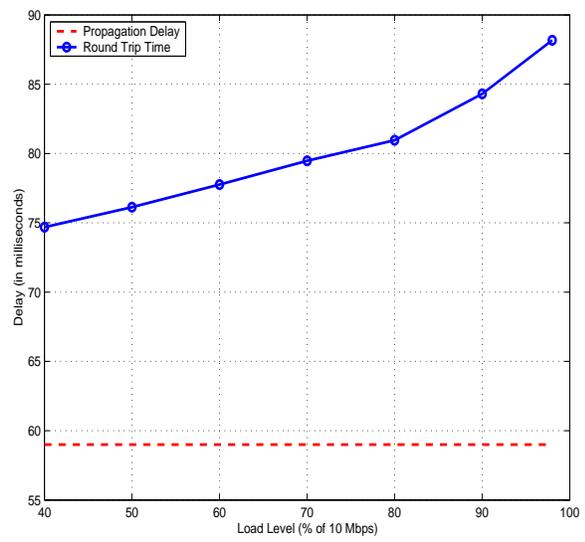
Figure 5: Loss Rates in NS Simulations

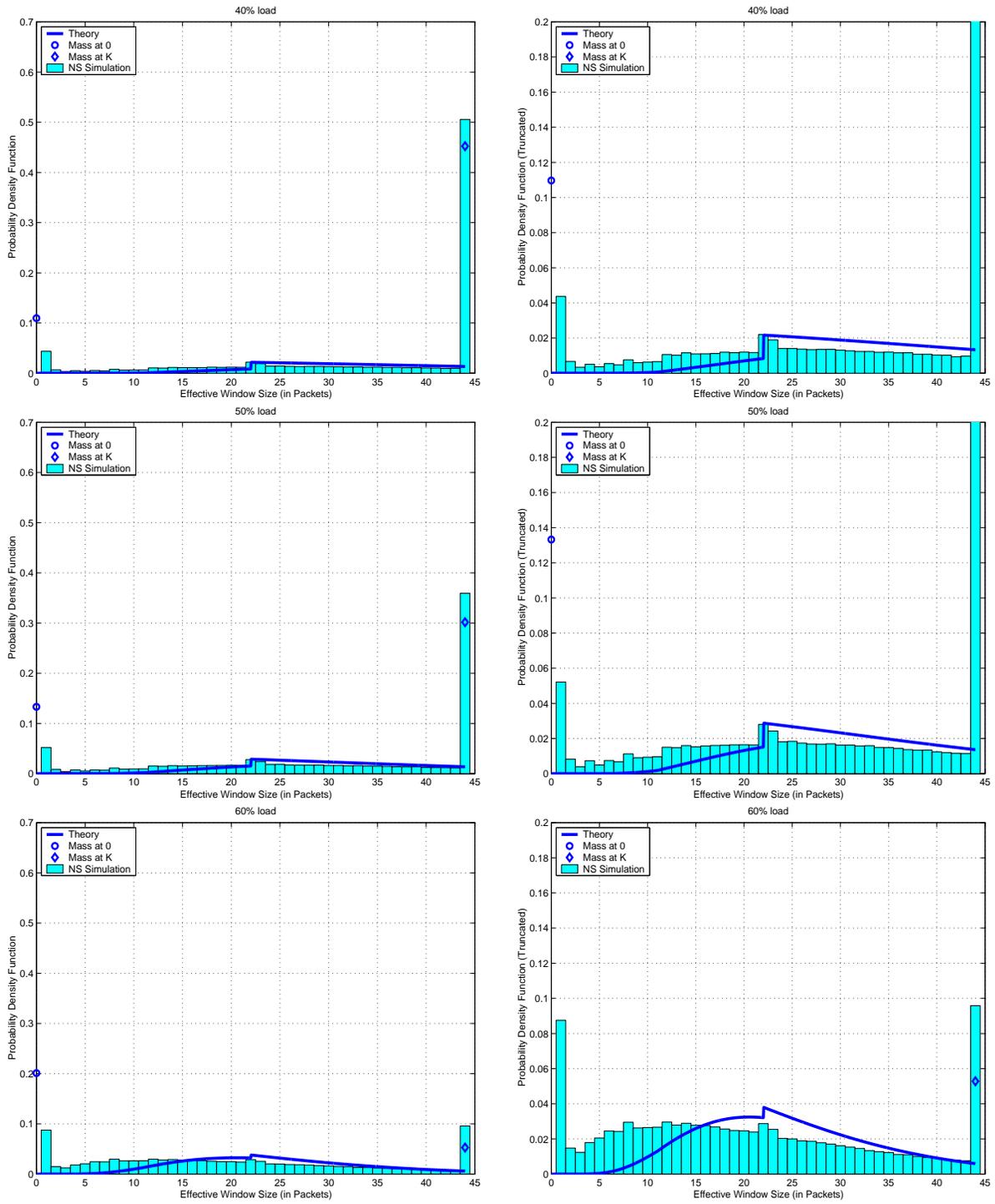Figure 6: Mean RTTs in NS Simulations

23

