

A COMPREHENSIVE STOCHASTIC MODEL FOR TCP LATENCY AND THROUGHPUT

D. Zheng, G. Y. Lazarou

Electrical and Computer Engineering

Mississippi State University

Box 9571

Mississippi State, MS 39762, USA

glaz@ece.msstate.edu

Abstract

Understanding the nature of TCP behavior is critical in order to properly engineer, operate, and evaluate the performance of the Internet, as well as to properly design and implement future networks. In this paper, we first develop a better and tractable model for the congestion window growth pattern in the slow-start phase. Using this new slow-start phase model, we construct an accurate model for the short-lived TCP flows and then an extended and more accurate TCP steady-state model. We validate our models with simulations and compare them against existing models. The results show that our model for the short-lived flows yields more accurate performance predictions (up to 20%) than the ones developed in [1] and [2]. In addition, our extended steady-state model is up to 75% more accurate than the model proposed in [3].

Key Words

TCP, performance evaluation, stochastic model

1. Introduction

A multitude of Internet applications, such as the world wide web, usenet news, file transfer and remote login, have opted TCP as the transport mechanism. Thus, TCP greatly influences the performance of Internet [4], [5], and a well-designed TCP is of utmost importance to the level of satisfaction of Internet users. Several stochastic TCP models have been proposed [1]-[3], [6] for predicting its performance in terms of latency and throughput. Considerable emphasis has been given into better understanding of the dynamics of TCP and its sensitivity to network

parameters, such as the TCP round trip time and the packet loss rate. Understanding the impact of TCP dynamics on its performance is critical for optimizing TCP and the design of active queue management techniques [7], [8] and TCP-friendly multicast protocols [9], [10]. However, TCP is a very complex protocol, and the fast-changing network conditions make the development of an accurate TCP stochastic model to be a very challenging task.

Stochastic models of TCP can be classified into three classes: (1) steady-state models for predicting the performance of bulk transfer flows [11], [3], (2) models for short-lived flows assuming low loss rates [12], [6], [13], and (3) models that combine the two above models [2], [1].

To the best of our knowledge, none of the steady-state models proposed so far account for the slow-start phase which begins at the end of every single time-out. The work in [3] assumes that the slow-start phase happens less frequently than the congestion-avoidance phase and the throughput in the slow-start phase is less than that in the congestion-avoidance phase, and that the slow-start phase can be ignored safely. While this could be the case with very small loss rates, the assumption does not hold in general. Empirical measurements show that 90% of the packet losses lead to time-outs [3]. Since TCP enters the slow-start phase when a time-out occurs, accurate TCP performance models must take into consideration of the aggregate effects of the slow-start phases.

All steady-state models assume the availability of unlimited data to send. Hence, the impact of the transient phase on performance is considered insignificant, and therefore is ignored. These models work well only for predicting the TCP send rate or the throughput of bulk data transfers, and are not applicable to predicting the performance of short-lived TCP flows.

It is noted in [14]-[15] that the majority of TCP traffic in the Internet consists of short-lived flows, i.e., the transmission comes to an end during the slow-start phase before switching to the congestion-avoidance phase. Hence, new models are needed that are capable of predicting the performance of short-lived TCP flows; and this is one of the main subject of our study.

In this paper, we first develop a better and tractable model for the congestion window growth pattern in the slow-start phase. Using this new slow-start phase model, we construct an accurate model for the short-lived TCP flows and then an extended and more accurate TCP steady-state model. Major improvement in both models is achieved by relaxing key assumptions and enhancing critical approximations that have been made in existing popular models.

The remainder of the paper is organized as follows. Section 2 presents an analysis in developing the improved and extended steady-state model. Section 3 builds the stochastic model for short-lived flows. In section 4, both models are validated with simulations and compared against existing models. Finally, Section 5 concludes the paper.

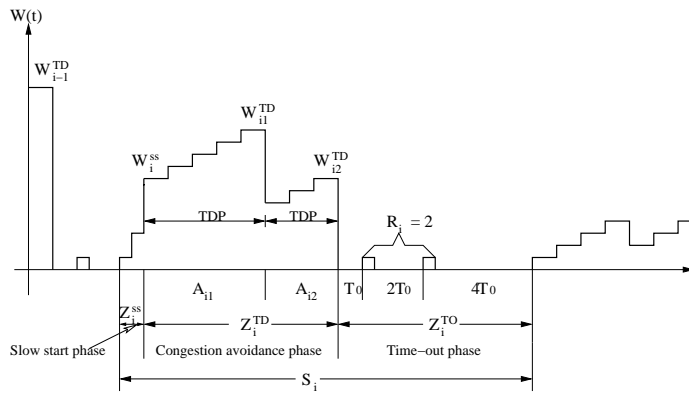


Figure 1. The extended steady state model - evolution of congestion window size when loss indications are triple-duplicate ACK's and time-outs.

2. Steady-state Model Incorporating the Slow-start Phase

2.1 Assumptions

As in [3], we develop our models based on the BSD TCP Reno release [16]. We assume that the link speed is very high, the round-trip time (RTT) remains fairly constant at all times, and the sender sends full-sized segments whenever the congestion window ($cwnd$) allows. The advertised window is assumed to be always a constant and large. Thus, the congestion window evolution alone determines the send rate, which roughly can be described as $cwnd/RTT$.

We model the dynamics of TCP in terms of “rounds” as done in [3]. A round starts when a window of packets is sent by the sender and ends when one or more acknowledgments are received for these packets. The effect of the delayed acknowledgment is taken into consideration, but neither the Nagle algorithm nor the silly window syndrome avoidance is considered. In addition, we assume that the packet losses are in accordance with the bursty loss model. The packet losses in different rounds are independent, but they are correlated within a single round; that is, if one packet in a round is lost, then the following back to back packets in the same round are also assumed to be lost. This is an idealization of the packet loss dynamics observed in the paths where FIFO drop-tail queues are used [1]. Finally, we assume that the sender has unlimited data to send.

2.2 Model Development

Fig. 1 depicts an instance of the congestion window's evolution over time. As shown in the figure, when a time-out occurs due to lost packets, TCP enters into the slow-start phase to recover from a perceived network congestion.

Let TDP be the period between two triple-duplicate (TD) losses, Z_i^{ss} be the time spent in the slow-start phase, Z_i^{TD} be the duration of the congestion-avoidance phase, and Z_i^{TO} be the time interval of the time-out phase. Let

M_i be the number of packets sent during the total time S_i . Then, we have that:

$$M_i = Y_i^{ss} + \sum_{j=1}^{n_i} Y_{ij} + R_i, \quad (1)$$

$$\begin{aligned} S_i &= Z_i^{ss} + Z_i^{TD} + Z_i^{TO} \\ &= Z_i^{ss} + \sum_{j=1}^{n_i} A_{ij} + Z_i^{TO} \end{aligned} \quad (2)$$

where Y_i^{ss} is the number of packets sent during the slow-start phase, A_{ij} is the duration of the j th TDP, n_i is the total number of the TDPs in the interval Z_i^{TD} , Y_{ij} is the number of packets sent during the j th TDP of interval Z_i^{TD} , and R_i is the number of packets sent during the time-out phase. W_i^{ss} is the window size at the end of a slow start and finally W_{ij}^{TD} is the window size at the end of the j^{th} TDP.

Assuming (S_i, M_i) to be a sequence of independent and identically distributed (i.i.d.) random variables, we determine the send rate as:

$$B = \frac{E[M]}{E[S]}$$

Considering n_i to be i.i.d. random variables and independent of Y_{ij} and A_{ij} , we have:

$$\begin{aligned} B &= \frac{E[Y^{ss}] + E[\sum_{j=1}^{n_i} Y_{ij}] + E[R]}{E[Z^{ss}] + E[\sum_{j=1}^{n_i} A_{ij}] + E[Z^{TO}]} \\ &= \frac{E[Y^{ss}] + E[n]E[Y] + E[R]}{E[Z^{ss}] + E[n]E[A] + E[Z^{TO}]} \end{aligned} \quad (3)$$

We next derive the closed form expressions for these expected values in the different TCP phases: the slow-start, the congestion-avoidance and the time-out phases.

