A Stochastic Model of TCP Reno Congestion Avoidance and Control

CMPSCI Technical Report 99-02 *

Jitendra Padhye†, Victor Firoiu‡, Don Towsley†

†Dept. of Computer Science,
University of Massachusetts
Amherst, MA 01003
{jitu,towsley}@cs.umass.edu

‡Bay Architectures Lab,
Nortel Networks,
Billerica, MA 01821
vfiroiu@nortel.com

Abstract

The steady state performance of a bulk transfer TCP flow (i.e. a flow with a large amount of data to send, such as FTP transfers) may be characterized by three quantities. The first is the *send rate*, which is the amount of data sent by the sender in unit time. The second is the *throughput*, which is the amount of data received by the receiver in unit time. Note that the throughput will always be less than or equal to the send rate due to losses. Finally, the number of non-duplicate packets received by the receiver in unit time gives us the *goodput* of the connection. The goodput is always less than or equal to the throughput, since the receiver may receive two copies of the same packet due to retransmissions by the sender. In [9], we presented a simple model for predicting the steady state send rate of a bulk transfer TCP flow as a function of loss rate and round trip time. In this paper, we extend that work in two ways. First, we analyze the performance of bulk transfer TCP flows using more precise, stochastic analysis. We show that the predictions of the approximate model in [9] closely match the predictions of the more precise model, thus validating the approximate model. Second, we build upon the analysis in [9] to provide both an approximate formula as well as a more accurate stochastic model for the steady state throughput of a bulk transfer TCP flow.

*This material is based upon work supported by the National Science Foundation under Grant Nos. CDA-9502639 and NCR-9508274. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

The steady state performance of a bulk transfer TCP flow (i.e. a flow with a large amount of data to send, such as FTP transfers) may be characterized by three quantities. The first is the *send rate* (SR), which is the amount of data sent by the sender in unit time. The second is the *throughput*, which is the amount of data received by the receiver in unit time. Note that the throughput will always be less than or equal to the send rate due to losses. Finally, the number of non-duplicate packets received by the receiver in unit time give us the *goodput* of the connection. The goodput is always less than or equal to the throughput, since the receiver may receive two copies of the same packet due to retransmissions by the sender. In [9], we presented a simple model for predicting the steady state send rate of a bulk transfer TCP flow as a function of loss rate and round trip time. Note that in [9] used the term "throughput" to describe what we call "send rate" in this paper.

Here, we extend the work in [9] in two ways. First, we analyze the performance of bulk transfer TCP flows using more precise, stochastic analysis. We show that the predictions of the approximate model in [9] closely match the predictions of the more precise model, thus validating the approximate model. Second, we build upon the analysis in [9] to provide both an approximate formula as well as a more accurate stochastic model for the steady state throughput of a bulk transfer TCP flow.

Stochastic models of TCP have also been proposed in [5] and [8]. Our approach differs from these works in several ways. The model proposed in [8] is a fluid model, and measures the SR of the connection. The model does not take into account the effect of TCP timeouts. As a result, the closed form solution obtained in [8] resembles the formula derived in [6].

In [5], stochastic models for predicting the throughput of OldTahoe, Tahoe, Reno and NewReno flavors of TCP are proposed. The focus of the paper is on LAN-like environment with wireless links. The packet loss process is assumed to be Bernoulli and the impact of round trip time is not considered. In addition, timeouts are assumed to be limited to two successive backoffs. The actual timeout backoff mechanism [11] allows for six backoffs, and places no limit on the number of successive timeouts. The model, however, does take into account the impact of slow start and fast retransmits, which our model ignores.

The rest of this paper is organized as follows. In Section 2, we present the stochastic model for TCP-Reno, that predicts both the SR and the throughput of the connection. In Section 3, we compare the SR predicted by this model to the SR predicted by the approximate model proposed in [9]. Finally, in Section 4, we present an approximate formula, based on work in [9] that predicts the throughput of a TCP connection.

2 A model for TCP congestion control and avoidance

We investigate the impact of loss rate, limited receiver window size and average round trip time (RTT) on the long term, steady state SR of a TCP connection. We assume that the reader is familiar with TCP Reno congestion control [4, 11, 12] and the approximate analysis in [9]. We

adopt most of our terminology from [4, 11, 12, 9].

We model TCP's congestion avoidance behavior in terms of "rounds." A round starts with the back-to-back transmission of W packets, where W is the current size of the TCP congestion window. Once all packets falling within the congestion window have been sent in this back-to-back manner, no other packets are sent until ACKs are received for some or all of these W packets. We assume that ACKs arrive in a burst, just as the packets are sent in a burst. This ACK reception marks the end of the current round and the beginning of the next round. We assume that the duration of a round is equal to the round trip time and that it is independent of the window size. This assumption is also adopted (either implicitly or explicitly) in [6, 7, 8]. Note that we have also assumed here that the time needed to send all the packets in a window is smaller than the round trip time; this behavior can be seen in observations reported in [2, 10].

