

# Stochastic Modeling of TCP over Lossy Links

Alhussein A. Abouzeid, Sumit Roy, Murat Azizoglu

Department of Electrical Engineering  
University of Washington, Box 352500  
Seattle, WA 98195-2500, USA

E-mail: {hussein,roy,azizoglu}@maxwell.ee.washington.edu

**Abstract**—An analytical framework for modeling the performance of a single TCP session in the presence of random packet loss is presented. A Markovian approach is developed that allows us to study both memoryless channels (i.i.d packet loss) and channels with memory (correlated packet loss) modeled by a two state continuous time Gilbert model. The analytical results are validated against results using the *ns* simulator. It is shown that the model predicts throughput for LANs/WANs (low and high bandwidth-delay products) with good accuracy. Further, throughput for the i.i.d loss model is found to be relatively insensitive to the probability density function (p.d.f) of the loss inter-arrival process. For channels with memory, we present an empirically validated rule of thumb to categorize the channel transition frequency.

**Keywords**—Transport control protocol, wireless networks, performance analysis.

## I. INTRODUCTION

MOST of today's Internet traffic is carried by networks using TCP. While initial research efforts relied mainly on direct measurements and TCP protocol simulations to predict key TCP characteristics, several recent efforts have been directed towards developing an analytical model for the dynamic behavior of TCP. This paper presents a further contribution towards analytical modeling of TCP's congestion avoidance mechanism that extends the results in the literature in several important ways. Most notably, we concentrate on TCP behavior over *lossy* channels - with (i) random (i.i.d) packet loss and (ii) with memory (i.e. bursty) losses.

Our model presented here for lossy links uses most of the notation and results of the ideal channel window dynamics analyzed in [1].

The literature on TCP analysis for lossy links can be categorized based on assumption of a) independent packet loss, and b) correlated packet loss (bursty errors), respectively. The work in [1], [2], [3] falls in the first category. In the pioneering paper of [1], a complete analytical description of TCP congestion window evolution over *ideal* (non lossy) channels is derived, which provides the point of departure (and much of the notation) for our work. In these works, a packet loss event is assumed with probability  $q$  independent of all other packets loss events, inducing a geometric distribution on the number of successfully transmitted packets between consecutive packet losses, which may not be appropriate for specific lossy links. Our approach for modeling independent packet losses is fundamentally different. Instead of assuming a specific packet loss distribution at

the link layer, we derive the packet loss statistics from an underlying continuous time model of the physical link. Hence, our model for the independent packet loss can be also applied to situations where the distribution of the number of successful packet transmissions between consecutive random packet losses is not geometric. In addition to this, our work has the advantage of applicability to a wider range of network parameters, since [1] considers links with high bandwidth-delay products (WANs) only, [3] considers links with very small (zero) propagation delay (LANs) only, and [2] only considers links with low bandwidth-delay products (LANs).

The work of [4], [5], [6], [7] falls in the second category, where the authors study TCP behavior over links experiencing bursty packet losses. In [4], the author uses an approach similar to that in [1], [2], [3] except that, conditioned on a packet loss (which takes place independently with probability  $q$ ), all subsequent packets in the same congestion window are assumed to be lost. While a drop-tail policy as used by IP routers may justify this, the assumptions on the underlying physical channel that would result in such a packet-level loss behavior was not defined. Further, [4] approximates the window size evolution to be always linear in the congestion avoidance phase, although it is known that the evolution is sub-linear for higher window sizes. While this approximation may result in negligible deviation in the predicted throughput for links with small bandwidth-delay product and/or high packet loss rate (which has the effect of limiting the congestion window size), the approximation effectively increases the channel capacity for higher window sizes, resulting in an optimistic predicted throughput for links with large bandwidth-delay products and/or low loss probability.

Our model assumptions are closest to [5], [6], [7] in that we assume the channel can be modeled by a *continuous-time* two-state Markov Chain alternating between a good and a bad state, where packets are lost in the good state w.p. (with probability) 0 and in the bad state w.p. 1. An embedded Markov Chain for the congestion window at the instants of random loss is identified, and by analyzing the interaction between the deterministic window evolution for the ideal channel case and the stochastic error process, the steady state probability distribution of the window size is computed. Hence using renewal theory, the (long-term) average throughput is computed. We remark that the discrete-time (in units of a packet transmission duration or slot) models in [5], [6], [7] effectively assume that the channel state transi-

This work was supported in part by AFOSR Grant F49620-1-0472

tions occur synchronously at the packet boundaries. This is a) inconsistent with the physical channel state evolution which is not predicated on the timing of packet transmissions and more importantly, b) does not allow for modeling channel variations faster than a packet transmission interval (i.e. ‘fast’ fading). In addition, our model is applicable to a wider range of network parameters - specifically, [5] assumes a buffer with infinite capacity (no packet loss due to buffer overflow), [6] considers links with low bandwidth-delay products ( $\text{RTT} > \text{Buffer size}$ , both being expressed in units of packets) and [7] considers links with very small delay (instantaneous acknowledgments) and is limited to slow fading channels. Our model thus provides a unified approach to model all these situations. Finally, we use results using *ns* simulator to validate model accuracy.

The remainder of this paper is organized as follows. In Section II, we describe our proposed analytical model and summarize previous research relevant to our work. In Section III, we present the channel loss models and deduce a set of analytical expressions characterizing the steady state throughput of TCP-Reno (TCP-R) as a function of the channel loss statistics and the link parameters. Section IV is devoted to model validation, where we describe the channel loss models developed for the *ns* simulator and its use towards validating our model accuracy. It contains a catalog of representative simulation results that highlights the wide applicability of the model. The work and its possible extensions are summarized in the concluding remarks of Section V.

## II. A MODEL FOR TCP CONGESTION AVOIDANCE

The window-based congestion avoidance mechanism in TCP/IP [8] acts as a self-clocking regulator based on receiver feedback (or lack thereof). In TCP/IP, when a node successfully receives a packet, it sends an acknowledgment (ACK) back to the source. At all times, the source keeps a current record of the number of unacknowledged packets that it has released into the network - called the congestion *window size*, as well as an estimate of the round-trip time (RTT). The source is allowed to increase its window size as long as packets are being acknowledged. The source detects a packet loss by either the non-arrival of a packet ACK within a certain time (i.e. via timer expiry or time out), or by the arrival of multiple ACKs with the same next expected packet number (typically 3 duplicate ACKs). In this paper, these two modes of packet loss detection are denoted by ‘TO’ and ‘TD’ respectively. A packet loss is interpreted by the source as an indication of congestion, and the source responds by reducing its window size, thereby indirectly controlling the data rate. Modeling the dynamic behavior of congestion window size is thus key to analyzing TCP throughput performance in a variety of situations.

Every time TCP receives an ACK, it updates its estimate of the RTT. We assume that the RTT estimate denoted by  $\Delta$ , is constant. Hence, in normal operation, the timer is set to  $\Delta$  each time a packet is transmitted. However, when multiple TO

events take place consecutively (i.e. without the reception of any ACKs in between), TCP applies the Binary Exponential Back-off (BEBO) algorithm, where, for the  $n^{\text{th}}$  consecutive TO event ( $n$  is an integer  $> 0$ ), the packet is retransmitted and the timer is set to  $\min\{2^n \Delta, 64 \Delta\}$ .

