Modelling TCP Reno with Spurious Timeouts in Wireless Mobile Environments

Shaojian Fu and **Mohammed Atiquzzaman**
Telecommunications and Networks Research Lab
School of Computer Science
University of Oklahoma,
Norman, OK 73019-6151, USA.
Email: {sfu,atiq}@ou.edu

***Abstract*— TCP has been found to perform poorly in the presence of spurious timeouts (ST) caused by delay spikes which are especially more frequent in today's wireless mobile networks than in traditional wired network. Because STs do not frequently occur in wired networks, and are generally considered to represent a transient state, previous research did not consider the effects of ST on the steady state performance of TCP. However, ST is more frequent in wireless mobile environments, and must be considered explicitly to accurately model the steady state sending rate and throughput of TCP. In this paper, we propose an analytical model for the sending rate and throughput of TCP Reno as a function of packet error rate and characteristics of spurious timeouts. The accuracy of the proposed model has been validated against simulation results. The accuracy of the model has also been compared with previous models, and has been found to be more accurate than previous models in the presence of spurious timeouts.**

I. Introduction

TCP is the dominant transport layer protocol in the IP protocol suite, which carries most of the Internet traffic such as Web browsing, bulk file transfer and Telnet. TCP was initially designed for wired networks, and hence performs poorly in the presence of delay spikes which are especially more frequent in today's wireless mobile networks than in traditional wired network [1], [2], [3], [4], [5]. A sudden increase of the instantaneous round trip time (RTT) beyond the sender's retransmission timeout value (RTO) causes retransmission ambiguity [6], [7], resulting in *Spurious Timeouts*[1] (ST) and *Spurious Fast Retransmissions*[2] (SFR) which produce serious end-to-end TCP performance penalty [4], [7]. Causes of delay spikes in a wireless mobile environment include [1]:

- The *handoff* of a mobile host between cells requires the base station to do channel allocation before data can be transmitted from the mobile host. This causes segments at the mobile host to be queued until the completion of the channel allocation, giving rise to sudden extra delay (in addition to the normal RTT).
- The *physical disconnection* of the wireless link during a hard handoff will also result in a sudden increase of the RTT.
- A Radio Link Control (RLC) layer between the LLC and MAC layers, to carry out retransmission at the link layer (for error recovery) in wireless mobile networks (such as GPRS and CDMA2000), may result in delay spikes due to repeated retransmission attempts during link outages and periods of high link errors.
- Higher-priority traffic, such as circuit-switched voice, can preempt (block) the data traffic temporarily. The duration of this blocking may be very long as compared to TCP's RTT estimate.

A number of mechanisms have been proposed in the literature to improve the performance of TCP in the presence of ST or SFR. For example, Ludwig et.al. [7] uses the TCP timestamp option in the Eifel algorithm to resolve retransmission ambiguity; a retransmit flag, called "rtx", is added in the TCP header [8] also to resolve retransmission ambiguity; Blanton et.al. proposed using the TCP D-SACK option to detect spurious retransmissions [9]; Gurtov et.al. have suggested conservative management of TCP's retransmission timer [5]. However, there is no analytical framework to compare the performance and effectiveness of these improvements.

During recent years, several papers have reported analytical models to predict the throughput of TCP during *bulk file transfers* [10], [11], [12], [13]. Lakshman et. al. [10] analyzed the performance of one or more TCP connections that share a bottleneck link in a WAN environment for large bandwidth-RTT product connections. They considered slow start and congestion avoidance, but not timeouts. The model by Mathis et. al. [11] did not consider retransmission timeouts, and hence cannot be applied to non-random losses caused by drop-tail queues. Kumar et. al. [12] modelled and compared several TCP versions (TCP Tahoe, Reno, NewReno), where they considered timeouts, and are therefore, more appropriate for the analysis of local wireless networks. The model proposed by Padhye et. al. [13] improves the one in [11] by considering the effect of timeouts and limited receiver window; this model is more accurate than previous models for correlated losses and a wide range of packet loss rates. However, *none of the above models considers the effect of ST on the steady state throughput of TCP*. This

[1] **Spurious timeout** is defined as a timeout which would not have happened if the sender waited long enough. It is a timeout resulting in retransmission due to a segment being delayed (but NOT lost) beyond RTO [7].