At the beginning of the next round, a group of W' new packets is sent, where W' is the new size of the congestion control window. Many TCP receiver implementations send one cumulative ACK for two consecutive packets received (i.e., delayed ACK, [12]). If W packets are sent in the first round and are all received and acknowledged correctly, then $W/2$ acknowledgments will be received. Since each acknowledgment increases the window size by $1/W$, the window size at the beginning of the second round is then $W' = W + 1/2$. That is, during congestion avoidance and in the absence of loss, the window size increases linearly in time, by one packet every two round trip times.

To model TCP's behavior in the presence of packet loss, we assume that a packet is lost in a round independently of any packet lost in *other* rounds. This assumption is justified to some extent by past studies [1] that have shown that periodic UDP packets that are separated by as little as 40ms tend to be lost only in singleton bursts. We assume, however, that packet losses are correlated among the back-to-back transmissions within a round: if a packet is lost, all remaining packets transmitted until the end of that round are also lost. This bursty loss behavior, which has been shown to arise from the drop-tail queuing discipline (adopted in many Internet routers), is discussed in [2, 3]. Packet loss can be detected in one of two ways, either by the reception at the TCP sender of "triple-duplicate" acknowledgments, i.e., four ACKs with the same sequence number, or via time-outs. In the former case, the sender reduces its window by half, and continues. In case of the latter, the sender waits for a retransmission timeout and reduces its window size to one. If the following loss event is also a timeout then the retransmission timer is exponentially backed off. See [12] for more details.

To model the impact of packet losses, it is useful to consider Figure 1. Consider a round in which TCP's window size is w. Let f_k be the last packet in this round that makes it to the receiver. Thus, packets $f_{k+1} \ldots f_w$ are lost. The sender will receive ACKs for the first k packets that made it to the receiver. This will result in a subsequent "short" round, consisting of k packets. Assume that m packets from this short round make it through. Then the sender will get m duplicate ACKs for packet f_k. If three or more duplicate ACKs are received, the sender reduces its window size by

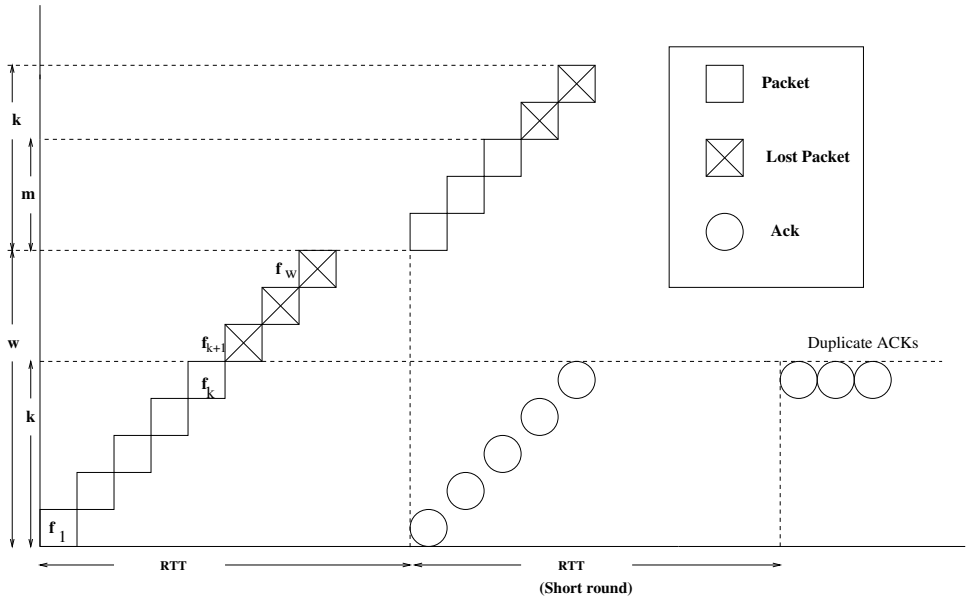

Figure 1: Packet and ACK transmissions preceding a loss indication

half and continues. Otherwise, it waits for a retransmit timeout and reduces its window to one.

We do not model certain aspects of TCP's behavior (e.g., fast recovery) but believe we have captured the essential elements of TCP behavior, as indicated by the generally very good fits between model predictions and measurements made on numerous commercial TCP implementations, as discussed in [9]. Also note that in the following, we measure SR and throughput in terms of packets per unit time, instead of bytes per unit of time.

2.1 The stochastic model

To formally model the behavior of TCP in terms of rounds, we define several sequences of random variables that define the state space of the modeled system.

- $\{W_i\}_{i=0}^{\infty}, \quad 1 \leq W_i \leq W_{max}$.
  W_i is window size for round i, $i = 0, \ldots \infty$.

- $\{C_i\}_{i=0}^{\infty}, \quad C_i = 0, 1$.
  C_i allows us to model the increment of window by one every two rounds during no-loss period. $C_i = 0$ indicates first of these two rounds and $C_i = 1$ indicates the second. We impose the following constraint on the value of C_i: If $W_i = W_{max}$, then $C_i = 0$. This captures the fact that the window size cannot grow beyond W_{max}.