Our basic system model assumes an infinite source (i.e. one that always has a packet to send) that releases packets into a buffer of size  $B$  packets upon receiving ACKs from the destination. The packets are then sent over a single link with capacity  $\mu$  packets per second and a net fixed delay of  $\tau$  (propagation delay through the channel, any other processing delays etc.) is assumed. Define  $T = \tau + 1/\mu$  to be the time between the start of transmission of a packet and the reception of an ACK for this packet, excluding any queuing delays in the buffer. Then  $\mu T$  is the bandwidth-delay product and the ratio  $\beta = \frac{B}{\mu T}$  is the buffer size normalized by the bandwidth-delay product. Let  $w_p$  denote the maximum number of packets that could be in transit between the source and destination (including the packets in the link buffer). Thus  $w_p = B + \mu T = \mu T (\beta + 1)$ .

The above model has been analysed by [1] for ideal channel operation where the only packet loss is due to buffer overflow. The window size evolution was found to be periodic. We summarize their analytical results relevant to our work next. Readers interested in the details of the derivations are encouraged to refer to [1]. Let  $t' = 0$  denote the time of establishment of the TCP session under consideration and  $W(t')$  denote the congestion window size at time  $t'$ . Let  $n$  denote the number of packets acknowledged during a time interval  $t$  and consider two instants  $t'_0, t'_0 + t$  during the congestion avoidance phase of any TCP cycle (see Figure 1). Choose  $t'_0$  such that  $W(t'_0) = W_0$ , and choose  $W_0 = 1$  in the slow start phase. Then,

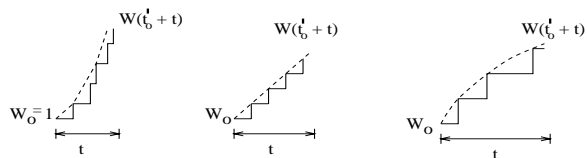


Fig. 1. Sketch of the exponential, linear and sub-linear  $O(\sqrt{t})$  phases for window evolution. Solid lines indicate the actual window size evolution while dotted lines indicate the envelope.

**Slow Start** ( $1 < W(t') < W_{th}$ )

$$W(t'_0 + t) = 2^{t/T} \quad (1)$$

$$n = W(t'_0 + t) - 1 \quad (2)$$

**Congestion Avoidance - Phase I** ( $W_{th} < W(t') < \mu T$ )

$$W(t'_0 + t) = W_0 + t/T \quad (3)$$

$$n = \frac{1}{T}(W_0 t + t^2/(2T)) \quad (4)$$

**Congestion Avoidance - Phase II** ( $\mu T < W(t') < w_p$ )

$$W(t'_0 + t) = \sqrt{W_0^2 + 2\mu t} \quad (5)$$

$$n = \mu t \quad (6)$$

Note that in the sequel, we focus on time instants  $t'$  where  $W(t')$  is discrete - the details of the original derivation are in [1]. The window evolution of TCP-R over ideal channels is found to be periodic. Only the first cycle of TCP-R starts with slow start, and after that the session continues in the congestion avoidance phase, where the window size increases until a buffer overflow takes place each time the window size reaches  $w_p$ , and hence the average throughput  $\rho$  (average packet transmission rate normalized by the link capacity) is given by

$$\beta < 1 : \rho = \left( \frac{n_A + n_B}{t_A + t_B} \right) \left( \frac{1}{\mu} \right) \quad (7)$$

$$\beta > 1 : \rho \simeq 1 \quad (8)$$

where  $n_A(n_B)$  is the number of successful packet transmission in the Congestion Avoidance Phase I (Phase II), with  $t_A(t_B)$  denoting the time duration of each phase, respectively. Define  $t_p$  to be the time duration of a single cycle of the window evolution (from  $w_p/2$  till  $w_p$ ). Then,

$$\beta < 1 : t_p = t_A + t_B \quad (9)$$

$$\beta > 1 : t_p = t_B \quad (10)$$

In our subsequent analysis of random packet loss, we refer to  $t_p$  as the typical cycle.

### III. LOSSY CHANNELS MODEL

The previous discussion assumed that the only source of packet loss is buffer overflow. In this section, we assume the channel causes additional random packet loss, and hence packet loss will be a mixture of buffer overflows and random loss.

We assume the channel can be modeled by a continuous time two-state alternating process,  $\{S(t)|t \geq 0\}$  taking values described by  $k = 0$  (Good) or 1 (Bad) with properties that:

1. When the channel leaves one state, it will enter the other state with probability 1 (i.e. an alternating process).
2. The durations of time in the good state,  $\{X_i, i = 1, 2, \dots\}$  are independent and identically distributed (i.i.d) with known cumulative distribution function  $F(x)$  (respectively, the probability density function  $f(x)$ ) with mean  $E[X_i] = \frac{1}{\lambda_0}$ .
3. The durations of time in the bad state, denoted by  $\{Y_i, i = 1, 2, \dots\}$  are i.i.d with mean  $E[Y_i] = \frac{1}{\lambda_1}$ , and independent of the  $\{X_i\}$ .
4. In each of the states, the packet loss mechanism is that of a discrete memoryless (packet) channel, with respective loss probabilities  $p_0$  and  $p_1$ . For simplicity, in this work we assume that  $p_0 = 0, p_1 = 1$ , i.e. there are no packet losses in the good state and it occurs with probability one during the bad state.

In subsection A, we will consider the special case of IID loss by assuming that  $Y_i$  have a deterministic distribution  $Y_i = \epsilon w_p$ . 1 where  $\epsilon \simeq 0$ . Subsection B considers the correlated loss case.

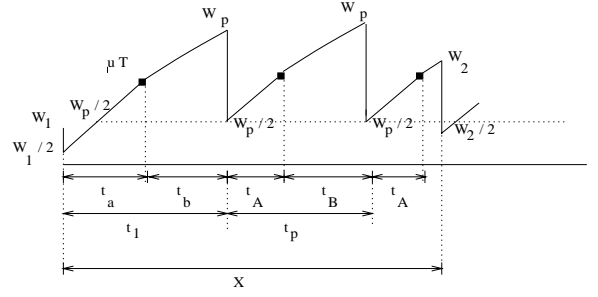


Fig. 2. A sketch of a sample function of window size in the presence of random loss for  $\beta < 1$  ( $\mu T > w_p/2 > W_1/2$ )

#### A. Lossy Memoryless Channels - IID Loss

Let  $L_i$  denote the time of the  $i^{\text{th}}$  random packet loss and  $W_i$  denote the window size just before a random packet loss takes place. Then  $X_i = L_i - L_{i-1}$  denotes the time between the  $(i-1)^{\text{th}}$  and  $i^{\text{th}}$  random packet losses with  $X_1 = L_1$  by convention. Since  $X_i$  are IID random variables, the process defined by the loss occurrence times  $\{L_1, L_2, \dots\}$  is a renewal process with inter renewal p.d.f.  $f(x)$ .