1) *The Slow-Start Phase:* According to TCP Reno [17], [16], the current state of a TCP connection is determined based upon the values of the congestion window size ($cwnd$) and the slow-start threshold ($ssthresh$). If $cwnd$ is less than $ssthresh$, TCP is in the slow-start phase, otherwise, it is in the congestion-avoidance phase.

Since TCP has no knowledge of the network conditions, during the slow-start phases, it probes for the available bandwidth ‘‘greedily’’, i.e., it increases the $cwnd$ by one upon the receipt of a non-repeated acknowledgment. This algorithm can be formulated as:

$$cwnd_i = \lceil \frac{cwnd_{i-1}}{2} \rceil + cwnd_{i-1} \quad (4)$$

in which $cwnd_i$ is the congestion window size for the i^{th} round. (4) is due to the fact that assuming no loss, in round $(i - 1)$, there is a total of $cwnd_{i-1}$ packets sent to the destination, which, in turn, causes the receiver to generate $\lceil cwnd_{i-1}/2 \rceil$ acknowledgments¹. According to the slow-start algorithm, upon receiving these ACKs, the sender increases the $cwnd$ by the number of ACKs it has obtained, which is $\lceil cwnd_{i-1}/2 \rceil$.

¹ $\lceil x \rceil$ = the smallest integer bigger than x .

Noting that the congestion window is an integer, we can simplify (4) as follows²:

$$cwnd_i = \lceil \frac{3}{2} cwnd_{i-1} \rceil \quad (5)$$

Rearranging, we get:

$$\begin{aligned} \lceil \frac{cwnd_{i-1}}{2} \rceil &= \lceil \frac{1}{2} \lceil \frac{3}{2} cwnd_{i-2} \rceil \rceil \\ &\approx cwnd_{i-2} \end{aligned} \quad (6)$$

Substituting this in (4), we get the following:

$$cwnd_i \approx cwnd_{i-2} + cwnd_{i-1} \quad (7)$$

In order to examine the accuracy of this approximation, a typical evolution of $cwnd$ is given as follows:

$$1, 2, 3, 5, 8, 12, 18, 27, \dots$$

Compared with the sequence generated by (7):

$$1, 2, 3, 5, 8, 13, 21, 34, \dots$$

and the evolution of $cwnd$ proposed by the model in [1]:

$$1, 1.5^1, 1.5^2, 1.5^3, 1.5^4, 1.5^5, 1.5^6, 1.5^7, \dots$$

or calculated as:

$$1, 1.5, 2.25, 3.38, 5.06, 7.59, 11.39, 17.09\dots$$

The similarity between the two previous sequences and the discrepancy between the real evolution of $cwnd$ with the proposed model in [1] show that (7) gives a better approximation of the slow-start phase.

Noting that (7) generates the Fibonacci sequence, we can therefore express $cwnd$ as follows:

$$cwnd_n = C_1 X_1^n + C_2 X_2^n, \quad n = 1, 2, 3, \dots \quad (8)$$

where³:

$$X_{1,2} = \frac{1 \pm \sqrt{5}}{2} \quad (9)$$

C_1 and C_2 are determined by the initial value of $cwnd$. Assuming the initial value of $cwnd$ is 1, we get:

$$C_{1,2} = \frac{5 \pm \sqrt{5}}{10} \quad (10)$$

²In deriving a model for the latency of the short-lived TCP flows, (5) was approximated in [1] as: $cwnd_i = 3cwnd_{i-1}/2$.

³ X_1 is also called the golden number which will be denoted as g in the later parts of this paper.

By knowing the evolution of the congestion window, we can calculate the total number of packets, Y_n^{ss} , that are sent until the n_{th} round, by summing the congestion window size during each round:

$$\begin{aligned} Y_n^{ss} &= \sum_{i=1}^n cwnd_i \\ &= C_1 X_1^{n+2} + C_2 X_2^{n+2} - 2 \\ &\approx C_1 X_1^{n+2} - 2 \end{aligned} \quad (11)$$

The last approximation is due to the fact that:

$$C_2 X_2^{n+2} \leq \left| \frac{5 - \sqrt{5}}{10} \times \left(\frac{1 - \sqrt{5}}{2} \right)^3 \right| = 0.065$$

Thus, from (11), the number of rounds, n , can be computed as:

$$n = \log_g \left(\frac{Y_n^{ss} + 2}{C_1} \right) - 2 \quad (12)$$

Substituting (12) into (8), we can get the approximate relationship between the congestion window size and the total number of packets that have been sent, as follows:

$$cwnd_n = \frac{Y_n^{ss} + 2}{g^2} \quad (13)$$

Taking the expectation of both sides of (13), we have:

$$E[W^{ss}] = \frac{E[Y^{ss}] + 2}{g^2} \quad (14)$$

in which $E[W^{ss}]$ is the expectation of $cwnd_n$.

If the slow-start phase is ended by a packet loss, the expected data that have been sent during this phase can be calculated as:

$$E[Y^{ss}] = \frac{1 - p}{p} \quad (15)$$

where p is the loss rate.

Substituting the value of $E[Y^{ss}]$ in (14), we get:

$$E[W^{ss}]^* = \frac{1 + p}{pg^2} \quad (16)$$

This is the expected value of the congestion window when the slow-start phase ends due to a lost packet. Observing that when p is small, the expected value would be much bigger than the expected value of $ssthresh$, i.e.,:

$$E[W^{ss}]^* \gg E[ssthresh] = \frac{E[W^{TD}]}{2} \quad (17)$$

where the last equality comes from the fact that after each time-out, the slow-start threshold is set to half of the current congestion window W^{TD} .

and:

$$Y_i = \frac{X_i}{2} \left(\frac{W_{i-1}^{TD}}{2} + W_i^{TD} - 1 \right) + \beta_i \quad (24)$$

where X_i is the penultimate round in the TDP which experiences packet losses, r_{ij} is the round trip time, α_i is the number of packets sent in a TDP until the first loss happens, b is the number of packets acknowledged by a received ACK, and β_i is the number of packets sent in the fast retransmit phase, which is the last round [3].

Based on our assumptions, α_i is obviously geometrically distributed. Hence:

$$P[\alpha_i = k] = (1 - p)^{k-1} p, \quad k = 1, 2, \dots \quad (25)$$

and therefore, we have that:

$$E[Y] = \frac{1 - p}{p} + E[W^{TD}] \quad (26)$$

In addition, based on (23) and (24), we also have that:

$$E[X] = b \left(\frac{E[W^{TD}]}{2} + 1 \right) \quad (27)$$

$$E[Y] = \frac{E[X]}{2} \left(\frac{E[W^{TD}]}{2} + E[W^{TD}] - 1 \right) + E[\beta] \quad (28)$$

where we assume X_i and W_i^{TD} are mutually independent. Combining (26), (27), and (28), we get:

$$\begin{aligned} \frac{1 - p}{p} + E[W^{TD}] &= \frac{E[X]}{2} \left(\frac{E[W^{TD}]}{2} + E[W^{TD}] - 1 \right) + E[\beta] \\ &= \frac{b \left(\frac{E[W^{TD}]}{2} + 1 \right)}{2} \times \left(\frac{E[W^{TD}]}{2} + E[W^{TD}] - 1 \right) + E[\beta] \end{aligned} \quad (29)$$

Since β_i is the number of packets sent when k packets in the penultimate round are ACKed, its value equals to k with probability:

$$A(w, k) = \frac{(1 - p)^k p}{1 - (1 - p)^w} \quad (30)$$

Therefore:

$$\begin{aligned} E[\beta] &= E \left[\sum_{k=0}^{w-1} k \cdot P(\beta = k) | w \right] \\ &= E \left[\sum_{k=0}^{w-1} \frac{k(1 - p)^k p}{1 - (1 - p)^w} | w \right] \\ &= E \left[\frac{(1 - p)(1 - pw(1 - p)^{w-1} - (1 - p)^w)}{p(1 - (1 - p)^w)} | w \right] \\ &\approx (E[W^{TD}] - 1)(1 - p) \end{aligned} \quad (31)$$

for p small. Using (31) in (29) and rearranging, we get:

$$E[W^{TD}] = -\frac{2(b - 2p)}{3} + \sqrt{\frac{4(bp + 2(1 - p^2))}{3bp} + \left(\frac{2b - 4p}{3b} \right)^2} \quad (32)$$

Inserting (32) in (27), we obtain:

$$E[X] = \frac{(2p+3)b - b^2}{3} + \sqrt{\frac{b^2p + 2b(1-p^2)}{3p} + \left(\frac{b-2p}{3}\right)^2} \quad (33)$$

and:

$$\begin{aligned} E[A] &= (E[X] + 1)E[r] \\ &= RTT \left(-\frac{b^2 - (2p+6)b}{3} + \sqrt{\frac{b^2p + 2b(1-p^2)}{3p} + \left(\frac{b-2p}{3}\right)^2} \right) \end{aligned} \quad (34)$$

where we assume r_{ij} 's to be i.i.d. and $E[r] \approx RTT$.