- $\{L_i\}_{i=0}^{\infty}, \quad 0 \leq L_i < W_{i-1}$.
  L_i is the number of packets lost in the $(i-1)^{st}$ round, $i = 0, \ldots \infty$. We define $L_0 = 0$. If $L_i = 0$ it is an indication that either no packets were lost in the previous round, or the previous round was a short round and effects of packet loss have already been accounted for

via a triple duplicate event or a timeout event. $L_i > 0$ indicates that this round is a short round, and is a result of L_i packets being lost in the previous round. For short rounds, C_i is always equal to 0. In other words, if $L_i > 0$, then $C_i = 0$.

- $\{T_i\}_{i=0}^{\infty}, \quad 0 \leq T_i \leq 7.$
  T_i denotes whether the connection is in a timeout state in round i, $i = 0 \ldots \infty$. $T_i = 0$ if transmission of packets in this round was not a result of a retransmission timer expiration. In other words, $T_i = 0$ if a timeout did not occur as a result of packet loss in the previous round. Moreover, $T_i = 1, \ldots, 7$, if timeout did occur in the previous round, and the backoff value was $1, 2, 4, \ldots, 64$, respectively.

- $\{R_i\}_{i=0}^{\infty}, \quad RTT \leq R_i \leq 64 * T_0.$
  R_i denotes the duration of round i, $i = 0, \ldots \infty$. RTT denotes the round trip time, and T_0 denotes the base timeout value. During a timeout sequence R_i can have value of $T_0, 2T_0, \ldots 64 * T_0$, as explained earlier. All other rounds have a length of RTT.

- $\{N_i\}_{i=0}^{\infty}, 1 \leq N_i \leq W_{max}.$
  N_i denotes the number of packets transmitted in round i, $i = 0, \ldots \infty$.

- $\{M_i\}_{i=0}^{\infty}, 0 \leq M_i \leq W_{max}.$
  M_i denotes the number of packets transmitted in round i, $i = 0, \ldots \infty$, that make it to the receiver.

The sequence of random variables, $\{(W_i, C_i, L_i, T_i)\}_{i=0}^{\infty}$ is a finite state Markov chain with probability transition matrix, $\mathbf{Q} = [q_{v,a,k,s;w,c,l,t}]$, which we will soon calculate. Two states of this MC are transient, namely: $(1, 0, 0, 0)$ and $(1, 1, 0, 0)$. This is because we assume that a TCP sender starts with a window size of 1. However, once the window reaches size of 2, it is never reduced to 1 unless either a timeout occurs ($T_i > 0$) or the previous round had size k and $k - 1$ packets got lost, ($W_{i-1} = k, L_i = k - 1$). As a result, there are no transitions *to* these two states *from* rest of the states in the MC. It can be easily seen, from the transition probability matrix defined later in this section, that the remaining states in the MC form an irreducible sub-chain, and that this subchain is aperiodic. Define set S as the set consisting of the states of this subchain. Consequently the following limiting probabilities exist,

$$\pi_{w,c,l,t} = \lim_{i \to \infty} P(W_i = w, C_i = c, L_i = l, T_i = t), \quad (w, c, l, t) \in S$$

and they satisfy

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{Q} \tag{1}$$

Let $\{(W_r, C_r, L_r, T_r)\}$ and $\{(W_{r'}, C_{r'}, L_{r'}, T_{r'})\}$ two successive states in the sequence $\{(W_i, C_i, L_i, T_i)\}_{i=0}^{\infty}$. Let:

$$W_r = v, \ C_r = a, \ L_r = k, \ T_r = s, \quad (v, a, k, s) \in S$$

$$W_{r'} = w,\ C_{r'} = c,\ L_{r'} = l,\ T_{r'} = t, \quad (w, c, l, t) \in S$$

The number of packets sent during this transition is denoted by N_r and the time taken for the transition is denoted by R_r. Of the N_r packets, M_r packets make it to the receiver. Define the conditional expectations:

$$n_{v,a,k,s;w,c,l,t} = E[N_r | W_r = v, C_r = a, L_r = k, T_r = s, W_{r'} = w, C_{r'} = c, L_{r'} = l, T_{r'} = t]$$

$$m_{v,a,k,s;w,c,l,t} = E[M_r | W_r = v, C_r = a, L_r = k, T_r = s, W_{r'} = w, C_{r'} = c, L_{r'} = l, T_{r'} = t]$$

$$r_{v,a,k,s;w,c,l,t} = E[R_r | W_r = v, C_r = a, L_r = k, T_r = s, W_{r'} = w, C_{r'} = c, L_{r'} = l, T_{r'} = t]$$

These conditional expectations, along with the transition probabilities $q_{v,a,k,s;w,c,l,t}$, capture the behavior of TCP's congestion control and avoidance algorithm. Define:

$$\begin{aligned} N &= \lim_{i \to \infty} N_i \\ M &= \lim_{i \to \infty} M_i \\ R &= \lim_{i \to \infty} R_i \end{aligned}$$

For given time $t > 0$, define N_t to be the number of packets transmitted in interval $[0, t]$, and M_t to be the number of packets that make it to the receiver. Then the SR during that interval is given by: N_t/t, and the throughput by M_t/t.