The window size  $W(t')$  is a semi-Markovian stochastic process, because the window size evolution after a random loss (except for its starting value which is half of that just before the random loss) is statistically independent from the window size evolution before the random loss. Further, since  $\{X_1, X_2, \dots\}$  are i.i.d, the window sizes  $\{W_1, W_2, \dots\}$  at instants just prior to a random loss form a finite state Markov Chain.

With the above model of random loss, two quantities associated with the just defined Markov Chain are of interest;

1.  $E[N|W_1 = w_1]$ , the expected number of packets successfully transmitted before another random packet loss occurs, given that the most recent random loss took place at a window size  $w_1$ .
2. The conditional probability  $P[W_2 = w_2|W_1 = w_1]$  (denoted  $P_{w_1, w_2}$ ); the probability that the next random loss takes place at  $W_2 = w_2$  given that the previous random loss took place at  $W_1 = w_1$ .

Since packet losses in this section are single isolated events, a random packet loss will be typically detected by a TD. Hence, all packet losses, whether random or due to buffer overflow, will be detected via TD events for the memoryless channels case. We will neglect the delay between a packet loss event and its detection (this delay is small, and is typically bounded by  $\frac{3}{\mu}$  for the IID loss case).

To evaluate the above two quantities of interest, an approximation for TCP-R similar to that previously used by [1] in deriving (7) and (8) for channels without random loss is invoked. We ignore the first cycle of TCP-R and assume that the TCP-R session starts with window size  $w_p/2$  instead of starting with a window size of 1. This approximation should have a negligible effect on the average throughput, due to two reasons. First, a source with an infinite number of packets was assumed; hence

the effect of the transient behavior (slow start) at the beginning of the connection is expected to be negligible, even for the case of random loss; Second, the duration as well as the number of packets successfully sent during this slow start phase is low (recall that slow start is actually ‘fast’). In other words, this transient behavior disappears very fast.

In the analysis that follows, the first TCP cycle just after a random packet loss (from  $w_1/2$  till  $w_p$ ) is called the ‘atypical’ cycle. Between any two consecutive random packet losses, TCP-R starts by entering the atypical cycle, and depending on the duration of the good state, one or more typical cycles follow until the subsequent random packet loss takes place.

Two ranges of  $\beta$  ( $\beta < 1$  and  $\beta > 1$ ) are considered separately for computation of  $E[N|W_1]$  and  $P_{w_1, w_2}$ . The reason for this is that for  $\beta < 1$ ,  $w_1/2 \leq w_p/2 < \mu T$  and hence the atypical cycle will start by a linear growth of the window size. On the other hand, for  $\beta > 1$ ,  $w_p/2 > \mu T$  and hence, depending on the value of  $W_1$ , the atypical cycle will start by either a linear growth ( $w_1/2 < \mu T$ ) or an  $O(\sqrt{t})$  growth ( $w_1/2 > \mu T$ ).

Define,

$N_a, N_b$  : the number of successful packet transmissions during Congestion Avoidance Phase I and II, respectively, of the atypical cycle following a random packet loss at a window size  $W_1 = w_1$ .

$N_A, N_B$  : the number of successful packet transmissions during Congestion Avoidance Phase I and II, respectively, of a typical cycle.

$N_p = N_A + N_B$  is the number of packets sent in a typical cycle ( $N_A = 0$  for  $\beta > 1$ ).

The corresponding durations where the above number of packets is transmitted (time is counted since the beginning of the phase referenced) are  $t_a, t_b, t_A$ , and  $t_B$  respectively. Thus,  $t_1 = t_a + t_b, t_p = t_A + t_B$  are the durations of the atypical cycle and a typical cycle, respectively (Figure 2). For  $\beta < 1$ ,  $N_A = 0$  and  $t_A = 0$ .

The values of  $N$  and  $t$  with capitalized subscripts ( $A, B$ ) do not depend on  $W_1$  and can be computed using (3)-(6) as in the case of channels without random packet loss. On the other hand, the values with small letter subscripts ( $a, b$ ) refer to the atypical cycle of TCP-R just after a random loss at a window size  $W_1$ . Those values depend on the value of  $W_1$ , and can also be computed from (3)-(6) by substituting by the appropriate initial and final values of the window size.

A set of parameters that will simplify the calculation of  $E[N|W_1]$  will henceforth be defined. Conditioned on  $W_1 = w_1$ , let us assume that no more random packet losses will take place. For  $\beta < 1$ , let  $a(n)$  denote the time at which the  $n^{th}$  packet of the linear phase of the atypical cycle is transmitted. Similarly, let  $b_1(n)$  denote the  $n^{th}$  packet of the  $O(\sqrt{\cdot})$  phase. Let  $A(n)$  denote the time at which the  $n^{th}$  packet of the linear phase of the  $j + 1^{st}$  typical cycle is transmitted, and let  $B_1(n)$  denote the  $n^{th}$  packet of the  $O(\sqrt{\cdot})$  phase. Then, expressions for these parameters are computed from (4) and (6) by substituting

by the appropriate values of  $W_0$ , solving for  $t$ , and then adding the appropriate time shift value. For example,  $a(n)$  is calculated by solving for  $t$  in (4) and substituting  $W_0 = W_1/2$ . In this case, there is not time shift adjustment, since  $t$  is counted from the beginning of the phase in question, which is also the time at which random packet loss took place. Similarly,  $b_1(n)$  is calculated by solving for  $t$  in (6) and then adding the time at which this phase of the cycle starts ( $t_a$ ). The set of parameters  $a(n), b_1(n), A(n)$  and  $B_1(n)$  can be used for the case of  $\beta > 1$  only when  $w_1/2 < \mu T$ . For  $w_1/2 > \mu T$ , then, in a similar fashion, we define  $b_2(n)$  and  $B_2(n)$ . Hence,

$$\begin{aligned} a(n) &= T/2\sqrt{w_1^2 + 8(n+1)} - T/2w_1 \\ b_1(n) &= t_a + (n+1)/\mu \\ b_2(n) &= (n+1)/\mu \\ A(n) &= t_1 + jt_p + T/2\sqrt{w_p^2 + 8(n+1)} - T/2w_p \\ B_1(n) &= t_1 + t_A + jt_p + (n+1)/\mu \\ B_2(n) &= t_1 + jt_p + (n+1)/\mu \end{aligned} \quad (11)$$

where,  $0 \leq n < \infty$  is an incremental counter that is initialized at the beginning of each congestion avoidance phase (linear or  $\sqrt{\cdot}$ ) and  $0 \leq j < \infty$  is an incremental counter that refers to the  $j + 1^{st}$  typical cycle after the first atypical cycle following a random packet loss.