Figure 7: Comparison of Effective Window Size Distributions at 40% , 50% and 60% loads
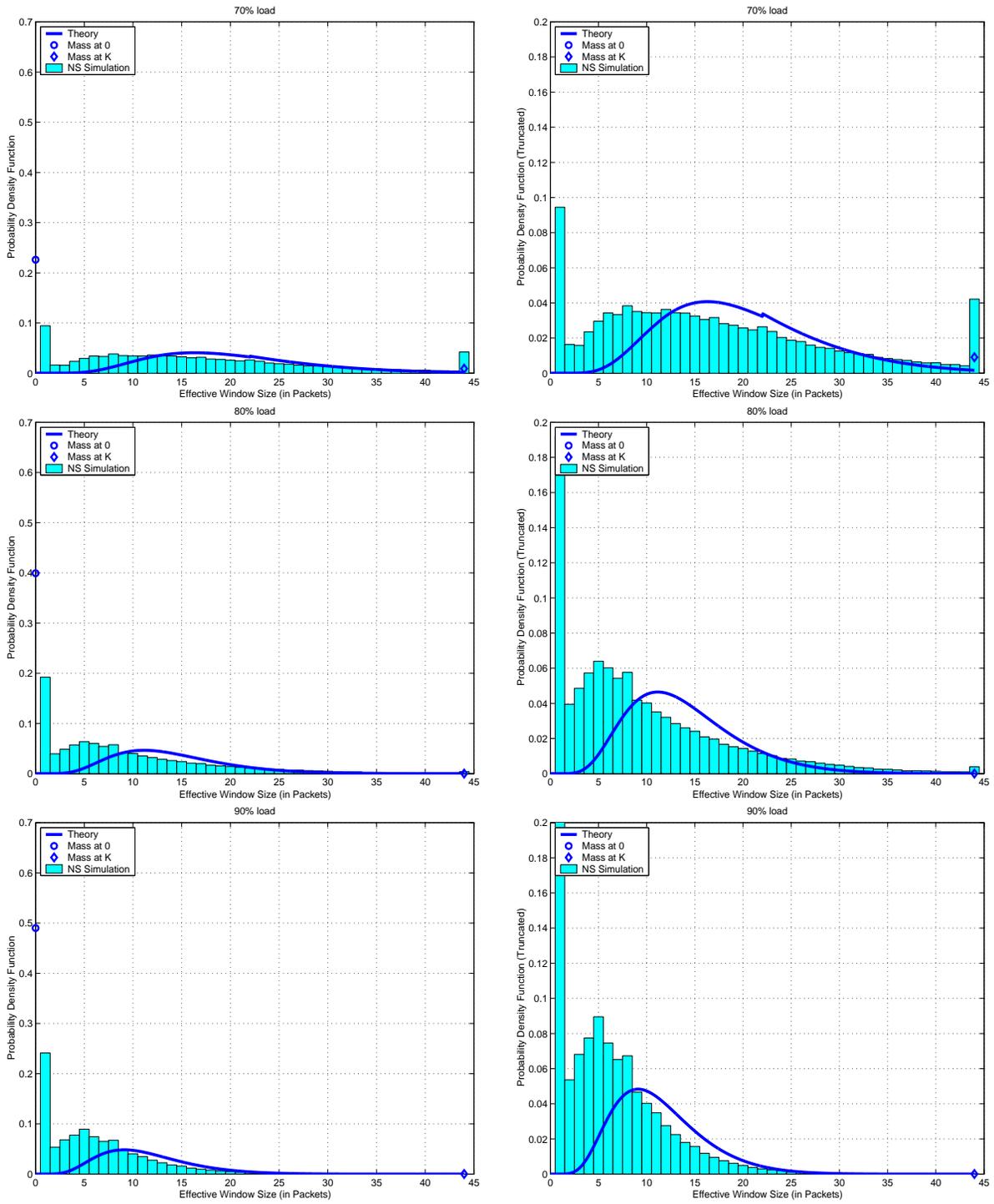
Figure 8: Comparison of Effective