The long term, steady state SR of a TCP connection is then defined by:

$$\begin{aligned} \text{SR} &= \lim_{t \to \infty} \frac{N_t}{t} \\ &= \frac{E[N]}{E[R]} \\ &= \frac{\sum_{(v,a,k,s) \in S} \pi_{v,a,k,s} \sum_{(w,l,c,t) \in S} q_{v,a,k,s;w,c,l,t} n_{v,a,k,s;w,c,l,t}}{\sum_{(v,a,k,s) \in S} \pi_{v,a,k,s} \sum_{(w,l,c,t) \in S} q_{v,a,k,s;w,c,l,t} r_{v,a,k,s;w,c,l,t}} \end{aligned} \tag{2}$$

Similarly, the long term, steady state throughput of a TCP connection is defined by:

$$\begin{aligned} \text{Throughput} &= \lim_{t \to \infty} \frac{M_t}{t} \\ &= \frac{E[M]}{E[R]} \\ &= \frac{\sum_{(v,a,k,s) \in S} \pi_{v,a,k,s} \sum_{(w,l,c,t) \in S} q_{v,a,k,s;w,c,l,t} n_{v,a,k,s;w,c,l,t}}{\sum_{(v,a,k,s) \in S} \pi_{v,a,k,s} \sum_{(w,l,c,t) \in S} q_{v,a,k,s;w,c,l,t} r_{v,a,k,s;w,c,l,t}} \end{aligned} \tag{3}$$

We also need to formalize the "drop-tail" loss model. If the packet loss probability is p, then each packet in a given round may be lost with probability p, until a packet *is* lost. All the packets in the round, following the lost packet, are lost with probability 1. Losses in different rounds are independent of one another.

We now calculate the transition probabilities and the conditional expectations of packets sent in each round and time required to complete the round.

2.1.1 No packets are lost

When no packets are lost, the TCP window size increases by one every two round trip times. During each such round, a window worth of packets are sent in one round trip time. Therefore:

$$
\begin{array}{lcll}
q_{w,0,0,0;w,1,0,0} & = & (1-p)^w, & 1 \le w < W_{max} \\
q_{w,1,0,0;w+1,0,0,0} & = & (1-p)^w, & 1 \le w < W_{max} \\
q_{w,0,0,0;w,0,0,0} & = & (1-p)^w, & w = W_{max} \\
 & & & \\
n_{w,0,0,0;w,1,0,0} & = & w, & 1 \le w < W_{max} \\
n_{w,1,0,0;w+1,0,0,0} & = & w, & 1 \le w < W_{max} \\
n_{w,0,0,0;w,0,0,0} & = & w, & w = W_{max} \\
 & & & \\
m_{w,0,0,0;w,1,0,0} & = & w, & 1 \le w < W_{max} \\
m_{w,1,0,0;w+1,0,0,0} & = & w, & 1 \le w < W_{max} \\
m_{w,0,0,0;w,0,0,0} & = & w, & w = W_{max} \\
 & & & \\
r_{w,0,0,0;w,1,0,0} & = & RTT, & 1 \le w < W_{max} \\
r_{w,1,0,0;w+1,0,0,0} & = & RTT, & 1 \le w < W_{max} \\
r_{w,0,0,0;w,0,0,0} & = & RTT, & w = W_{max}
\end{array}
\tag{4}
$$

2.1.2 One or more packets are lost in a round

When one or more, but not all, packets are lost in a round, the following round is a "short" round, as shown in Figure 1. This transition is accompanied by sending of a window worth of packets in a round trip time. When all packets in a window are lost, there is no following "short" round, instead, the sender waits for a retransmission timeout, and the window size goes to one.

$$
\begin{array}{lcll}
q_{w,c,0,0;w-l,0,l,0} & = & p(1-p)^{w-l}, & 2 \le w \le W_{max}, \quad c = 0,1, \quad 1 \le l < w \\
q_{w,c,0,0;1,0,0,1} & = & p, & 1 \le w \le W_{max}, \quad c = 0,1 \\
 & & & \\
n_{w,c,0,0;w-l,0,l,0} & = & w, & 2 \le w \le W_{max}, \quad c = 0,1, \quad 1 \le l < w \\
n_{w,c,0,0;1,0,0,1} & = & w, & 1 \le w \le W_{max}, \quad c = 0,1 \\
 & & & \\
m_{w,c,0,0;w-l,0,l,0} & = & w-l, & 2 \le w \le W_{max}, \quad c = 0,1, \quad 1 \le l < w \\
m_{w,c,0,0;1,0,0,1} & = & 0, & 1 \le w \le W_{max}, \quad c = 0,1 \\
 & & & \\
r_{w,c,0,0;w-l,0,l,0} & = & RTT, & 2 \le w \le W_{max}, \quad c = 0,1, \quad 1 \le l < w \\
r_{w,c,0,0;1,0,0,1} & = & TO, & 1 \le w \le W_{max}, \quad c = 0,1
\end{array}
\tag{5}
$$