A similar set of parameters  $c(w_2), d_1(w_2), d_2(w_2), C(w_2), D_1(w_2)$  and  $D_2(w_2)$ , denoting the times at which the window size reaches  $w_2$  in the linear and  $\sqrt{\cdot}$  phases of the atypical and typical cycles, can be defined in a way that parallels the definition of the parameters in (11). Expressions for these parameters are deduced by substituting by the appropriate values of  $W_0$  in (3) or (5), solving for  $t$ , and then adding the appropriate time shift parameter. Hence,

$$\begin{aligned} c(w_2) &= T(w_2 - w_1/2) \\ d_1(w_2) &= t_a + \frac{w_2^2 - (\mu T)^2}{2\mu} \\ d_2(w_2) &= \frac{w_2^2 - (w_1/2)^2}{2\mu} \\ C(w_2) &= t_1 + jt_p + T(w_2 - w_p/2) \\ D_1(w_2) &= t_1 + jt_p + t_A + \frac{w_2^2 - (\mu T)^2}{2\mu} \\ D_2(w_2) &= t_1 + jt_p + \frac{w_2^2 - (w_p/2)^2}{2\mu} \end{aligned} \quad (12)$$

#### A.1 Channels with large bandwidth-delay product ( $\beta < 1$ )

Since

$$E[N|W_1 = w_1] = \sum_{n=0}^{\infty} Pr[N > n|W_1 = w_1], \quad (13)$$

using (11), the above can be written in terms of the complementary distribution function of  $X$ , i.e.,  $\bar{F}(a) = Pr[X > a]$  as follows.

$$E[N|W_1 = w_1] = \sum_{n=0}^{N_a-1} \bar{F}[a(n)] + \sum_{n=0}^{N_b-1} \bar{F}[b_1(n)] + \sum_{j=0}^{\infty} \sum_{n=0}^{N_A-1} \bar{F}[A(n)] + \sum_{j=0}^{\infty} \sum_{n=0}^{N_B-1} \bar{F}[B_1(n)] \quad (14)$$

In a similar fashion, using (12),  $P_{w_1, w_2}$  can be written in terms of  $\bar{F}(x)$ , where we introduce  $\bar{F}(a, b) = \bar{F}(a) - \bar{F}(b)$ .

$$P_{w_1, w_2} = \begin{cases} 0 & R_1 \\ \bar{F}(c(w_2), c(w_2 + 1)) & R_2 \\ \bar{F}(c(w_2), c(w_2 + 1)) + \sum_{j=0}^{\infty} \bar{F}(C(w_2), C(w_2 + 1)) & R_3 \\ \bar{F}(d_1(w_2), d_1(w_2 + 1)) + \sum_{j=0}^{\infty} \bar{F}(D_1(w_2), D_1(w_2 + 1)) & R_4 \end{cases} \quad (15)$$

where  $R_1: 0 < w_2 < w_1/2$ ,  $R_2: \frac{w_1}{2} < w_2 < \frac{w_p}{2}$ ,  $R_3: \frac{w_p}{2} < w_2 < \mu T$  and  $R_4: \mu T < w_2 < w_p$ .

A remark on the derivation of (15) is appropriate. The transition probability is computed in terms of the inter-loss distribution by summing the probabilities that the inter-loss duration ends at any of the instants where the window size is at  $w_2$ . Depending on the value of  $w_2$  and  $w_1$ , TCP-R may take on the value  $w_2$  in the atypical cycle, or in any of the following typical cycles.

## A.2 Channels with small bandwidth-delay product ( $\beta > 1$ )

For this case, the summation in (13) can be written using (11) in terms of  $\bar{F}(\cdot)$  as before with the following difference - we have to differentiate between the situations when  $W_1/2 < \mu T$  and  $W_1/2 > \mu T$ . Recall for the non-random-loss case,  $w_p/2 > \mu T$ , and hence the typical cycles will always consist of a  $O(\sqrt{t})$  evolution between  $w_p/2$  and  $w_p$ . However, for the first cycle after a loss that occurs at  $W_1 = w_1$ , the cycle starts with window size  $w_1/2$ . If  $w_1/2 < \mu T$ , the window size will have a linear growth from  $w_1/2$  until  $\mu T$  and then a  $O(\sqrt{t})$  evolution (unless another random loss takes place). On the other hand, if  $w_1/2 > \mu T$ , the window size will have a  $O(\sqrt{t})$  growth directly from  $w_1/2$  without going into a linear phase first.

(i)  $w_1 < 2\mu T$

$$E[N|W_1 = w_1] = \sum_{n=0}^{N_a-1} \bar{F}(a(n)) + \sum_{n=0}^{N_b-1} \bar{F}(b_1(n)) + \sum_{j=0}^{\infty} \sum_{n=0}^{N_p-1} \bar{F}(B_2(n)) \quad (16)$$

$$P_{w_1, w_2} = \begin{cases} 0 & 0 < w_2 < w_1/2 \\ \bar{F}(c(w_2), c(w_2 + 1)) & \frac{w_1}{2} < w_2 < \mu T \\ \bar{F}(d_1(w_2), d_1(w_2 + 1)) & \mu T < w_2 < \frac{w_p}{2} \\ \sum_{j=0}^{\infty} \bar{F}(D_2(w_2), D_2(w_2 + 1)) & \frac{w_p}{2} < w_2 < w_p \end{cases} \quad (17)$$

(ii)  $w_1 > 2\mu T$

$$E[N|W_1 = w_1] = \sum_{n=0}^{N_b-1} \bar{F}(b_2(n)) + \sum_{j=0}^{\infty} \sum_{n=0}^{N_p-1} \bar{F}(B_2(n)) \quad (18)$$

$$P_{w_1, w_2} = \begin{cases} 0 & 0 < w_2 < \frac{w_1}{2} \\ \bar{F}(d_2(w_2), d_2(w_2 + 1)) & \frac{w_1}{2} < w_2 < \frac{w_p}{2} \\ \sum_{j=0}^{\infty} \bar{F}(D_2(w_2), D_2(w_2 + 1)) & \frac{w_p}{2} < w_2 < w_p \end{cases} \quad (19)$$

## A.3 Steady state distribution of the window size

The average throughput of a TCP-R session under the current model is given by

$$\rho = \left( \frac{E[N]}{E[X]} \right) \left( \frac{1}{\mu} \right) \quad (20)$$

where  $E[N]$  is the average number of packets successfully sent in an inter-loss duration, and  $E[X]$  is the average time between two random losses (which is equal to  $1/\lambda$ ).

$E[N]$  is given by

$$E[N] = \sum_{W=0}^{w_p} E[N|W] \pi(W) \quad (21)$$

where  $\pi$  is the steady state distribution of the MC.

MATLAB<sup>TM</sup> routines for solving the eigenvalue problem were used to compute  $\pi(W)$  for different values of  $\lambda$ ,  $\mu$ ,  $\tau$  and  $B$  and for different distributions for the inter-loss times. The results will be discussed in Section IV.

## B. Channels with Memory - Correlated Errors

For channels with memory, random packet loss can take place in bursts as well as in isolation. In this case, packet loss detection will be a mixture of TD's and TO's.