In the previous subsection, we stated without proof that the slow-start phase will enter the congestion-avoidance phase before a packet loss happens. This can be proved if $E[W^{ss}]^*$, the expected congestion window size at the end of the slow-start phase due to a packet loss, is bigger than the value of $E[ssthresh] = E[W^{TD}]/2$, which is the expected threshold at the beginning of the slow-start phase. In other words, we need to show that:

$$\frac{1+p}{pg^2} \geq \frac{E[W^{TD}]}{2}, \quad (35)$$

where $E[W^{TD}]$ is given by (32). This is easily shown below, under the (normal) condition that p is small:

$$\begin{aligned} \frac{1+p}{pg^2} &\geq \sqrt{\frac{2}{3bp}} \\ \Leftrightarrow 1 - 0.3p + p^2 &\geq 0 \end{aligned}$$

The last inequality stands obviously. In fact, (35) is valid $\forall p \in [0, 1]$.

3) *The Time-out Phases:* The probability that a loss indication is a time-out under the current congestion window size w , is given in [3] as:

$$\min\left(1, \frac{(1 - (1-p)^3)(1 + (1-p)^3(1 - (1-p)^{w-3}))}{1 - (1-p)^w}\right) \quad (36)$$

which gets simplified when the loss rate, p , is small:

$$Q^{TD}(w) = \min\left(1, \frac{3}{w}\right)$$

Thus, Q^{TD} , the expected probability that a loss leads to a time-out at the end of the congestion-avoidance phase, is approximated in [3] as follows:

$$\begin{aligned} Q^{TD} &= E[Q^{TD}(w)] \\ &= \min\left(1, \frac{3}{E[W^{TD}]}\right) \end{aligned} \quad (37)$$

The traffic traces collected in [3] indicate that the effect of the time-outs must always be captured by any TCP performance prediction model. In most of the traces, time-out events out-numbered the fast retransmit events, i.e.,

Q^{TD} is around 90% of the total loss. This value is larger than the value given by the formula of (37), as we further calculated that the $E[W^{TD}]$ is greater than 10, which, in turn, renders the Q^{TD} to be less than 30%. So, we believe that this approximation underestimates the real Q^{TD} . As a matter of fact, the underestimation of Q^{TD} in [3] is due to the approximation of $E[1/W] \approx 1/E[W]$ by noting that:

$$\begin{aligned} E\left[\left(\frac{1}{\sqrt{W}}\right)(\sqrt{W})\right]^2 &\leq E\left[\left(\frac{1}{\sqrt{W}}\right)^2\right]E[(\sqrt{W})^2] \\ \implies \frac{1}{E[W]} &\leq E\left[\frac{1}{W}\right] \end{aligned}$$

The equality holds only when W is a constant.

Now, using Taylor's formula and expectation properties, we obtain the following⁵:

$$E\left[\frac{1}{W}\right] \approx \frac{1}{E[W]} \left(1 + \frac{Var(W)}{E[W]^2}\right) \quad (38)$$

Hence, to find a more accurate approximation of Q^{TD} , we must find the variance of W .

After a rigorous analysis⁶, we obtain the variance of W^{TD} , the congestion window size at the end of TDP, to be:

$$Var[W^{TD}]_{p \rightarrow 0} \approx \frac{8(\sqrt{3}-1)}{3bp} \quad (39)$$

Substituting (32) and (39) into (38), we get:

$$\begin{aligned} E\left[\frac{1}{W^{TD}}\right] &= \frac{1}{E[W^{TD}]} \left(1 + \frac{Var(W^{TD})}{E[W^{TD}]^2}\right) \\ &= \frac{1}{E[W^{TD}]} \left(1 + \frac{\frac{8(\sqrt{3}-1)}{3bp}}{\frac{8}{3bp}}\right) \\ &= \frac{\sqrt{3}}{E[W^{TD}]} \end{aligned} \quad (40)$$

(40) gives a better, but still simple, estimation of $E[1/W^{TD}]$. Then, Q^{TD} , the probability that a loss detection is a time-out (TO), can be found to be:

$$Q^{TD} \approx \min\left(1, \frac{3\sqrt{3}}{E[W^{TD}]}\right) \quad (41)$$

The probability of n_i , the number of TDPs, is derived according to Q^{TD} :

$$p(n_i = k) = (1 - Q^{TD})^{(k-1)} \cdot Q^{TD}$$

This is due to the fact that, with probability Q^{TD} , the packets lost at the end of the congestion control phase lead to a TO, and, with probability $1 - Q^{TD}$ the TCP connection stays in TDP. By taking the expectation of n_i , we get:

$$E[n] = \frac{1}{Q^{TD}} \quad (42)$$

⁵See Appendix 1 for the derivation.

⁶See Appendix 2 for details.

The expressions for the number of packets sent in the time-out phase, $E[R]$ and its duration, $E[Z^{TO}]$ are given in [3] as:

$$E[R] = \frac{1}{1-p} \quad (43)$$

$$E[Z^{TO}] = T_0 \frac{f(p)}{1-p} \quad (44)$$

where $f(p)$ is defined as:

$$f(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 15p^5 + 32p^6 \quad (45)$$

4) *The Steady State Send Rate and Throughput:* Substituting (19), (20), (26), (32), (34), (41), (42), (43) and (44) into (3), and taking into consideration the limitation of the window size [3], we finally derive the send rate as:

$$B = \begin{cases} \frac{\frac{E[W^{TD}]g^2}{2} - 2 + \frac{1}{Q^{TD}(E[W^{TD}])} \left(\frac{1-p}{p} + E[W^{TD}] \right) + \frac{1}{1-p}}{\left(\log_g \left(\frac{E[W^{TD}]}{2C_1} \right) + \frac{1}{Q^{TD}(E[W^{TD}])} \left(\frac{bE[W^{TD}] + b + 1}{2} \right) \right) RTT + \frac{f(p)T_0}{1-p}} & \text{when } E[W^{TD}] < W_m \\ \frac{\frac{W_m g^2}{2} - 2 + \frac{1}{Q^{TD}(W_m)} \left(\frac{1-p}{p} + W_m \right) + \frac{1}{1-p}}{\log_g \left(\frac{W_m}{2C_1} \right) RTT + \frac{1}{Q^{TD}(W_m)} \left(\left(\frac{b}{8} W_m + \frac{1-p}{pW_m} + 2 \right) + 1 \right) RTT + \frac{f(p)T_0}{1-p}} & \text{when } E[W^{TD}] \geq W_m \end{cases} \quad (46)$$

This can be further simplified as:

$$\min \left(\frac{W_m}{RTT}, \frac{1}{RTT \sqrt{\frac{2bp}{3}} + \min \left(1, 9\sqrt{\frac{bp}{8}} \right) p \left(\frac{RTT}{2} \log_g \left(\frac{2}{3bpC_1^2} \right) + T_0(1+32p^2) \right)} \right) \quad (47)$$