Window Size Distributions at 70% , 80% and 90% loads
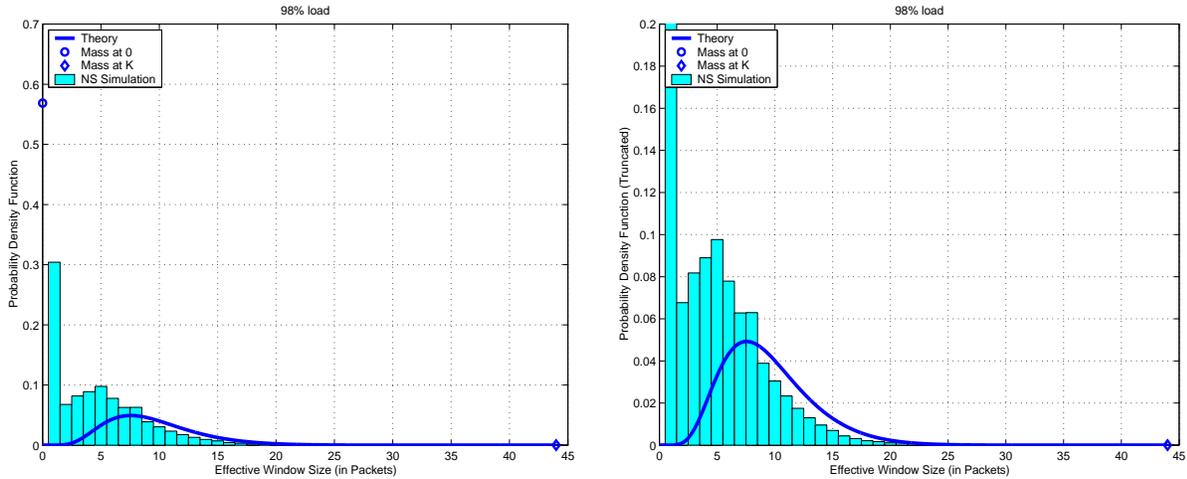
Figure 9: Comparison of Effective Window Size Distributions at 98% loads