Let  $f$  denote the fraction of time the channel spends in the bad state (i.e. the steady state probability of being in the bad state), then

$$f = \frac{E[Y]}{E[X] + E[Y]}. \quad (22)$$

It is worth noting that in the limit  $1/\lambda_1 \rightarrow 0$ , this model becomes identical to the memoryless channel model with average inter-loss distribution of  $1/\lambda_0$ .

The difficulty in analyzing TCP for channels with memory lies in the delay in TCP response to changes in channel state. A

simple example shows this aspect (Figure 3): consider the quiescent scenario where TCP is successfully transmitting packets (i.e. channel is in a good state) when the channel enters a bad state for a duration long enough to initiate multiple consecutive TO events. Suppose the most recent TO event caused TCP to set its timer expiry period to a suitably large value. Assume the bad state duration beyond the final TO event is sufficiently short such that only the current packet is lost, and then the channel enters the good state. Then, if this good state ends before the end of the timer-expiry period, such a good state sojourn is wasted since TCP will not attempt to retransmit during this period (the average throughput during this time is zero). This is an example when a good state duration is ‘skipped’ by TCP, which we refer to as ‘unsampled’ good state.

Our analytical model henceforth attempts to accurately describe the behavior of TCP in such environments. We make some simplifying assumptions that allows analytical modeling without sacrificing the essential characteristics of the TCP behavior.

Let  $t$  denote the time at which the channel enters the good state *and* TCP resumes transmitting packets. Then, the source will continue to successfully transmit packets for a duration  $X_i = x$  at which time the channel enters the bad state for a duration of time  $Y_i = y$ . Let  $U_i = u$  denote the time interval between  $t + x$  and the time TCP will start to *successfully* retransmit the lost packet(s) (Figure 3). Notice that  $u$  may include a number of good state durations that were not detected by TCP (i.e. unsampled states). From a modeling point of view, these unsampled good states are considered bad states, since the instantaneous throughput during these skipped states is zero.

Let  $W_1 = W(t^-)$ ,  $W = W(t + x^-)$  and  $W_2 = W(t + x + u^-)$ .

In the above definitions, we emphasize that the good state is one that is successfully used by the TCP source. Good channel durations in which TCP doesn’t attempt to transmit packets (i.e. unsampled states) do not result in a statistical renewal, and we will incorporate them into the effective bad state duration  $u$ .

The window evolution between  $t$  and  $t + x$  is identical to that depicted earlier in Section A. The evolution between  $t + x$  and  $t + x + u$  is however rather complicated, and we make the following simplifying approximations :

1. The distribution of the duration of stay in the good and bad states are exponential.
2. The initial timer-expiry period  $\Delta$  is constant, and is equal to  $\tau + B/\mu$ .
3. At the end of a sampled good state  $X_i = x$ , if the following bad state duration  $Y_i = y < \Delta$ , there are enough packets in transit to cause a TD (this assumption is not required if  $y > \Delta$ ) and further, the TD takes place immediately.

In light of the above assumptions, we will describe the behavior of TCP in two different cases;  $y > \Delta$  and  $y < \Delta$ .

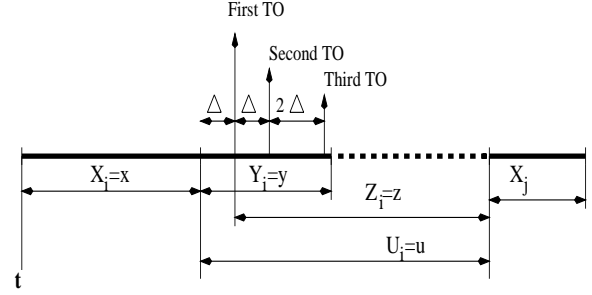


Fig. 3. Sketch of TCP behavior for  $y > \Delta$ .

### B.1 The case of $y > \Delta$

In this case, no packets will get through the channel until a TO takes place at  $t = x + \Delta$ , at which time TCP will retransmit the packet, set its timer-expiry period to  $\Delta$  and will wait for ACKs. However, this packet will be lost since the channel is in the bad state. If the channel state is good at  $t = x + 2\Delta$ , then TCP will resume its transmission, else, another TO will take place followed by a retransmission, and the timer expiry will now be set to  $2\Delta$ . This doubling of the TO expiry period continues up to  $64\Delta$  after which it remains constant. This process of timer expiry and retransmission continues until TCP successfully transmits a packet in some good state, which we call a sampled good state (renewal). Let  $Z$  denote the time from the first TO event (at  $t + x + \Delta$ ) until the first successful packet transmission ( $U = Z + \Delta$ ). Then the above scenario shows that  $Z$  will take only multiple values of  $\Delta$  (Figure 3).

Since  $y > \Delta$ ,  $S(t + x + \Delta) = 1$  is the channel state just after the first TO event. Let  $\delta > 0$  and define

$$\begin{aligned} P_\delta &= Pr(S(t + x + \Delta + \delta) = 0 | S(t + x + \Delta) = 1) \\ &= Pr(S(\delta) = 0 | S(0) = 1) \\ &= (1 - f)(1 - e^{-(\lambda_0 + \lambda_1)\delta}) \end{aligned} \quad (23)$$

where we used the Memoryless (assumption (1)) property in (23) together with a first order analysis of the transient behavior of the channel two-state MC.

Then,

$$Pr(Z = \Delta) = Pr(S(\Delta) = 0 | S(0) = 1) = P_\Delta \quad (24)$$

$$Pr(Z = 3\Delta) = P_{2\Delta}(1 - P_\Delta) \quad (25)$$

where we used the Markov property in (24) and (25). The p.d.f of  $Z$  conditioned on  $Y = y$  ( $y > \Delta$ ) can be computed, and after some algebraic manipulations,

$$E[Z|Y = y] = a + \frac{\alpha b(191 - 127b)}{(1 - b)^2}, \quad y > \Delta \quad (26)$$

where

$$M_1(k) = \prod_{i=1}^k (1 - P_{2^{i-1}\Delta}),$$

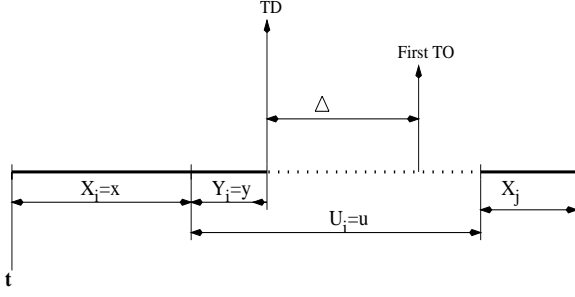


Fig. 4. Sketch of TCP behavior for  $y < \Delta$ .

$$a = \sum_{k=1}^6 [(2^{(k+1)} - 1)\Delta P_{2^k \Delta} M_1(k)],$$

$$\alpha = \Delta P_{64\Delta} M_1(6),$$

and

$$b = 1 - P_{64\Delta}.$$

### B.2 The case of $y < \Delta$

The treatment of the case where  $y < \Delta$  is similar (see Figure 4). In this case, the channel enters the bad state at time  $t + x$ , and since we assume that the channel has enough packets in transit (assumption (3)), a TD will take place at  $(t + x + y)$ . Now, we further assume that if  $S(t + x + y + 1/\mu) = 0$ , this packet will be successfully transmitted and  $u = y$ , otherwise, a TO will take place at  $(t + x + y + \Delta)$  and the behavior follows just as in the case of  $y > \Delta$  (i.e. a sequence of TO and retransmissions until a successful packet transmission takes place at a sampled good state). The assumption made here for  $S(t + x + y + 1/\mu)$  is mainly inspired by our observations from tracing *ns* simulations. That is, we have observed that when a the good state is shorter than  $1/\mu$ , TCP almost always skips some good states and ends up in a TO.