To derive the throughput, we only need to change $E[Y]$, the expected size of packets that have been sent in a TDP, to $E[Y']$, the expected size of packets that have been received in a TDP. $E[Y']$ can be expressed as:

$$E[Y'] = E[\alpha] + E[\beta] - 1 \quad (48)$$

where $E[\alpha]$ is $1/p$ and $E[\beta]$ is given by (31). Also we substitute $E[R]$ with $E[R']$, the expected number of packets received in the time out phase, where [3]:

$$E[R'] = 1$$

Thus, the throughput can be formulated as:

$$H = \frac{E[Y^{ss}] + E[n]E[Y'] + E[R']}{E[Z^{ss}] + E[n]E[A] + E[Z^{TO}]} \quad (49)$$

or:

$$H = \begin{cases} \frac{\frac{E[W^{TD}]g^2}{2} - 2 + \frac{1}{Q^{TD}(E[W^{TD}])} \left(\frac{1-p}{p} + (E[W^{TD}] - 1)(1-p) \right) + 1}{\left(\log_g \left(\frac{E[W^{TD}]}{2C_1} \right) + \frac{1}{Q^{TD}(E[W^{TD}])} \left(\frac{bE[W^{TD}] + b + 1}{2} \right) \right) RTT + \frac{f(p)T_0}{1-p}} & \text{when } E[W^{TD}] < W_m \\ \frac{\frac{W_m g^2}{2} - 2 + \frac{1}{Q^{TD}(W_m)} \left(\frac{1-p}{p} + (W_m - 1)(1-p) \right) + 1}{\log_g \left(\frac{W_m}{2C_1} \right) RTT + \frac{1}{Q^{TD}(W_m)} \left(\left(\frac{b}{8} W_m + \frac{1-p}{pW_m} + 1 \right) + 1 \right) RTT + \frac{f(p)T_0}{1-p}} & \text{when } E[W^{TD}] \geq W_m \end{cases} \quad (50)$$

which, when p is small, can be simplified as (47). This can be explained by noting that, if a loss seldom happens, then the send rate should just equal to the throughput.

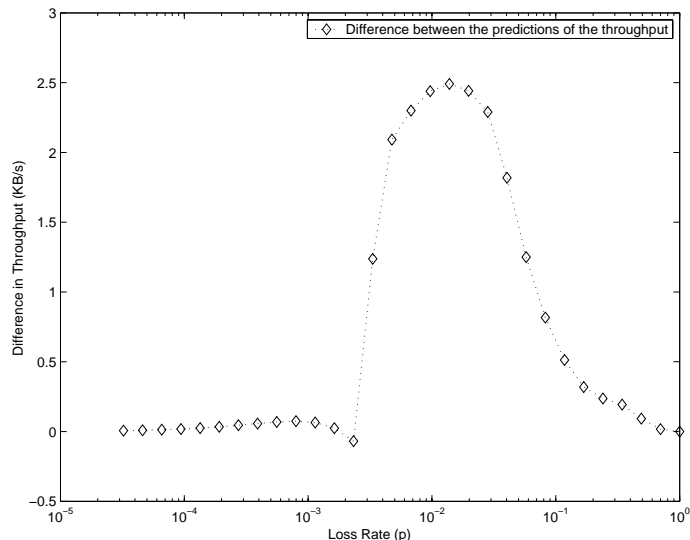


Figure 3. Our proposed model is compared with the one developed in [3] in terms of the predicted throughput difference versus the loss rate (p) for the case of: $RTT = 200ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$, $b = 2$.

Fig. 3 compares our model against the one proposed in [3]. It shows the predicted throughput difference versus p for the case of $RTT = 200 ms$, $MSS = 536$ bytes, $w_1 = 1$ segment, $T_0 = 1$ sec, $W_m = 20$ segments, and $b = 2$. With both models, when $p \rightarrow 0$, then $H \rightarrow W_m/RTT$. However, for $10^{-3} < p < 10^{-1}$, the model in [3] overestimates the throughput by up to a factor of 2.5 (at $p \approx 10^{-2}$). Obviously, when $p \rightarrow 1$, again both models obtain the same performance values.

3. Stochastic Model for Short-lived Flows

Our proposed model for the short-lived TCP flows is partially based on our results given in Section 2-2.1. In addition, it is composed of four parts according to a typical short-lived flow evolution: the start of the connection (three-way-handshake), the initial slow-start phase, the first loss, and the subsequent losses. We first derive the latency a flow experience in each part, and then sum them to obtain the total latency.

3.1 The Connection Start-up Phase

Every TCP connection starts with the three-way-handshake process. Assuming that no ACK packets can get lost, this process can be well modeled as follows [1]:

$$E[T_{twhs}] = RTT + T_s \left(\frac{1-p}{1-2p} - 1 \right) \quad (51)$$

where T_s is the duration of SYN time-out and p is the packet loss rate.

We further assume that two or more time-outs within the three-way-handshake process is very rare. Otherwise, the slow-start threshold would get set to one, and therefore, the connection would get forced directly into the congestion-avoidance phase instead of into the slow-start phase.

3.2 The Initial Slow-start Phase

After the three-way-handshake, the slow-start phase begins. In this phase, the sender's congestion window (*cwnd*) increases exponentially until either of the following two events occur: a packet gets lost or the *cwnd* reaches its maximum value W_m .

In order to derive the latency for this phase, $E[Y_{init}]$, the expected number of packets sent until a loss occurs is given by the following enhanced equation (based on the one given in [1]):

$$E[Y_{init}] = \frac{(1 - (1 - p)^d)(1 - p)}{p} \quad (52)$$

where d is the total file size measured in packets that must be transmitted.

Substituting (52) in (14), we obtain the expected congestion window size at the end of the slow-start phase due to packet losses as:

$$E[W_{init}] = \frac{(1 - (1 - p)^d)(1 - p) + 2p}{pg^2} \quad (53)$$

If $E[W_{init}] > W_m$, then the congestion window first grows to W_m and then remains there while sending the rest of the packets. Thus, the whole procedure is divided into two parts [1]. From (14), the number of packets sent when the *cwnd* grows to W_m is given by:

$$data_1 = g^2 \cdot W_m - 2 \quad (54)$$

Substituting (54) into (12), we can obtain the duration of this step measured in rounds:

$$n_1 = \log_g\left(\frac{W_m}{C_1}\right) \quad (55)$$

In the second part:

$$n_2 = \frac{E[Y_{init}] - data_1}{W_m} \quad (56)$$

rounds are needed to transmit the remaining $E[Y_{init}] - data_1$ packets.

Combining the previous results together and using (12) for the $E[W_{init}] \leq W_m$ case, the expected slow-start latency is computed as follows:

$$E[n] = \begin{cases} \lceil \log_g\left(\frac{W_m}{C_1}\right) \rceil + \frac{1}{W_m}(E[Y_{init}] - g^2 W_m - 2) & \text{when } E[W_{init}] > W_m \\ \lceil \log_g\left(\frac{E[Y_{init}] + 2}{C_1}\right) \rceil - 2 & \text{when } E[W_{init}] \leq W_m \end{cases} \quad (57)$$

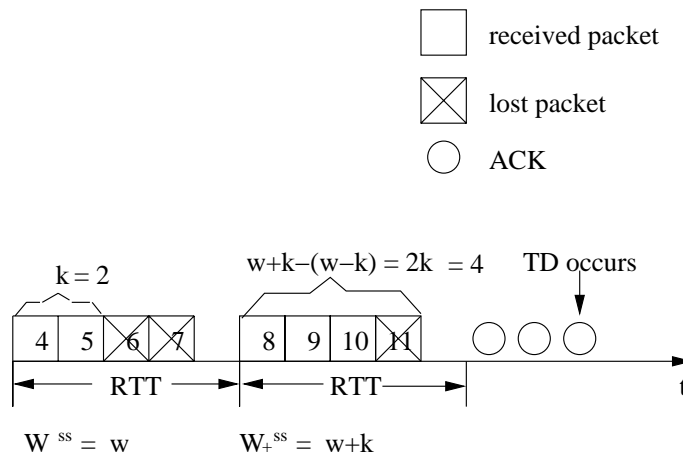


Figure 4. An illustration of a triple-duplicate (TD) event.

3.3 The First Loss

The initial slow-start phase ends when a packet loss is detected with a probability of $1 - (1 - p)^d$. When a packet gets lost, it could cause retransmission time-out (RTO) or lead to a triple duplicate ACKs, in which case TCP could recover in a round or two by using the fast retransmit and the fast recovery mechanism. We first derive the probability that a packet loss leads to a time-out (TO).