In general, the analytic model matches well with the distribution of effective window sizes from the simulation and also shows mass peaks at 0, K, and, at lower loads, K/2 (resulting from halving the maximum window size). Note that the analytic model treats the minimum effective window size as 0 while the simulated TCP correctly uses a minimum window size of 1. The theoretical pdf of window sizes differs from the simulation results mainly for small window sizes. A significant factor contributing to this, especially at the higher loads, is that the model approximates slow-start after a timeout event by an instantaneous jump in the window size to the initial window for congestion avoidance and enters that phase. In the simulation, the window is advanced from 1 segment through some number of small values before reaching the congestion-avoidance phase.

9   Summary and Conclusions

We have described a new analytic model for the dynamic behavior of TCP windows as they respond to congestion indicated by data loss in the network. This model is based on stochastic differential equations and allows us to compute throughput as well as the complete probability density function for effective window sizes. The results computed from this model have been compared with NS simulations of TCP behavior in a realistic network environment and were found to produce comparable results.

26

References

[1] A. Abouzeid, S. Roy, and M. Azizoglu, Stochastic Modeling of TCP over Lossy Links, Proceedings of INFOCOM 2000.

[2] A. Abouzeid, S. Roy, and M. Azizoglu, Stochastic Modeling of TCP over Random Loss Channels, Proceedings of the 6th International Conference on High Performance Computing, 1999

[3] E. Altman, K. Avrachenkov and C. Barakat A Stochastic Model of TCP/IP with Stationary Random Loss, Proceedings of SIGCOMM 2000.

[4] E. Altman, K. Avrachenkov and C. Barakat, TCP in presence of bursty losses, Proceedings of SIGMETRICS 2000.

[5] E. Altman, K. Avrachenkov and C. Barakat, Impact of Bursty Losses on TCP Performance, Performance Evaluation, Vol. 42, no. 2-3, pp. 129-147, 2000.

[6] E. Altman, K. Avrachenkov, C. Barakat, TCP Network Calculus: The case of large delay-bandwidth product, Proceedings of INFOCOM 2002.

[7] E. Altman, T. Jimenez, R. Nunez-Queija, Analysis of two competing TCP/IP connections, Performance Evaluation, Vol 49, no. 1-4, September 2002, pp. 43-55.

[8] F. Anjum and L. Tassiulas, On the Behavior of Different TCP Algorithms over a Wireless Channel with Correlated Packet Losses, Proceedings of SIGMETRICS 1999.

[9] F. Baccelli and D. Hong, TCP is Max-Plus Linear, Proceedings of SIGCOMM 2000.

[10] F. Baccelli and D. Hong, AIMD, Fairness and Fractal Scaling of TCP Traffic, Proceedings of INFOCOM 2002.

[11] C. Barakat, TCP modeling and validation, IEEE Network, Vol. 15, no. 3, May 2001, pp. 38-47.

[12] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varad-han, Y. Xu, H. Yu, Advances in Network Simulation, IEEE Computer, vol. 33 no. 5, May 2000, pp. 59-67.

[13] N. Cardwell, S. Savage and T. Anderson. Modeling TCP Latency, Proceedings of INFOCOM 2000.

[14] V. Dumas, F. Guillemin and P. Robert. Limit results for Markovian models of TCP, Proceedings of GLOBECOM 2001.

[15] V. Dumas, F. Guillemin and P. Robert, A Markovian Analysis of Additive- Increase Multiplicative-Decrease (AIMD) Algorithms, Advances in Applied Probability, Vol. 34, no. 1, 2002, pp. 85-111.

[16] A. Feldmann, A. Gilbert, P. Huang, and W. Willinger, Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control, Proceedings of SIGCOMM 1999.

[17] 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, pp. 30-47.

[18] S. Floyd and V. Jacobson, On Traffic Phase Effects in Packet-Switched Gateways, Internetworking: Research and Experience, V.3 N.3, September 1992, pp. 115-156.

[19] S. Floyd, and K. Fall, Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, Vol. 7, no. 4, August 1999, pp. 458-472.

[20] S. Floyd, M. Handley, J. Padhye, and J. Widmer, Equation-Based Congestion Control for Unicast Applications, Proceedings of SIGCOMM 2000.

[21] M. Garetto, R. Cigno, M. Meo, and M. Marsan. A Detailed and Accurate Closed Queueing Network Model of Many Interacting TCP Flows, Proceedings of INFOCOM 2001.

[22] M. Goyal, R. Guérin, and R. Rajan, Predicting TCP Throughput from Non- invasive Network Sampling, Proceedings of INFOCOM 2002.

[23] C. Hollot, V. Misra, D. Towsley and W. Gong. On Designing Improved Controllers for AQM Routers Supporting TCP Flows, Proceedings of INFOCOM 2001.

[24] C. Hollot, V. Misra, D. Towsley and W. Gong. A Control Theoretic Analysis of RED, Proceedings of INFOCOM 2001.

[25] F. Kelly, A. Maulloo, and D. Tan. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability, JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY vol. 49, no. 3, March 1998, pp. 237-252.

[26] A. Kumar. Comparative Performance Analysis of Versions of TCP in local network with a lossy link, IEEE/ACM Transactions on Networking, Vol. 6, no. 4, August 1998, pp. 485-498.