The subsequent analysis parallels the case  $y > \Delta$ , and after some algebraic manipulations,

$$E[U|Y = y] = Cy + D, \quad y < \Delta \quad (27)$$

where

$$C = c_1 + c_2 + \alpha_1 \frac{b}{(1-b)},$$

$$D = d_1 + d_2 + 64\Delta \alpha_1 b \frac{(2-b)}{(1-b)^2},$$

$$c_1 = P_{\frac{1}{\mu}},$$

$$c_2 = \sum_{i=1}^6 Pr[U = y + 2^i \Delta],$$

$$\alpha_1 = (1 - P_{\frac{1}{\mu}})(1 - P_{\Delta})^2 M_2(6),$$

$$d_1 = \frac{1}{\mu} P_{\frac{1}{\mu}},$$

$$d_2 = \sum_{i=1}^6 2^i \Delta Pr[U = y + 2^i \Delta],$$

and for  $2 \leq k \leq 5$

$$Pr(U = y + \Delta) = (1 - P_{1/\mu})P_{\Delta}$$

$$Pr(U = y + 2\Delta) = (1 - P_{1/\mu})(1 - P_{\Delta})P_{\Delta}$$

$$Pr(U = y + 4\Delta) = (1 - P_{1/\mu})(1 - P_{\Delta})^2 P_{2\Delta}$$

$$Pr(U = y + 2^{(k+1)}\Delta) = (1 - P_{1/\mu})(1 - P_{\Delta})^2 M_2(k)$$

$$M_2(k) = P_{2^k \Delta} \prod_{i=1}^{k-1} (1 - P_{2^i \Delta})$$

And finally, using (26) and (27),

$$E[U] = \int_{y=0}^{\Delta} E[U|Y = y] f_Y(y) dy$$

$$+ (\Delta + E[Z])(1 - F_Y(\Delta)) \quad (28)$$

where the expression in (28) can be computed in a closed form.

### B.3 Steady State Distribution of the Window Size

Let  $X_j$  denote the used durations of the sampled good states, and  $U_j$  denote the durations of the time between the end of the good sampled state  $j$  and the beginning of the next good sampled state  $j + 1$ . Since  $\{X_1 + U_1, X_2 + U_2, \dots\}$  are independent and identically distributed (IID) random variables, the sequence of window sizes  $\{W_1, W_2, \dots\}$  just before the transition to a sampled good state form a finite state MC (the number of states is  $W_p/2$ ).

The conditional expected number of successful packet transmissions during a good state  $E[N|W_1]$  can be computed exactly as in the case of memoryless channels and will be also given by (14), (16) and (18). The only difference is that  $w_1/2$  in the above equations will be replaced by  $w_1$  since  $W_1$  here denotes the starting window size and not the window size before packet loss.

In order to compute the MC transitional probabilities  $P_{w_1, w_2}$ , let us first compute  $P_{w_1, w}$  and  $P_{w, w_2}$ .  $P_{w_1, w}$  has the same expression as  $P_{w_1, w_2}$  in (15), (17) and (19) where  $W_2$  and  $w_2$  in these equations are replaced by  $W$  and  $w$  respectively and  $w_1/2$  is replaced by  $w_1$ .

Using assumptions (1)-(3),  $P_{w, w_2}$  can be computed in terms of the statistics of  $Y$  and the value of  $\Delta$  as follows. If a TD takes place, the window size is set to  $W/2$ . On the other hand, after the first TO at a window  $W$ , the window size is set to one and TCP reverts to slow-start with a threshold window  $W/2$ . If another consecutive TO occurs, then the window is reset to one and the slow start threshold is set to half the window size just before the second consecutive TO. We will approximate this behavior in case of TO's by assuming that, in case of one or more TO's, the window is set to one. This is a good approximation to the

transition probability in case of TO's since, in case of a single isolated TO, the window size is expected to recover to a high value before another random loss takes place, while, for multiple TO's, since the threshold is halved each time a TO takes place and no packets are ACKed in between TO's the window size will decrease exponentially in between TO's, making the scenario close to our assumption. Hence,

$$P_{w,w_2} = \begin{cases} Pr[Y < \Delta]P_{1/\mu} & w_2 = w/2 \\ Pr[Y < \Delta](1 - P_{1/\mu}) \\ + Pr[Y > \Delta] & w_2 = 1 \\ 0 & otherwise \end{cases} \quad (29)$$

Since  $Pr[W_2|W_1, W] = Pr[W_2|W] = P_{w,w_2}$ , we have

$$P_{w_1,w_2} = \sum_{w=1}^{w_p/2} P_{w_1,w} P_{w,w_2} \quad (30)$$

Finally, using the renewal reward theory and defining the reward as the number of successful packet transmissions, we can compute, although not in a close form, the steady state distribution of the MC  $\pi$ ,  $E[N]$  (from (21), where the sum is from  $W = 1$  to  $w_p/2$  instead of  $w_p$ ), and hence the average throughput

$$\rho = \left( \frac{E[N]}{E[X] + E[U]} \right) \left( \frac{1}{\mu} \right). \quad (31)$$

#### IV. MODEL VALIDATION USING THE *ns* SIMULATOR

In order to validate the analytical model, we used the well known *ns* simulator version 2.1b4a from UCB/LBNL. Since *ns* is under development, only a core of the TCP protocol suite as written in *ns* has been validated.

The two-state error model implemented by UCB/LBNL in *ns* does not fall in the validated portion of *ns* modules. We found that this implementation was not suitable for our purpose. since it uses the packet reception as the clock for proceeding the two-state channel Markov Chain, and hence, when packets are not being received (due to TO or empty buffer), the channel Markov Chain is "frozen". This contradicts with the physical layer modeling of the channel, where the channel state changes proceed in time independent of whether packets are or are not being received.