Due to the exponential growing pattern of $cwnd$ in the slow-start phase, Q^{ss} , the probability that a packet loss leads to a TO is different from the probability that when the sender is in the congestion-avoidance phase. With reference to Fig. 4, we derive the expression of Q^{ss} as follows.

In the round with a TD event, let W^{ss} be the current size of $cwnd$, which has a value w . In this round, w packets were sent. Among them, k packets are assumed to be ACKed. Since the connection is still in the slow-start phase, $cwnd$ increases to $w + k$ and another $2k$ packets are sent in the next round⁷. If more than three packets from these $2k$ packets get ACKed, then a TD would occur; otherwise, a TO would take place. Letting:

$$h(m) = \sum_{i=0}^2 (1-p)^i p \quad \text{if } m \geq 3 \quad (58)$$

be the probability that no more than 2 packets have been transmitted successfully in a round of m packets, we then obtain Q^{ss} to be:

$$Q^{ss}(W^{ss}) = \begin{cases} 1 & \text{for } W^{ss} \leq 2 \\ \sum_{k=0}^1 A(W^{ss}, k) + \sum_{k=2}^{W^{ss}-1} A(W^{ss}, k)h(2k) & \text{otherwise} \end{cases} \quad (59)$$

where $A(w, k)$ is as given by (30) and gives the probability that the first k packets have been successfully transmitted and ACKed in a round of w packets, provided that there might be one or more packets got lost. Simplifying (59),

⁷The delayed acknowledgment concept is not applied here, but we show later that it does not affect the analysis of the Q^{ss} .

we get $Q^{ss}(W^{ss})$ to be equal to:

$$\min\left(1, \frac{p(2-p) + (1-(1-p)^3)(1-p)^2(1-(1-p)^{W^{ss}-2})}{1-(1-p)^{W^{ss}}}\right) \quad (60)$$

As p approaches zero, (60) reduces to:

$$Q^{ss} = \lim_{p \rightarrow 0} E[Q^{ss}(W^{ss})] = \min\left(1, \frac{2}{E[W^{ss}]}\right) \quad (61)$$

In case of delayed acknowledgment, k successfully received packets generate $\lfloor k/2 \rfloor^8$ ACKs, and thus the size of the *cwnd* increases to $\lfloor k/2 \rfloor + w$ and $\lfloor k/2 \rfloor + k$ packets are sent. Therefore Q^{ss} can be computed as:

$$Q^{ss} = \begin{cases} 1 & \text{for } W^{ss} \leq 2 \\ \sum_{k=0}^1 A(W^{ss}, k) + \sum_{k=2}^{W^{ss}-1} A(W^{ss}, k)h(\lfloor \frac{k}{2} \rfloor + k) & \text{otherwise} \end{cases}$$

which is same as (60) since:

$$h(2k) = h(\lfloor \frac{k}{2} \rfloor + k) \quad \text{for } k \geq 2$$

The expected time that TCP spends in the RTOs is given by (44). The time that TCP spends in the fast retransmit phase, n_t , depends on where the loss would happen [2]:

$$n_t = \begin{cases} 2RTT & \text{if the lost packet is in the last three packets of the window} \\ RTT & \text{otherwise} \end{cases} \quad (62)$$

Thus, when the congestion window size W^{ss} is bigger than three, the expected time, $E[n_t]$ can be found to be:

$$\begin{aligned} E[n_t] &= \frac{1-(1-p)^{W^{ss}-3}}{1-(1-p)^{W^{ss}}} \times 2RTT + \frac{(1-p)^{W^{ss}-3}(1-(1-p)^3)}{1-(1-p)^{W^{ss}}} \times RTT \\ &= RTT \times \frac{2-(1-p)^{W^{ss}-3}-(1-p)^{W^{ss}}}{1-(1-p)^{W^{ss}}} \end{aligned} \quad (63)$$

Finally, the expected latency that this loss would incur is:

$$T_{loss} = (1-(1-p)^d)(Q^{ss}E[Z^{TO}] + (1-Q^{ss})E[n_t]) \quad (64)$$

where W^{ss} is:

$$W^{ss} = \min\left(W_m, \frac{E[Y_{init}] + 2}{g^2}\right) \quad (65)$$

3.4 Sending the Rest of the Packets

After the first packet loss, the transmission latency of the rest $(d - E[Y_{init}])$ packets is obtained by using our extended steady-state model as follows:

$$\begin{aligned} T_{rest} &= \frac{d - E[Y_{init}]}{H} \\ &= \frac{dp - (1-(1-p)^d)(1-p)}{p \cdot H} \end{aligned} \quad (66)$$

where H is as given by (49).

⁸ $\lfloor k/2 \rfloor$ is the biggest integer small than $k/2$.

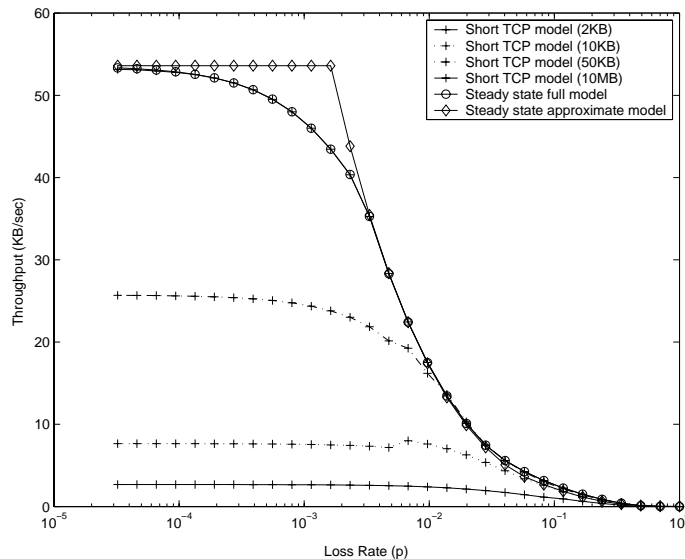


Figure 5. The model for short-lived TCP connections is compared with the steady-state model in terms of throughput versus loss rate for different file sizes. Model parameter values: $RTT = 200ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$, $b = 2$.

3.5 Total Latency

Grouping (51), (57), (63) and (66) together and considering the delay (T_{delay}) caused by the delayed acknowledgment for the first packet (whose mean value is 100ms for the BSD-derived implementations), we now have the total expected latency:

$$T_{latency} = E[T_{twhs}] + E[n]RTT + T_{loss} + T_{rest} + T_{delay} - \frac{RTT}{2} \quad (67)$$

Note that the last term is due to the fact that only half of a round is needed to send the last window of packets.

In Fig. 5, we compare this model for short-lived TCP connections against our steady-state model. Clearly, as the transferred file size increases, the short-lived TCP connection model approaches the steady state model. This is because when a connection has a large amount of data to send, TCP spends most of its time in the steady-state. In addition, as the loss rate increases, the throughput predicted by the short-lived TCP connection model approaches the one predicted by the steady-state model. This is because as the connection loses its packets more frequently, the transient slow-start phase ends quickly and the remaining packets are sent in the steady-state phase.

4. Model Validation through Simulation

We validated our proposed analytical models with simulation experiments. We performed all experiments in *ns-2* [18] using the FullTCP agent. The FullTCP agent is modeled based on the 4.4BSD TCP implementation and can simulate all the important features of TCP Reno. The *ns-2* simulation model used in all experiments is shown in Fig. 6.

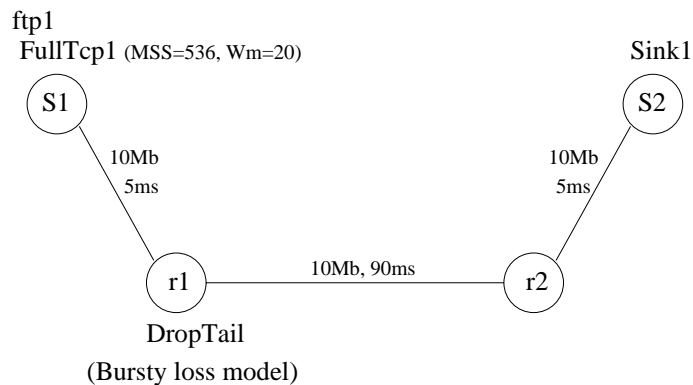


Figure 6. The *ns-2* model that was used to validate our analytical TCP models.