2.1.3 One or more packets are lost in a short round

The loss of one or more packets in a short round affects the number of duplicate ACKs received by the sender as shown in Figure 1. If the number of packets that make it to the receiver is greater than two, the sender will get at least three duplicate ACKs and it will reduce its window size to half. Otherwise, it will wait for a timeout, and reduce its window size to one. Note that if the short round consists of less than three packets, $(w < 3)$, then timeout will always occur. Note that for all the equations in (6), $0 < l < W_{max}$, and $1 < w + l \leq W_{max}$.

$$
\begin{array}{lcll}
q_{w,0,l,0;1,0,0,1} & = & 1, & 1 \leq w < 3 \\
q_{w,0,l,0;1,0,0,1} & = & \sum_{i=0}^{2} p(1-p)^i, & 3 \leq w < W_{max} \\
q_{w,0,l,0;\lfloor (w+l)/2 \rfloor,0,0,0} & = & \sum_{i=3}^{w-1} p(1-p)^i + (1-p)^w, & 3 \leq w < W_{max} \\
 & & & \\
n_{w,0,l,0;1,0,0,1} & = & w, & 1 \leq w < 3 \\
n_{w,0,l,0;1,0,0,1} & = & w, & 3 \leq w + l < W_{max} \\
n_{w,0,l,0;\lfloor (w+l)/2 \rfloor,0,0,0} & = & w, & 3 \leq w < W_{max} \\
 & & & \\
m_{1,0,l,0;1,0,0,1} & = & (1-p) & \\
m_{2,0,l,0;1,0,0,1} & = & p(1-p) + 2(1-p)^2 & \\
m_{w,0,l,0;1,0,0,1} & = & \frac{\sum_{i=0}^{2} ip(1-p)^i}{\sum_{i=0}^{2} p(1-p)^i}, & 3 \leq w < W_{max} \\
m_{w,0,l,0;\lfloor (w+l)/2 \rfloor,0,0,0} & = & \frac{\sum_{i=3}^{w-1} ip(1-p)^i + w(1-p)^w}{\sum_{i=3}^{w-1} p(1-p)^i + (1-p)^w}, & 3 \leq w < W_{max} \\
 & & & \\
r_{w,0,l,0;1,0,0,1} & = & T_0 - RTT, & 1 \leq w < 3 \\
r_{w,0,l,0;1,0,0,1} & = & T_0 - RTT, & 3 \leq w < W_{max} \\
r_{w,0,l,0;\lfloor (w+l)/2 \rfloor,0,0,0} & = & RTT, & 3 \leq w < W_{max}
\end{array}
\tag{6}
$$

2.1.4 Exponential Backoff

If a packet is retransmitted as a result of a timeout, the retransmission timer backs off exponentially, but not exceeding 64 times the base timeout value. The timeout sequence ends when the retransmitted packet is successfully ACKed.

$$
\begin{array}{lcll}
q_{1,0,0,i;1,0,0,\min(i+1,7)} & = & p, & 1 \le i \le 7 \\
q_{1,0,0,i;2,0,0,0} & = & 1-p, & 1 \le i \le 7 \\
 & & & \\
n_{1,0,0,i;1,0,0,\min(i+1,7)} & = & 1, & 1 \le i \le 7 \\
n_{1,0,0,i;2,0,0,0} & = & 1, & 1 \le i \le 7 \\
 & & & \\
m_{1,0,0,i;1,0,0,\min(i+1,7)} & = & 0, & 1 \le i \le 7 \\
m_{1,0,0,i;2,0,0,0} & = & 1, & 1 \le i \le 7 \\
 & & & \\
r_{1,0,0,i;1,0,0,\min(i+1,7)} & = & 2^{(i-1)}T_0, & 1 \le i \le 7 \\
r_{1,0,0,i;2,0,0,0} & = & RTT, & 1 \le i \le 7
\end{array}
\tag{7}
$$

2.1.5 Completing the transition matrix

Define Q' as the set of all elements of $\mathbf{Q}$ that have been defined in (4)-(7). Then:

$$
q_{v,a,k,s;w,c,l,t} = 0, \quad (v,a,k,s),(w,c,l,t) \in S \quad \text{and } q_{v,a,k,s;w,c,l,t} \notin Q' \tag{8}
$$

This completes the formal description of our model. In the next section, we present a comparison between this model and the model proposed in [9].