[2] **Spurious fast retransmission** occurs when segments get re-ordered beyond the DUPACK-threshold in the network before reaching the receiver [7], i.e. the reordering length is greater than the DUPACK threshold (three for TCP).

may be due to the fact that (a) ST does not occur frequently in wired networks, and (b) ST is considered to be a transient state in a wired network, and thus cannot produce much impact on the steady state performance of TCP. However, in wireless mobile environments, STs are more frequent and must be considered explicitly in order to accurately model the steady state throughput of TCP. The *objective* of this paper is to develop an analytical model to enable us to understand and predict the performance of TCP during STs. This paper differs from previous research in the fact that the model proposed in the paper *explicitly takes into account the effect of ST* on the steady state performance of TCP. The proposed model of TCP will, therefore, enhance further development and evaluation of transport protocols in the area of wireless and mobile networks.

Our proposed model is based on a stochastic analysis of the steady state sending rate and throughput of TCP Reno as a function of packet error rate, interval between long delays, and duration of long delays. The model by Padhye et. al. [13] characterizes both the fast retransmit and the time out behavior of TCP Reno, and can accurately predict TCP performance over a wide range of loss rates. We therefore, use the result of Padhye et. al. as a basis of our work. From this point on, we will use ”PFTK” (the initials of the authors) to refer to this model. The main *contributions* of this paper can be summarized as follows:

- we developed an analytical model of TCP performance by *explicitly considering ST effect*;
- we compared the effectiveness of our proposed analytical model with that of PFTK model, and found that our proposed model is much more accurate for estimating TCP performance in the presence of frequent long delays.
- the model has been validated against simulation results;

The model proposed in this paper is expected to have significant *impact* on future transport layer research as follows: (1) There is always a fundamental trade off between the rapidness of detection of true losses versus the risk of unnecessary retransmissions when designing an *RTO* calculation algorithm or setting related parameters. For example, the TCP parameter RTO_{min}, the lower bound of the *RTO* value, has a significant impact on the effectiveness of the *RTO* estimator [14]. There is no existing method to optimally set RTO_{min}, and the current practice is to set it to twice the clock granularity. Since our proposed model considers the effect of ST, it can assist in *determining an appropriate value of* RTO_{min}. (2) There is an increasing research interest to study the interaction between TCP and lower layer protocols in wireless environments [4], [15], [2]. The settings of lower layer protocols, such as handoff schemes in Mobile IP and retransmission schemes at the link layer, have a nontrivial impact on the frequency of TCP spurious timeouts. The model proposed in this paper can facilitate the *fine-tuning* of these settings in a more coordinated fashion in order to achieve an optimal performance. (3) Our proposed model can provide a framework for evaluating the impact of modifications proposed to TCP to alleviate the effects of ST, and to compare the performance of the modified TCP with previous versions of TCP. This will improve the current situation where the modifications are mainly tested by simulations, and hence may not be able to cover all possible network scenarios.

The rest of the paper is organized as follows. In Sec. II, the assumptions for developing our model are discussed, followed by the model in Sec. III. We then validate the accuracy of the proposed model against simulations using the *ns-2* network simulator in Sec. IV. We present the accuracy of our proposed model, and compare the performance of TCP under ST obtained from our model with a previous TCP model in Sec. V. Finally, we present concluding remarks in Sec. VI.

II. Modelling Assumptions

The assumptions we have made for developing our analytical model of TCP with STs are described below.