Unlike in [1] where the Bernoulli loss model is used, in our experiments packets were getting lost according to the bursty loss model. Since *ns-2* does not have built-in bursty packet loss model, we added our own `BurstyError Model`, which was derived from the basic `Error Model` class. This `BurstyError Model` drops packets with probability p , which is a Bernoulli trial. After a packet is selected to be dropped with probability p , all the subsequent packets in transit are also dropped. This emulates the DropTail queues behavior under congestion conditions.

We used FTP⁹ as the application for sending a controlled number of packets over a 10Mbps link. The experiments were designed such that the minimum RTT was 200ms.

4.1 The Steady-state Model

Using the same system parameter values that were used to generate Fig. 3, we performed 1000 simulation experiments for each value of p , where p was varied from 0.005 to 0.1. The file size was set to 10MB. Fig. 7 compares the simulation results against the analytical results obtained from our proposed steady-state model (Full: (50), and Approximate: (47)) and the one developed in [3]. Clearly, the results match our expectations. The predicted throughput values at each value of p obtained from our model much closer to the simulation values.

To quantify the accuracy of our model relative to the simulation data, we computed the average error using the following expression taken from [3]):

$$\frac{\sum_{\text{observations}} |Th_{\text{predicted}}(p) - Th_{\text{observed}}(p)| / Th_{\text{observed}}(p)}{\text{Number of observations}}$$

where $Th_{\text{predicted}}$ is the throughput predicted by the models and Th_{observed} is the throughput observed from the simulation experiments. A smaller average error value indicates a better model accuracy. We plotted these average errors against loss rates in Fig. 8. It shows that in most cases the average error is under 8% for our proposed full

⁹FTP is a major Internet application that is used to remotely transfer files.

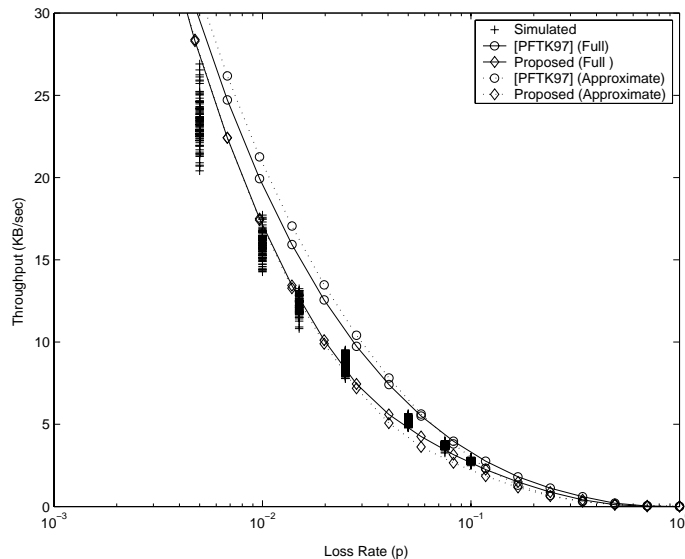


Figure 7. Predicted throughput obtained by our proposed steady-state model and the one developed in [3] are compared against simulation results for the case of $0.005 < p < 0.1$ $RTT = 200ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

model (i.e., (50)) and above 20% for the one from [3]. Approximately, in most cases, our model is 75% more accurate than the model proposed in [3]. This supports our claim that by including the slow-start phase into the steady-state model more accurate predictions can be obtained.

In addition, Fig. 8 depicts the following: the average error in predicted throughput from both analytical models increases as p approaches zero. Let say that $p = 0$ and the initial slow-start threshold is set to the maximum window size. Then, the initial slow-start phase is extended until the congestion window reaches the maximum window size. Since there are no packet losses, TCP never switches to the congestion avoidance phase, but rather continues transmitting packets at its maximum sending rate allowed by the maximum window size. For these cases that $p \approx 0$, our short-lived TCP flow model should be used instead of the steady-state model.

4.2 Short-lived Flows Model: Latency versus Transferred File Size

Fig. 9 shows the relationship between the latency and the transferred file size under no loss conditions. It compares the latency predictions given by our proposed short-lived TCP model ((67)) and the ones obtained by the short-lived TCP models developed in [1] and [2] against the simulation results. Obviously, our model's prediction values match the simulated values better than the values obtained by the other models. Our model resulted in 5.83% average error, compared to 9.40% and 14.53% obtained by the models in [1] and [2], respectively.

Analyzing the results, we also observed that all prediction errors resulted from our model are within $[-RTT/2, RTT/2]$. We believe that this is because our model accounts for the delayed acknowledgment mechanism. Thus,

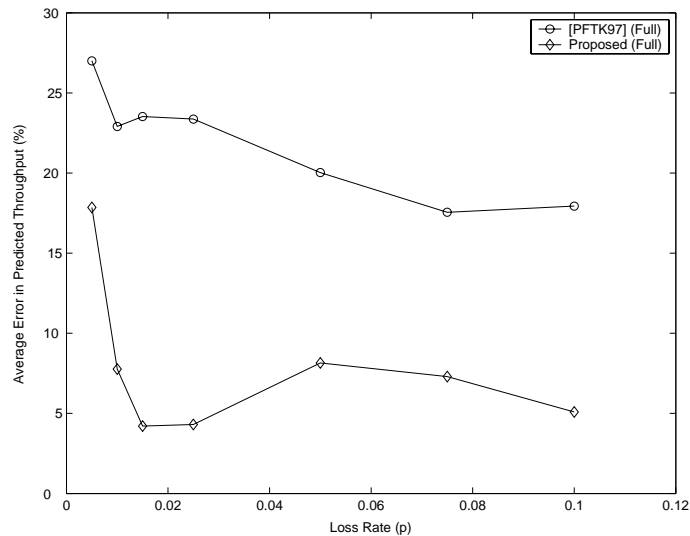


Figure 8. Our proposed steady-state model is compared with the one developed in [3] in terms of the *average error* for the case of $0.005 < p < 0.1$ $RTT = 200ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

TABLE I

OUR SHORT-LIVED TCP CONNECTION MODEL IS COMPARED AGAINST THE ONE PROPOSED IN [1] IN TERMS OF THE AVERAGE ERROR.

Loss Rate	$p = 0$	$3 \times 10^{-3} \sim 10^{-1}$		
		2KB	6KB	11KB
File Size	$0.5 \sim 26KB$			
[CSA00]	9.40%	4.08%	6.43%	8.38%
Proposed	5.83%	0.59%	7.54%	7.64%

for the cases where RTT is small, the prediction errors are insignificant. This is not valid for the other models propose by [1] and [2].

4.3 Short-lived Flows Model: Throughput versus Loss Rate and File Size

Fig. 10, 11 and 12 compares the throughput versus loss rate predictions given by our proposed short-lived TCP model and the one obtained by the short-lived TCP models developed in [1] against the simulation results for the cases of 2KB, 6KB, and 11KB file sizes. Table I compares the two models in terms of the average error.

As can be observed, when the transferred file size is small and the loss rate is low, our model yields more accurate predictions than the model from [1]. Again, this is because our model accounts for the delay acknowledgment mechanism and uses g (golden number) instead of γ (see [1]). However, for large file sizes and loss rates, both models yield similar predictions and in agreement with our steady-state model, as expected.