[27] T. V. Lakshman and U. Madhow, The Performance of Networks with High Bandwidth-delay Products and Random Loss, IEEE/ACM Transactions on Networking, Vol. 5, no. 3, June 1997, pp. 336-350.

[28] T. V. Lakshman, U. Madhow, and B. Suter, Window-based Error Recovery and Flow Control with a Slow Acknowledgment Channel: a Study of TCP/IP Performance, Proceedings of INFOCOM 1997.

[29] T. V. Lakshman, U. Madhow, and B. Suter, TCP/IP Performance with Random Loss and Bidirectional Congestion, IEEE/ACM Transactions on Networking, Vol. 8, no. 5, October 2000, pp.541-555.

[30] S. H. Low, A Duality Model of TCP and Queue Management Algorithms, Extended version of paper that appears in Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, 2000.

[31] S. Low, L. Peterson and L. Wang, Understanding TCP Vegas: A duality model Proceedings of SIGMETRICS 2001

[32] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm, Computer Communication Review, volume 27, number3, July 1997.

[33] A. Misra and T. Ott, The Window Distribution of Idealized TCP Congestion Avoidance with Variable Packet Loss, Proceedings of INFOCOM 1999..

[34] V. Misra, W. Gong and D. Towsley, Stochastic Differential Equation Modeling and Analysis of TCP Window size Behavior, Proceedings of Performance1999.

[35] V. Misra, W. Gong and D. Towsley. A Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, Proceedings of SIGCOMM 2000.

[36] M. Mitzenmacher and R. Rajaraman. Towards More Complete Models of TCP Latency and Throughput, Journal of Supercomputing, 2001.

[37] T. Ott, J.H.B. Kemperman, and M. Mathis, The Stationary Behavior of Ideal TCP Congestion Avoidance, 1996. (unpublished manuscript)

[38] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, Modeling TCP Reno Performance: a Simple Model and its Empirical validation, IEEE/ACM Transactions on Networking, Vol. 8, no. 2, April 2000, pp. 133-145.

[39] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, Modeling TCP Throughput: a Simple Model and its Empirical Validation, Proceedings of SIGCOMM 1998.

[40] M. Roughan, A. Erramilli and D. Veitch Network performance for TCP networks. Part 1: Persistent sources. , In the proceedings of Seventeenth International Teletraffic Congress, Salvador da Bahia, Brazil, September 24-28, 2001.

[41] S. Sahu, P. Nain, D. Towsley, C. Diot and V. Firoiu, On Achievable Service Differentiation with Token Bucket Marking for TCP, Proceedings of SIGMETRICS 2000.

[42] S. Savari and E. Telatar, The Behavior of Stochastic Processes Arising in Window Protocols, Proceedings of the 1999 IEEE International Symposium on Information Theory.

[43] H. Schwefel. Behavior of TCP-like Elastic Traffic at a Buffered Bottleneck Router, Proceedings of INFOCOM 2001.

[44] B. Sikdar, S. Kalyanaraman and K. S. Vastola. An Integrated Model for the Latency and Steady-State Throughput of TCP Connections. Performance Evaluation, Vol. 46, no. 2-3, September 2001, pp. 139-154.

[45] B. Sikdar, S. Kalyanaraman and K. S. Vastola, Analytic models and comparative study of the latency and steady-state throughput of TCP Tahoe, Reno and SACK . Proceedings of GLOBECOM 2001.

[46] W. Willinger, M. Taqqu, R. Sherman, and D.Wilson, Self-Similarity Through High-Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level, IEEE/ACM Transactions on Networking, Vol. 5, No. 1, February 1997, pp. 71-86.

[47] Y. Yang and S. Lam, General AIMD Congestion Control, Proceedings of ICNP 2000.

[48] Y. Yang, M. Kim, and S. Lam. Transient Behaviors of TCP-friendly Congestion Control Protocols, Proceedings of INFOCOM 2001.

[49] M. Zorzi, A. Chockalingam, and R. Rao, Performance Analysis of TCP on Channels with Memory, IEEE-JSAC Vol. 18, 2000, pp. 1289-1300.

[50] URL: http://www.cs.unc.edu/Research/dirt/proj/tcpmodel/