- To isolate only the impact of long delays and segment losses on TCP, we assume that the sending rate is not limited by the advertised receiver window, and the sender always has sufficient data to send. This assumption is satisfied easily by setting a large buffer size at the receiver. Possible extension of our model to limited receiver window is outlined in Sec. VI.
- Segment losses in a round are independent from losses in other rounds. Here, a ”round” is defined as the time between the sending of the first segment in a window to the receipt of the corresponding acknowledgment (ACK). We assume that all other segments which were sent after the first lost segment in a specific round are also lost (same assumption as in PFTK [13]). This loss model is similar to the 2-state Markov chain approximation of the loss behavior observed in previous research [16].
- The time required to send a window of data is smaller than an RTT. This assumption is justified by the observations of simulation trace plots of Fall et. al. performed in [17], and was also used in the PFTK model [13].
- Our goal is to model the effect of delay spikes caused by mobile handoffs, link layer retransmissions, and packet rerouting on the performance of TCP. We therefore, do not explicitly consider the fluctuation of round trip time measurements caused by queueing delays. We assume that, in the absence of delay spikes, these measurements compose a stationary random process with an expected value of RTT.
- Since our main concern in this paper is to model the effect of Spurious Timeouts on TCP, we assume that *BugFix*, proposed in RFC2582 [18], is enabled to prevent Spurious Fast Retransmission. However, note that if *BugFix* is not enabled, a Spurious Fast Retransmission usually follows a Spurious Timeout. This is because the spuriously retransmitted segments produce a sequence of duplicate acknowledgements at the receiver [7].

III. Analytical Model

In this section, our objective is to develop an analytical model for the ssending rate and throughput of TCP as a function of packet error rates and long delays. First, we will determine the sending rate (Sec. III-D), and then the throughput (Sec. III-E) will be obtained by subtracting the lost and spuriously retransmitted segments from the sending rate. The model for the sending rate is developed by analyzing the dynamics of the sender's window around a long delay (Sec. III-B). We describe below the notations that are used in our model.

A. Notations

The notations used in this paper are given below. Our model is based on the modelling approach used by PFTK model. For the sake of consistency and ease of understanding by the reader, we therefore use many of the terminology and notations used in [13].

- I interval between long delays.
- D duration of the long delay.
- p packet error rate.
- b number of segments acknowledged by one ACK segment. $b = 2$ when delayed acknowledgment is used at the receiver.
- RTT expected value of round trip time when there is no long delay.
- T_0 converged RTO value as defined in Sec. III-B.
- W TCP sender window size.
- B, T steady state sending rate and throughput, respectively, of TCP connection.
- TDP triple duplicate period, i.e. the time between two successive triple duplicate loss indications.
- LDP long delay period, which consists of one TDP, one long delay, one slow start, and a second TDP (see Fig. 1).
- NP "normal period", which consists of n instances of TDP and one instance of timeout period (see Fig. 3).
- n number of TDPs in one NP.
- LDC long delay cycle, which consists of m NPs and one LDP (see Fig. 3).
- m number of NPs in one LDC.
- Z^{TDP}, Z^{NP}, Z^{LDP} duration of one TDP, NP, and LDP, respectively, note that A and S is used in [13] instead of Z^{TDP} and Z^{NP}.
- Z^{TD} duration of n instances of TDPs in one NP (see Fig. 3).
- Z^{TO} duration of the timeout period in one NP (see Fig. 3).
- Y number of segments sent from the sender during one TDP.
- M_r number of segments sent during $r_{th}\,NP$, $r = 1, 2 \cdots m$.
- R number of retransmitted segments during the timeout period in one NP.
- R_D number of retransmitted segments during D.
- SST value of slow start threshold at the end of a long delay D.
- v the number of rounds needed to complete the slow start stage after a long delay.
- K number of segments sent during the slow start stage in an LDP.
- U, G number of segments sent during one LDP and one LDC respectively.