Hence, we embedded our own two-state continuous-time Markov Chain loss model. This required changing the two files "errmodel.cc", "errmodel.h" and "ns-errmodel.tcl" to redefine the mechanism of dropping packets. The new updated files as well as the simulation script is included in [9]. Figure 5 shows a schematic of the simulation topology. In this scenario, once a packet starts being received at the destination, a *receive* procedure is called. The procedure does not drop the packet only if the channel remains in the good state during the entire duration of the packet transmission time, otherwise, the packet is dropped. In the figure, 'S' and 'D' refers to successful and dropped packets, respectively, while 'G' and 'B' refers to the Good and Bad

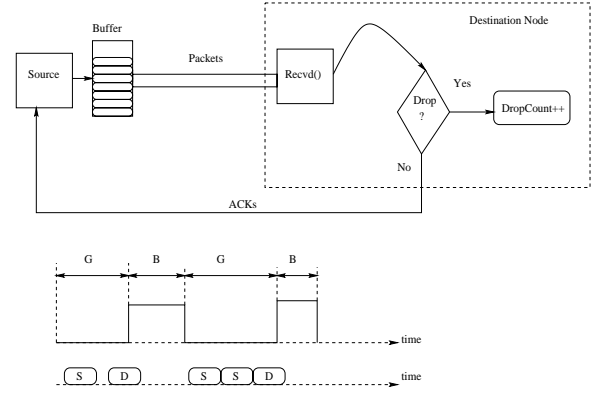


Fig. 5. NS simulation topology and packet dropping

states. In the simulation, an FTP application is attached to the TCP-R agent at the source. The simulation time is chosen long enough to guarantee the steady state as reached. This time is set to be at least  $\max\{100t_p, 100E[X + Y]\}$ .

TCP-R parameters are set to their default values. Equal packet size of 1000 bytes is used. The maximum receiver advertised window was set to 1000 (since we do not assume any limitations on the receiver's advertised window). We assume fine grained timers are used and hence *tcpTick* is set to 0.01.

Figures 6, 7 and 8 show a comparison between the analysis results and those obtained by simulation for the IID loss case assuming an exponential inter-loss distribution (refer to [10] for details of the calculation). We plot the average throughput against the average inter-loss duration. The figures show that the model provides accurate throughput predictions for a wide range of links.

To study the sensitivity of the results to the shape of the inter-loss distribution, we consider the Gamma family of densities

$$f_X(x) = \frac{\lambda^\alpha x^{(\alpha-1)} e^{-\lambda x}}{\Gamma(\alpha)} \quad (32)$$

with mean  $E[X] = \frac{\alpha}{\lambda}$ . For  $\alpha = 1$ , the Gamma distribution reduces to the exponential with parameter  $\lambda$ . Figure 9 shows the throughput behavior for the Gamma distribution with  $\alpha = 1, 2, \text{ and } 4$  and suggests that the dependence on the shape of the distribution within this family is weak.

Figures 10, 11 and 12 show a comparison between the analysis and the simulation results for the case of correlated packet loss. The throughput is plotted against the average holding time in the bad state. Each figure shows two pairs of curves, one for  $f = 1\%$  and the other for  $f = 10\%$ . Notice that, for a fixed  $f$ , the ratio of  $\bar{Y}$  to  $\bar{X}$  is fixed but the values of  $\bar{Y}$  and  $\bar{X}$  may vary. The analysis results agrees with those obtained by simulations for a wide range of links. While the parameters in Figure 10 represent a typical fast LEO link, those in Figure 12 represent a low speed LEO link. The parameters in Figure 11 are those of a typical high speed WAN link.

It is worth noting that the throughput behavior for channels

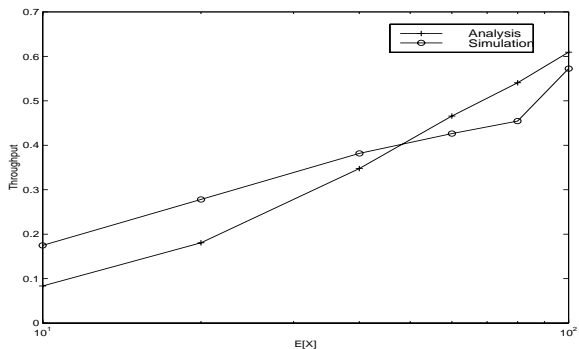


Fig. 6. Throughput comparison for the random packet loss case ( $\mu = 100, \beta = 0.5, \tau = 1.0, B = 50, w_p = 151$ )

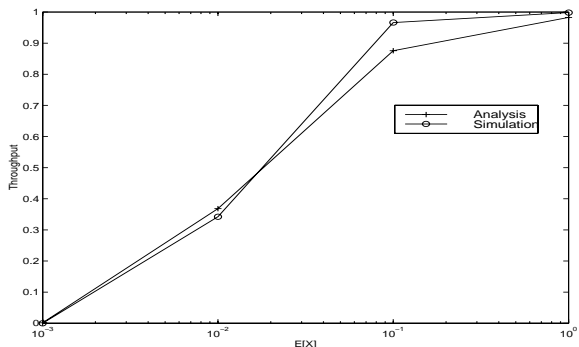


Fig. 7. Throughput comparison for the random packet loss case ( $\mu = 1000, \beta = 8.0, \tau = 0.001, B = 16, w_p = 18$ )

with memory (e.g. fading channels) exhibit a “ceiling” behavior for a fixed  $f$  (fading percentage) as  $\bar{Y}$  and  $\bar{X}$  increase. This observation has been also noted by [11]. The explanation of this is evident. When  $\bar{X}$  is ‘long’, TCP throughput in the good state approaches that of an ideal channel. Call this throughput value  $\rho_{free}$ . In the bad state, the throughput is zero. Also, since the good states are long, all good states are sampled. Hence, for a fixed  $f$ , the ceiling value  $\rho_{ceil}$  for slow channel transition frequency is given by

$$\rho_{ceil} = (1 - f)\rho_{free} \quad (33)$$

An empirical rule for deciding whether the frequency of channel transitions is fast or slow (i.e. fast or slow fading) is to compare  $\bar{X} + \bar{Y}$  to  $\Delta$  and  $t_p$ . If  $\bar{X} + \bar{Y} < \Delta$ , this is a fast fading channel. If  $\Delta < \bar{X} + \bar{Y} < t_p$ , then it is a moderate fading channel. Finally, if  $\bar{X} + \bar{Y} > 10t_p$ , then we have a slow fading channel and the throughput is given by (33). An example for applying this rule is shown in Table 1. The link parameters used are  $\mu = 100, \tau = 0.01$  and  $\beta = 8.0$ . Hence,  $t_p = 1.215$  and  $\Delta = 0.17$ . We fix  $f = 0.2$  (i.e.  $\bar{X} = 4\bar{Y}$ ).