3 Comparison of Send Rates

The approximate model proposed in [9] leads to the following formula for computing steady state sending rate of a long term TCP flow:

$$
\text{SR} = \begin{cases}
\dfrac{\frac{1-p}{p} + W(p) + \frac{Q(p,W(p))}{1-p}}{RTT(W(p)+1) + \frac{Q(p,W(p))G(p)T_0}{1-p}} & \text{if } W(p) < W_{max} \\
 & \\
\dfrac{\frac{1-p}{p} + W_{max} + \frac{Q(p,W_{max})}{1-p}}{RTT(\frac{W_{max}}{4} + \frac{1-p}{pW_{max}} + 2) + \frac{Q(p,W_{max})G(p)T_0}{1-p}} & \text{otherwise}
\end{cases} \tag{9}
$$

where:

$$
\begin{array}{lcl}
W(p) & = & \frac{2}{3} + \sqrt{\frac{4(1-p)}{3p} + \frac{4}{9}} \\
Q(p,w) & = & \min\left(1, \frac{(1-(1-p)^3)(1+(1-p)^3(1-(1-p)^{w-3}))}{1-(1-p)^w}\right) \\
G(p) & = & 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6
\end{array}
\tag{10}
$$

We can numerically compute the SR predicted by the stochastic model, using (1), (2) and (4)-(8). We find that the predictions of the model match well with the approximate model. An example is shown in Figure 2. The round trip time is 253 milliseconds, the base timeout value is 2.45 seconds and the maximum window size allowed by the receiver is 10 packets. The closeness of the match validates the approximate model.

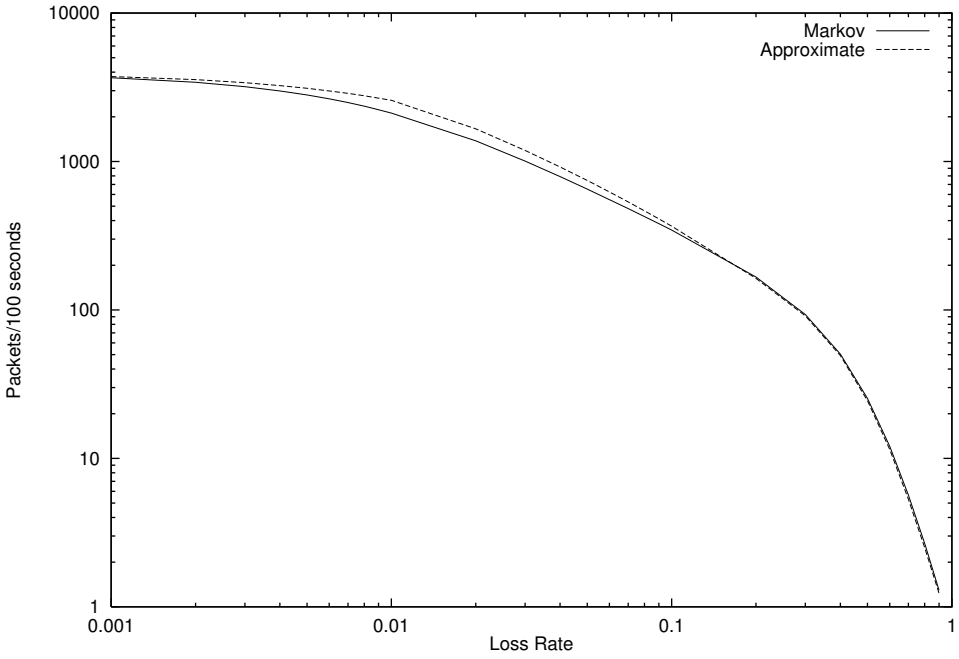

Figure 2: Comparison of detailed and approximate models: Send rates

4 Comparison of Throughputs

The formula proposed in [9] calculates the SR. The same analysis can be easily modified to calculate throughput. We start with a brief overview of the analysis in [9]. We model the behavior of TCP by considering the dynamics of the congestion window size. We ignore slow start, and assume that receiver sends delayed ACKs [12]. In the absence of packet loss, the congestion window size of the connection grows by one every two rounds, until it reaches the maximum window size allowed by the receiver, W_{max}. In the absence of loss, the congestion window size then remains constant at W_{max}. If a packet loss occurs, it may be detected by receipt of triple duplicate ACKs (TD) or a timeout (TO) [11]. In the case of a TD, the window size is reduced to half, while in case of TO, the sender waits for the timeout period and the congestion window size is reduced to one. In the case of successive TO events, the timeout period is backed off exponentially [11]. This gives rise to a regenerating pattern shown in Figure 3. The steady state SR is then given by:

$$\text{SR} = \frac{E[Y] + Q * E[R]}{E[A] + Q * E[Z^{TO}]} \tag{11}$$

where: $E[Y]$ is the expected number of packets sent in each TO period, $E[R]$ is the expected number of packets sent during the TO period, $E[A]$ is the expected duration of each TD period, $E[Z^{TO}]$ is the expected duration of the TO period, and $\frac{1}{Q}$ is the expected number of TD periods in the regeneration pattern. The analysis in [9] calculates each of these expected values under the drop-tail loss assumption described in Section 2. It should be clear that to calculate throughput, instead of SR, we only need to modify the numerator of (2). We need to calculate the number