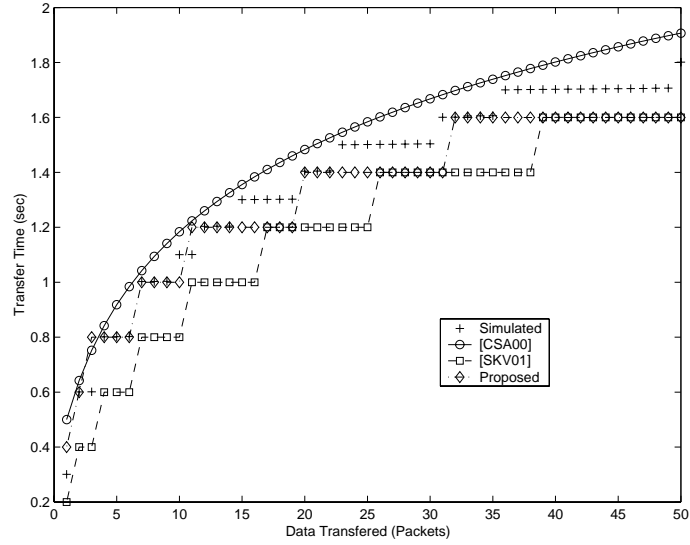


Figure 9. Predicted latency versus transferred file size obtained by our short-lived TCP connection model and the ones developed in [1] and [2] are compared against simulation results for the case of $p = 0$, $RTT = 100ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

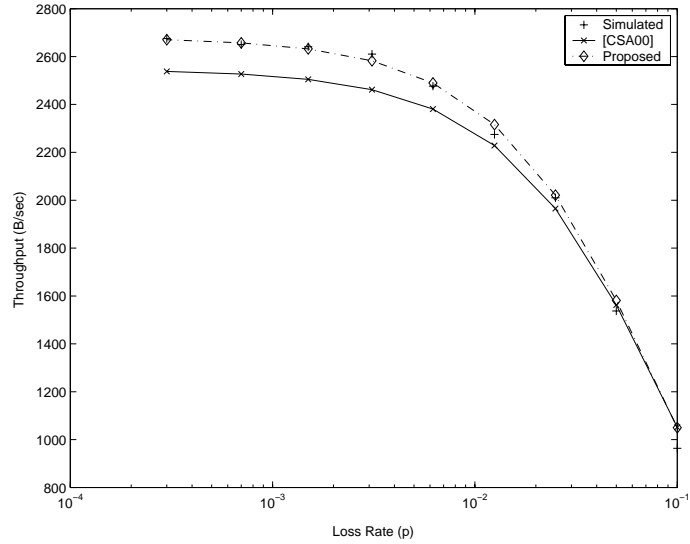


Figure 10. Predicted throughput versus loss rate obtained by our short-lived TCP connection model and the one developed in [1] are compared against simulation results for the case of a 2KB-file-size and $RTT = 100ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

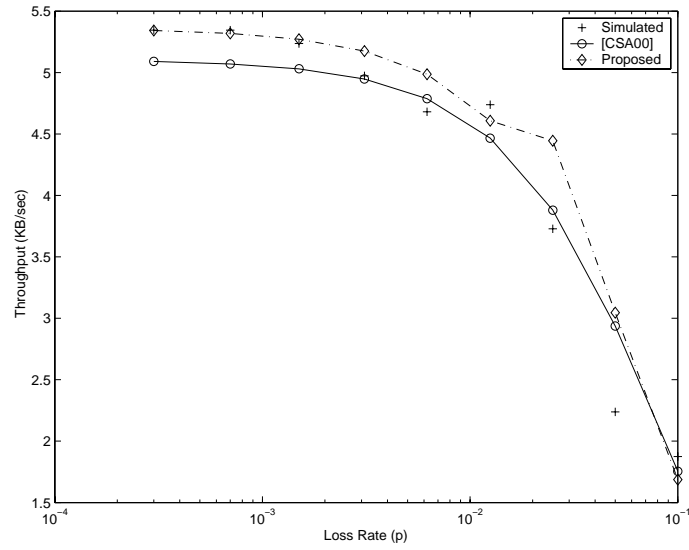


Figure 11. Predicted throughput versus loss rate obtained by our short-lived TCP connection model and the one developed in [1] are compared against simulation results for the case of a 6KB-file-size and $RTT = 100ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

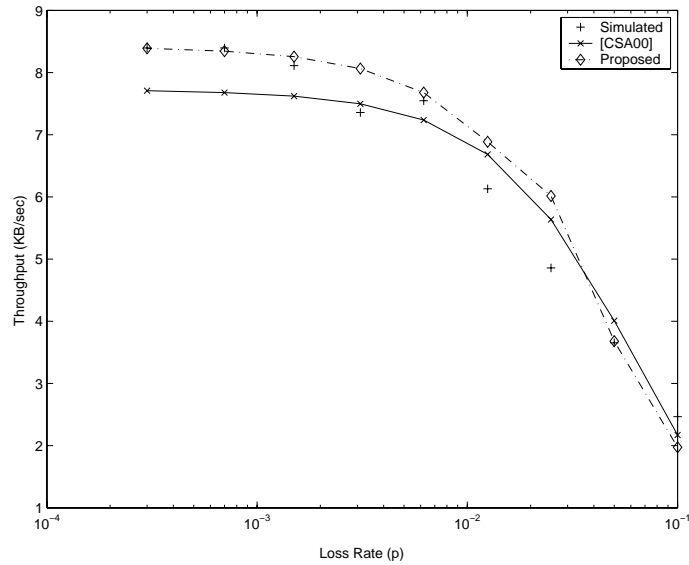


Figure 12. Predicted throughput versus loss rate obtained by our short-lived TCP connection model and the one developed in [1] are compared against simulation results for the case of a 11KB-file-size and $RTT = 100ms$, $MSS = 536bytes$, $w_1 = 1segment$, $T_0 = 1sec$, $W_m = 20segments$.

5. Conclusion

In this paper, we first developed a better and tractable model for the congestion window growth pattern in the slow-start phase. Using this new slow-start phase model, we constructed an accurate model for the short-lived TCP flows and then an extended and more accurate TCP steady-state model. Major improvement in both models was achieved by relaxing key assumptions and enhancing critical approximations that have been made in existing popular models. We validated our models with simulations and compared them against the models developed in [1], [2], [3]. The results support our claim that our models yield more accurate predictions.

References

- [1] N. Cardwell, S. Savage, and T. Anderson, "Modeling TCP latency," in *Proc. IEEE INFOCOM '2000*, Tel Aviv, Israel, Mar. 2000, vol. 3, pp. 1742–1751.
- [2] Biplab Sikdar, Shivkumar Kalyanaraman, and Kenneth S. Vastola, "An integrated model for the latency and steady-state throughput of TCP connections," *Performance Evaluation*, vol. 46, no. 2-3, pp. 139–154, 2001.
- [3] J. Padhye, V. Firoiu, D. F. Towsley, and J. F. Kurose, "Modeling TCP reno performance: A simple model and its empirical validation," *IEEE/ACM Trans. Networking*, vol. 8, no. 2, pp. 133–145, Apr. 2000.
- [4] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, November 1997.
- [5] K. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an internet backbone," in *Proc. of the INET '98*, July 1998.
- [6] J. Heidemann, K. Obraczka, and J. Touch, "Modeling the performance of HTTP over several transport protocols," *IEEE/ACM Trans. Networking*, vol. 5, no. 5, pp. 616–630, Oct. 1997.
- [7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the internet," *IEEE/ACM Transactions on Networking*, August 1999.
- [8] T. J. Ott, T. V. Lakshman, and L. H. Wong, "Sred: Stabilized red," in *Proceedings of the INFOCOM '99*, March 1999.
- [9] J. Blot and T. Turletti, "Experience with rate control mechanisms for packet video in the internet," *Computer Communications Review*, vol. 28, no. 1, January 1998.
- [10] L. Vivisano, L. Rizzo, and J. Crowcroft, "Tcp-like congestion control for layered multicast data transfer," in *Proc. of IEEE INFOCOM '98*, April 1998.
- [11] J. Bolliger, T. Gross, and U. Hengartner, "Bandwidth modeling for network-aware applications," in *In Proceedings of the INFOCOM'99*, March 1999.
- [12] C. Partridge and T. J. Shepard, "TCP/IP performance over satellite links," *IEEE Network*, pp. 44–49, September 1997.
- [13] J. Mahdavi, "TCP performance tuning," April 1997.
- [14] C. R. Cunha, A. Bestavros, and M. E. Crovella, "Characteristics of WWW client-based traces," technical report BU-CS-95-010, Boston University, July 1995.
- [15] M. Mellia, I. Stoica, and H. Zhang, "TCP model for short lived flows," *IEEE Commun. Lett.*, vol. 6, pp. 85–87, Feb. 2002.
- [16] W. R. Stevens, *TCP/IP illustrated*, vol. 1, Addison-Wesley, Reading, MA, 1994.
- [17] V. Jacobson, "Congestion avoidance and control," in *Proc. SIGCOMM '88*, Stanford, CA, 1988, pp. 314–329.