## V. CONCLUSION

In this paper, we presented an analytical model of TCP-R. The model captures the essential behavior of TCP and can be

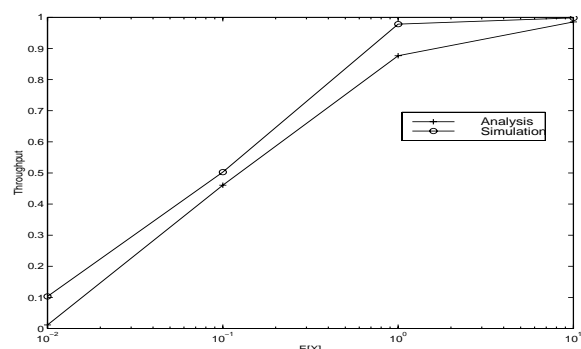


Fig. 8. Throughput comparison for the random packet loss case ( $\mu = 1000, \beta = 8.0, \tau = 0.01, B = 88, w_p = 99$ )

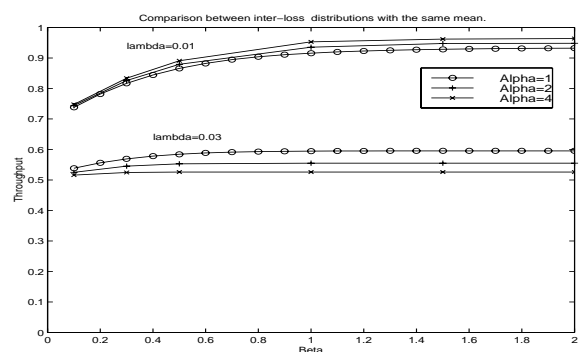


Fig. 9. Throughput comparison for different inter-loss distributions with the same mean ( $\mu = 100, \beta = 0.8, \tau = 1.0, B = 80, w_p = 181$ )

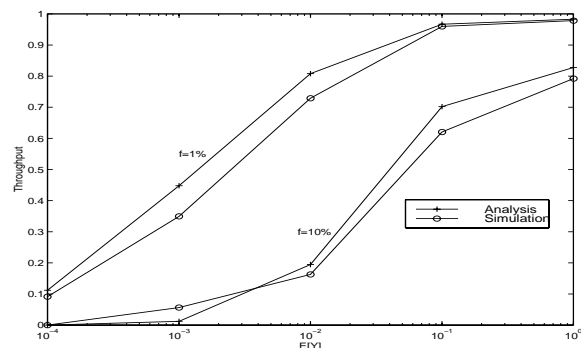


Fig. 10. Throughput comparison for the correlated packet loss case ( $\mu = 1000, \beta = 16.0, \tau = 0.01, B = 176, w_p = 187$ )

TABLE I  
AN EXAMPLE OF CHARACTERISING THE CHANNEL TRANSITION  
FREQUENCY (FADING FREQUENCY)

Fading	$\bar{X}$	$\bar{Y}$	Range	$\rho$
Fast	0.08	0.02	$\bar{X} + \bar{Y} < \Delta$	0.1591
Moderate	0.8	0.2	$\Delta < \bar{X} + \bar{Y} < t_p$	0.5367
Slow	8	2	$t_p < \bar{X} + \bar{Y} < 10t_p$	0.6973
Slow	80	20	$\bar{X} + \bar{Y} > 10t_p$	0.7189
Slow	800	200	$\bar{X} + \bar{Y} \gg 10t_p$	0.7480

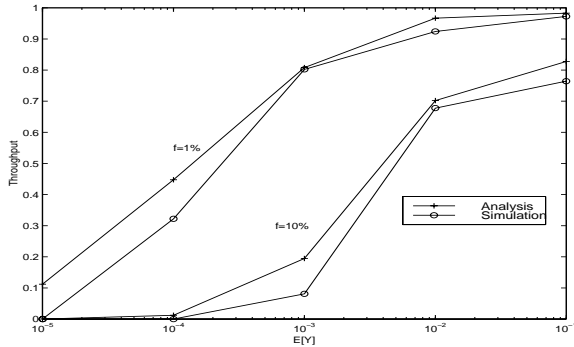


Fig. 11. Throughput comparison for the correlated packet loss case ( $\mu = 10000$ ,  $\beta = 16.0$ ,  $\tau = 0.001$ ,  $B = 176$ ,  $w_p = 187$ )

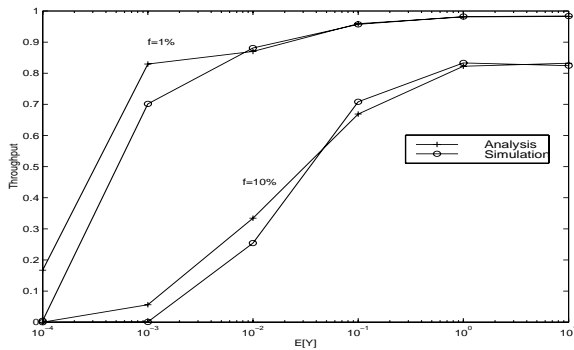


Fig. 12. Throughput comparison for the correlated packet loss case ( $\mu = 100$ ,  $\beta = 16.0$ ,  $\tau = 0.01$ ,  $B = 32$ ,  $w_p = 34$ )

tailored to specific TCP implementations. The model provides a close estimate of the throughput of TCP as measured against  $ns$  simulations for a variety of links (WANs/LANs). Equally accurate results are provided in the presence of IID losses as well as in bursty errors.

By modeling the channel statistics by a continuous-time finite-state MC, and accompanied by insight drawn from  $ns$  simulations, we were able to model the behavior of TCP in the presence of fast fading channels where we have shown that the effective good state durations (i.e. sampled good states) can be much less than expected resulting in a serious degradation of TCP throughput.

A number of avenues for future work remain. First, the model can be used to compare the performance of different TCP implementations. Second, with little additional effort, the effect of a fixed receiver advertised window size can be incorporated in the model. Third, the effect of the packet size, especially for fast fading channels, may be analyzed. Fourth, the effect of time-out granularity may also be studied. Finally, an extension of this model to incorporate the different packet drop policies would be valuable.

## REFERENCES

[1] T. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans-*

*actions on Networking*, vol. 5, no. 3, 1997.

[2] S. Shenker, L. Zhang, and D. D. Clark, "Some observations on the dynamics of a congestion control algorithm," *Computer Communications Review*, pp. 30–39, 1990.

[3] A. Kumar, "Comparative performance analysis of versions of TCP in a local network with a lossy link," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, 1998.

[4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *Proceedings of SIGCOMM'98*, 1998.

[5] A. Kumar and J. Holtzman, "Comparative performance analysis of versions of TCP in a local network with a lossy link, part II: Rayleigh fading mobile radio link," *Technical Report WINLAB-TR-133*, 1996.

[6] M. Zorzi and R. Rao, "Effect of correlated errors on TCP," *Proceedings of CISS'97*, 1997.

[7] A. Chockalingam, M. Zorzi, and R. Rao, "Performance of TCP on wireless fading links with memory," *Proceedings of ICC'98*, vol. 1, pp. 595–600, 1998.

[8] V. Jacobson, "Congestion avoidance and control," *Proceedings of ACM SIGCOMM '88*, 1988.

[9] A. Abou-Zeid, "Stochastic modeling of TCP/IP over lossy links," *M.S. Thesis, Dept. of Electrical Engineering University of Washington*, 1999.

[10] A. Abou-Zeid, Murat Azizoglu, and Sumit Roy, "Stochastic modeling of a single TCP/IP session over a link with random packet loss," *To appear, Proceedings of DIMACS Workshop on Mobile Networking and Computing*, 1999.

[11] M. Bhaskar and Murat Azizoglu, "Interference-robust TCP," *M.S. Thesis, Dept. of Electrical Engineering, University of Washington*, 1999.