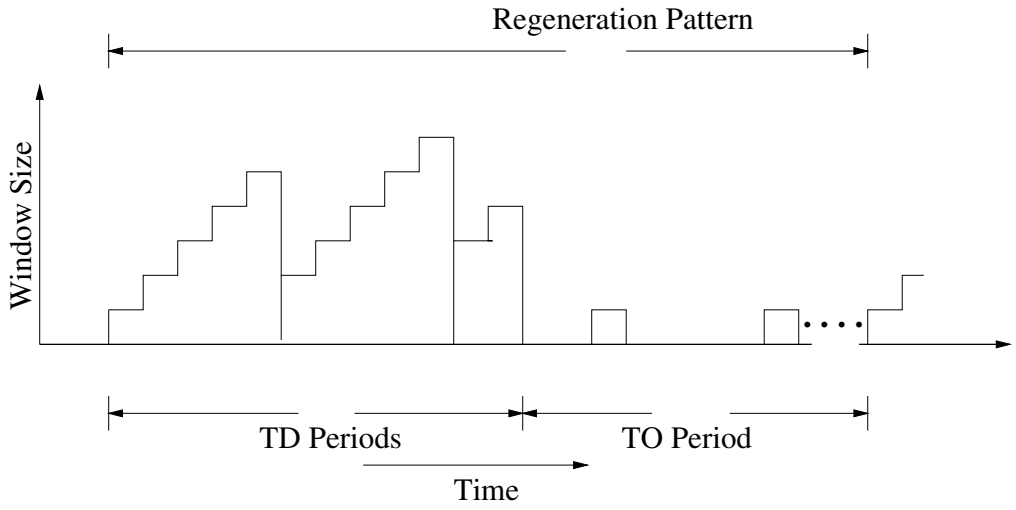

Figure 3: Regeneration Pattern

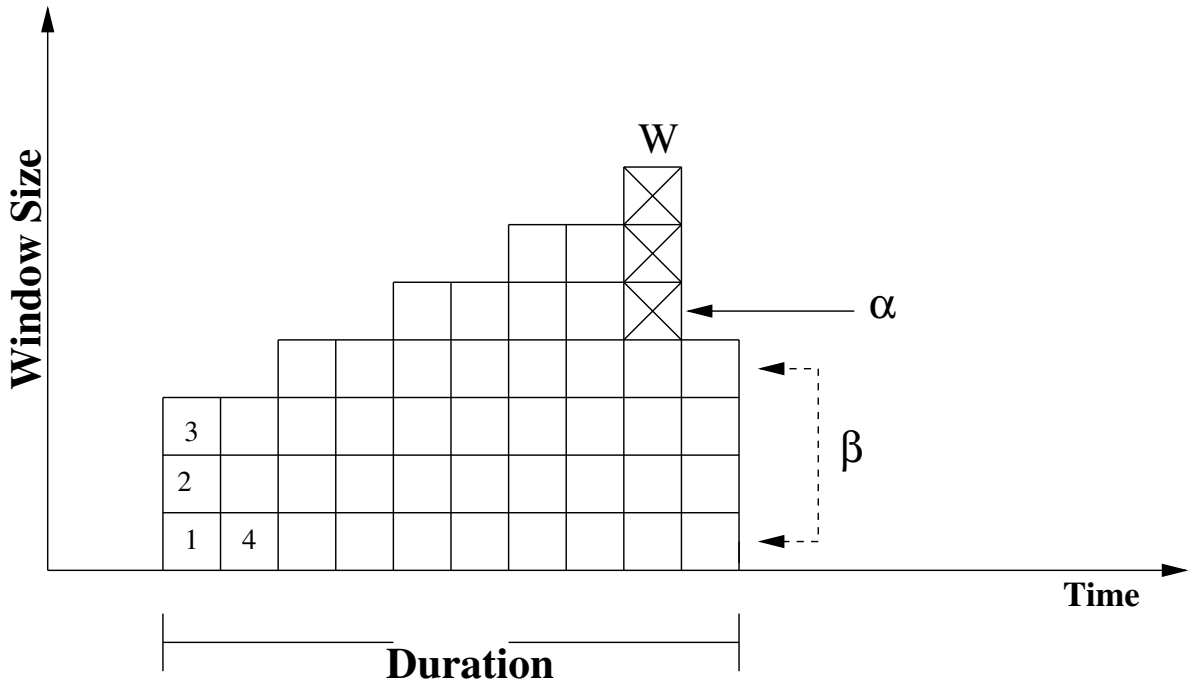

Figure 4: Analysis of TD Period

of packets that make it to the receiver in a TD period, (counterpart of $E[Y]$) and in the timeout sequence (counterpart of $E[R]$). Let us define these to be $E[Y']$ and $E[R']$, respectively. We can then define throughput as:

$$\text{Throughput} = \frac{E[Y'] + Q * E[R']}{E[A] + Q * E[Z^{TO}]} \tag{12}$$

Since only one packet makes it to the receiver in a timeout sequence (i.e. the packet that ends it), it should be clear that:

$$E[R'] = 1 \tag{13}$$

To calculate the number of packets that reach the receiver in a TD period, consider Figure 4. The TD event is induced by the loss of packet α. Let the window size be W, when the loss occurs. Then, the number of packets received by the receiver is:

$$E[Y'] = E[\alpha] + E[W] - E[\beta] - 1 \tag{14}$$

In [9], we have shown that: $E[\alpha] = 1/p$ and $E[\beta] = E[W]/2$. From (13) and (14), along with the analysis for $E[W]$ and Q from [9], we get:

$$\text{Throughput} = \begin{cases} \frac{\frac{1-p}{p} + \frac{W(p)}{2} + Q(p,W(p))}{RTT(W(p)+1) + \frac{Q(p,W(p))G(p)T_0}{1-p}} & \text{if } W(p) < W_{max} \\ \\ \frac{\frac{1-p}{p} + \frac{W_{max}}{2} + Q(p,W_{max})}{RTT(\frac{W_{max}}{4} + \frac{1-p}{pW_{max}} + 2) + \frac{Q(p,W_{max})G(p)T_0}{1-p}} & \text{otherwise} \end{cases} \tag{15}$$

where $W(p), Q(p,w)$ and $G(p)$ are defined in (10). In Figure 5, we compare the output predicted by the approximate formula in (15) and the output predicted by the Markov model. The values of RTT, T_0 and W_{max}, are the same as used for plotting Figure 2, in the previous section. In addition, we have plotted the approximate SR computed using (9), to allow easy comparison with Figure 2. We can see that the output predicted by the approximate formula in (15) and the Markov model are comparable.

References

[1] J. Bolot and A. Vega-Garcia. Control mechanisms for packet audio in the Internet. In *Proceedings IEEE Infocom96*, 1996.

[2] K. Fall and S. Floyd. Simulation-based comparisons of Tahoe, Reno, and SACK TCP. *Computer Communication Review*, 26(3), July 1996.

[3] S. Floyd and V. Jacobson. Random Early Detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4), August 1997.

[4] V. Jacobson. Modified TCP congestion avoidance algorithm. Note sent to end2end-interest mailing list, 1990.

[5] A. Kumar. Comparative performance analysis of versions of TCP in local network with a lossy link. *IEEE/ACM Transactions on Networking*, 6(4), August 1998.

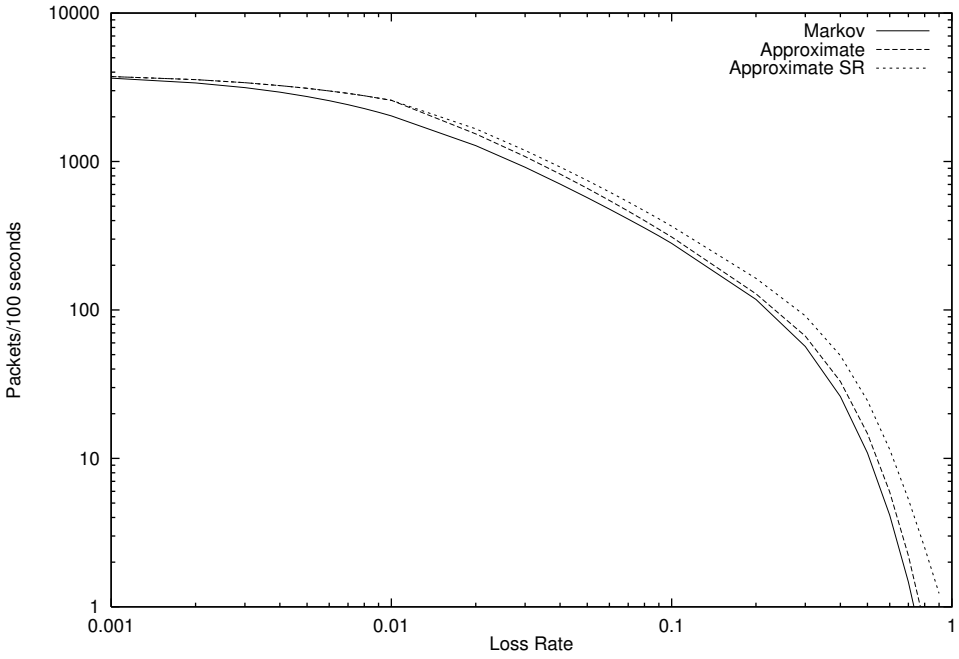

Figure 5: Comparison of detailed and approximate models: Throughput

[6] J. Mahdavi and S. Floyd. TCP-friendly unicast rate-based flow control. Note sent to end2end-interest mailing list, Jan 1997.

[7] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), July 1997.

[8] T. Ott, J. Kemperman, and M. Mathis. The stationary behavior of ideal TCP congestion avoidance. ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps.

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

[10] V. Paxson. Automated packet trace analysis of TCP implementations. In *Proceedings of SIGCOMM'97*, 1997.

[11] W. Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC2001, Jan 1997.

[12] W. Stevens. *TCP/IP Illustrated, Vol.1 The Protocols*. Addison-Wesley, 1997. 10th printing.

Appendix

A Aggregation of States

Note that the number of states in this MC is equal to $\frac{W_{max}(W_{max}+1)}{2} + W_{max} + 6$. However, it is possible to aggregate the states with $L_i > 0$, so that the resulting MC has only $2W_{max} + 7$ states.