- [18] UCB/LBNL/VINT, “The network simulator ns-2,” 2002.
- [19] M. Mitzenmacher and R. Rajaraman, “Towards more complete models of tcp latency and throughput,” *Journal of Supercomputing*, vol. 20, no. 2, September 2001.
- [20] T. Lakshman and U. Madhow, “The performance of TCP/IP for networks with high bandwidth-delay products and random loss,” *IEEE/ACM Trans. Networking*, vol. 5, no. 3, pp. 336–350, July 1997.
- [21] A. Kumar, “Comparative performance analysis of versions of TCP in a local network with a lossy link,” *IEEE/ACM Trans. Networking*, vol. 6, no. 4, pp. 485–498, 1998.
- [22] J. Mahdavi and S. Floyd, “Tcp-friendly unicast rate-based flow control,” Jan 1997, Note sent to end2end-interest mailing list.
- [23] M. Mathis, J. Semske, J. Mahdavi, and T. Ott, “The macroscopic behavior of the tcp congestion avoidance algorithm,” *Computer Communication Review*, vol. 27, no. 3, July 1997.
- [24] T. Ott, J. Kemperman, and M. Mathis, “Promoting the use of end-to-end congestion control in the internet,” August 1996, <ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps>.
- [25] C. Casetti and M. Meo, “A new approach to model the stationary behavior of TCP connections,” in *Proc. IEEE INFOCOM '2000*, Tel Aviv, Israel, Mar. 2000, pp. 367–375.
- [26] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. H. Katz, and M. Stemm, “TCP behavior of a busy internet server. analysis and improvements,” in *Proc. IEEE INFOCOM '98*, San Francisco, CA, Apr. 1998, pp. 252–262.
- [27] J. C. Hoe, “Start-up dynamics of TCP’s congestion control and avoidance schemes,” M.S. thesis, Massachusetts Institute of Technology, Cambridge, June 1995.

Appendix 1

The expectation of $1/w$

From Taylor formula, we know:

$$f(W) = \sum_{i=0}^{\infty} \frac{f^i(a)}{i!} (W - a)^i \quad (68)$$

Let $f(W)$ and a be $1/W$ and $E[W]$ respectively. We thus have:

$$f^n(W) = (-1)^n n! W^{-(n+1)} \quad (69)$$

Substituting $f^i(a)$ in (68) and making use of (69) and $E[W]$, we get:

$$\begin{aligned} \frac{1}{W} &= \sum_{i=0}^{\infty} \frac{(-1)^i i! E[W]^{-(i+1)}}{i!} (W - E[W])^i \\ &= \sum_{i=0}^{\infty} \frac{(-1)^i (W - E[W])^i}{E[W]^{(i+1)}} \end{aligned} \quad (70)$$

Taking expectation on both sides of (70), results in:

$$\begin{aligned}
E\left[\frac{1}{W}\right] &= E\left[\sum_{i=0}^{\infty} \frac{(-1)^i (W - E[W])^i}{E[W]^{(i+1)}}\right] \\
&= \sum_{i=0}^{\infty} E\left[\frac{(-1)^i (W - E[W])^i}{E[W]^{(i+1)}}\right] \\
&= \sum_{i=0}^{\infty} \frac{(-1)^i E[(W - E[W])^i]}{E[W]^{(i+1)}} \\
&= \frac{1}{E[W]} + \frac{Var(W)}{E[W]^3} + \sum_{i=3}^{\infty} \frac{(-1)^i E[(W - E[W])^i]}{E[W]^{(i+1)}} \\
&\approx \frac{1}{E[W]} + \frac{Var(W)}{E[W]^3} \\
&= \frac{1}{E[W]} \left(1 + \frac{Var(W)}{E[W]^2}\right)
\end{aligned} \tag{71}$$

The approximation holds when $E[W]^{(i+1)} \gg E[(W - E[W])^i]$.

Appendix 2

The variance of W^{TD}

Using similar assumptions as in the previous analysis, from (25), we know:

$$Var[\alpha] = \frac{1-p}{p^2} \tag{72}$$

Thus from (21):

$$Var[Y] = \frac{1-p}{p^2} + Var[W^{TD}] \tag{73}$$

Using (23), we get the auto-correlation at the zero point¹⁰:

$$\begin{aligned}
R_w(0) &= \frac{R_w(0)}{4} + \frac{R_x(0)}{b^2} \\
R_x(0) &= \frac{3b^2}{4} R_w(0)
\end{aligned} \tag{74}$$

And using (30), (31) and (32), we can compute the variance of β as following:

$$\begin{aligned}
Var[\beta] &= R_\beta(0) - E[\beta]^2 \\
&= E[\beta^2] - E[\beta]^2 \\
&= E\left[\sum_{k=0}^{w-1} k^2 p(\beta = k)|w\right] - E[\beta]^2 \\
&= E\left[\sum_{k=0}^{w-1} \frac{k^2 (1-p)^k p}{1 - (1-p)^w} |w\right] - E[\beta]^2 \\
&\approx \frac{2(1-p)^2}{p^2} - (1-p) \left[\sqrt{\frac{8}{3bp}} - 1\right]^2
\end{aligned} \tag{75}$$

¹⁰This is equal to $E[X^2]$.

Using (24), we can also get:

$$\begin{aligned}
Var[Y] &= Var\left[\frac{X_i}{2}\left(\frac{W_{i-1}^{TD}}{2} + W_i^{TD} - 1\right)\right] + Var[\beta] \\
&= E\left[\left(\frac{X_i}{2}\right)^2\right]E\left[\left(\frac{W_{i-1}^{TD}}{2} + W_i^{TD} - 1\right)^2\right] - \left(E\left[\frac{X_i}{2}\left(\frac{W_{i-1}^{TD}}{2} + W_i^{TD} - 1\right)\right]\right)^2 + Var[\beta] \\
&= \frac{R_x(0)}{4} \frac{5R_w(0)}{4} - E\left[\frac{X}{2}\right]^2 E\left[\frac{W_{i-1}^{TD}}{2} + W_i^{TD} - 1\right] + Var[\beta] \\
&= \frac{\frac{3b^2}{4}R_w(0)}{4} \frac{5R_w(0)}{4} - \frac{E[X]^2}{4} \left[\frac{3}{2}E[W^{TD}] - 1\right]^2 + Var[\beta] \\
&= \frac{15b^2}{64} [Var[W^{TD}] + E[W^{TD}]^2]^2 - \frac{E[X]^2}{4} \left[\frac{3}{2}E[W^{TD}] - 1\right]^2 + Var[\beta] \\
&\approx \frac{15b^2}{64} [Var[W^{TD}] + \frac{8}{3bp}]^2 - \frac{1}{4} \left(\frac{b}{2}\sqrt{\frac{8}{3bp}} + b\right)^2 \left(\frac{3}{2}\sqrt{\frac{8}{3bp}} - 1\right)^2 \\
&\quad + \frac{2(1-p)^2}{p^2} - (1-p)\left[\sqrt{\frac{8}{3bp}} - 1\right]^2
\end{aligned} \tag{76}$$

Combining (76) and (73), we obtain the final equation:

$$\begin{aligned}
\frac{1-p}{p^2} + Var[W^{TD}] &= \frac{15b^2}{64} [Var[W^{TD}] + \frac{8}{3bp}]^2 - \frac{1}{4} \left(\frac{b}{2}\sqrt{\frac{8}{3bp}} + b\right)^2 \left(\frac{3}{2}\sqrt{\frac{8}{3bp}} - 1\right)^2 \\
&\quad + \frac{2(1-p)^2}{p^2} - (1-p)\left[\sqrt{\frac{8}{3bp}} - 1\right]^2
\end{aligned} \tag{77}$$

Solving (77), we obtain the variance of W^{TD} as follows:

$$Var[W^{TD}]_{p \rightarrow 0} \approx \frac{8(\sqrt{3} - 1)}{3bp} \tag{78}$$