B. Dynamics of sender window around a long delay

Before we develop a model for the sender's sending rate in the next section, we analyze the dynamics of the sender's window around a long delay in this section. Fig. 1 shows the evolution of sender's window size as represented by the number of segments that can be sent. At each round the window is increased by $1/b$. After X_i rounds, the long delay (D) begins, when some of the segments in the X_i-th round are delayed (segments marked "d"). Since the long delay is of a much larger timescale than a round, any extra segments that were sent in round X_{i+1}, corresponding to the ACKs of successfully delivered segments of round X_i, are also delayed. After T_0 seconds, which is the converged value of *RTO* when the round trip time is stable for a relatively long period of time, the sender will timeout and reduce the window to one and retransmit the first delayed segment. If it is not acknowledged within $2T_0$, the sender will retransmit it again, and so on. The number of retransmissions during the long delay is denoted by R_D; all these retransmitted segments are also delayed. Eventually, when the ACK for the first delayed segments comes back after the long delay has cleared, the sender will enter slow start and spuriously retransmit all the delayed segments (segments marked "s"). The sender will exit slow start when the window hits the slow start threshold (denoted SST).

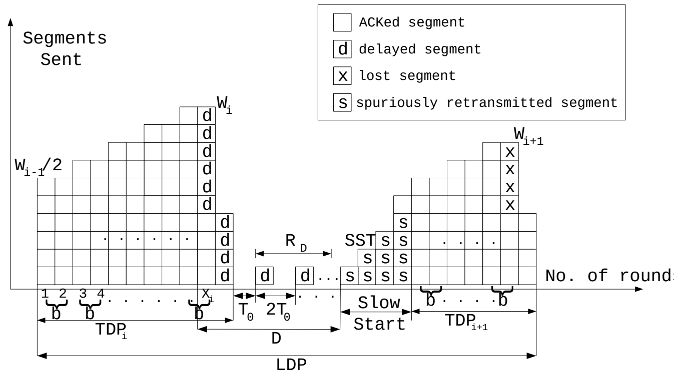
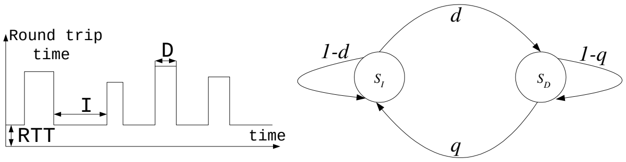


Fig. 1. Segments sent during one Long Delay Period (LDP).

TCP Reno starts fast retransmit after receiving three duplicate ACKs, which are called *triple-duplicate loss indications*. Triple Duplicate Period (TDP) is defined in [13] as a period between two successive triple-duplicate loss indications. We define a Long Delay Period (LDP) as consisting of two consecutive TDPs, one long delay, and one slow start as shown in Fig. 1. Note that even though the first period, labelled with TDP_i in the figure, does not end with a triple duplicate loss indication, the number of segments sent and the duration of TDP_i is the same as other TDPs, so we just use TDP for convenience. The sender's window was W_{i-1} at the end of TDP_{i-1}; after the fast retransmit, it has been reduced to $W_{i-1}/2$, which is the sender's window at the start of TDP_i.

C. Statistical modelling of the long delay pattern

In this section, we develop a model for the long delay pattern, which will be used to model the sending rate in Sec. III-D. The round trip times as measured by the sender in the presence of long delays is shown in Fig. 2(a). We use a two-state Markov chain to model the start and end of a long delay as shown in Fig. 2(b). The two states are: Interval between long delays (S_I)



(a) (b)

Fig. 2. (a) Variation of RTT showing four long delays, and (b) model of long delays.

and long Delay (S_D). Here, we assume that the length of the S_I and S_D states are both exponentially distributed, with d and q being the transition probabilities from state S_I to state S_D and state S_D to state S_I, respectively. By solving the Markov chain in Fig. 2(b), the relationship between I and D can be expressed as:

$$E(D) = \frac{d\,E(I)}{d+q} \tag{1}$$

Given a model for the lower layer events (such as link layer retransmission, mobile handoff, etc. [19]) that cause long delays, we can obtain the values of $E(I)$, $E(D)$, d, and q to be used in Eqn. (1).

D. Modelling the TCP sending rate

In this section, we consider the sending rate of TCP as a function of p, I, and D. The average sending rate of TCP can be calculated as

$$B(p,I,D) = \frac{E(G)}{mE\left(Z^{NP}\right) + E\left(Z^{LDP}\right)} \tag{2}$$

where, the numerator denotes the number of segments sent during one Long Delay Cycle (LDC) (to be derived in Sec. III-D.2) and the denominator is the duration of an LDC. We first look at the macroscopic behavior of one LDP in Sec. III-D.1, which will then be used to determine the number of segments sent and the duration of an LDC.

1) Analysis of a Long Delay Period (LDP): The total number of segments sent during one LDP is the sum of segments sent during two TDP periods, the timeout period, and the slow start stage (Fig. 1):

$$E(U) = 2E(Y) + E(R_D) + E(K) \tag{3}$$

The duration of LDP can be written as the sum of the time duration of the two TDP periods, the long delay, and one slow start stage, minus the overlapping area ($2RTT$) between D and TDP_i:

$$\begin{aligned} E(Z^{LDP}) &= 2E(Z^{TDP}) + E(D) + v\,RTT - 2RTT \\ &= 2E(Z^{TDP}) + E(D) + (v-2)\,RTT \end{aligned} \tag{4}$$

$E(Y)$ and $E(Z^{TDP})$ in Eqns. (3) and (4) can be determined from Eqns. (6) and (7), which can be obtained from the PFTK model [13] as follows:

$$E(W) = \sqrt{\frac{8}{3bp}} \tag{5}$$

$$E(Y) = \frac{1-p}{p} + E(W) = \frac{1-p}{p} + \sqrt{\frac{8}{3bp}} \tag{6}$$

$$E(Z^{TDP}) = \left(\frac{b}{2}E(W) + 1\right) RTT \tag{7}$$

Next, we derive the three unknown variables ($E(R_D)$, v, and $E(K)$) in Eqns. (3) and (4).

Determination of E(R_D): Since we assume that D is exponentially distributed (see Sec. III-C) with mean $E(D)$, if the sender experiences a long delay of D, the probability that there is one timeout is:

$$\begin{aligned} Pr\left(T_0 < D \le 2T_0\right) &= Pr(D \le 2T_0) - Pr(D \le T_0) \\ &= e^{-\frac{T_0}{E(D)}} - e^{-\frac{2T_0}{E(D)}} \end{aligned} \tag{8}$$

The probability that there are two or more timeouts is:

$$Pr\left(D > 2T_0\right) = e^{-\frac{2T_0}{E(D)}} \tag{9}$$

Because the sender sends out a segment when a timeout occurs, the number of segments sent during D is the same as the number of timeouts. since the sender can backoff a maximum of 6 times to get a RTO of $64T_0$, the number of segments sent can be expressed as:

$$\begin{aligned} E(R_D) &= 1Pr\left(T_0 < D \le 2T_0\right) + 2Pr\left(2T_0 < D \le 3T_0\right) + \\ &\quad \cdots + 6Pr\left(32T_0 < D \le 64T_0\right) \\ &= \sum_{j=0}^{5}\left(e^{-\frac{2^j T_0}{E(D)}} - e^{-\frac{64T_0}{E(D)}}\right) \end{aligned} \tag{10}$$

Determination of v: After the long delay, the SST value will be $\max(W_i/2, 2)$ if there is only one timeout during D, otherwise it will be two for two or more timeouts. Therefore, the expected value of SST after the long delay is:

$$E(SST) = \max(W_i/2,2)\left(e^{-\frac{T_0}{E(D)}} - e^{-\frac{2T_0}{E(D)}}\right) + 2e^{-\frac{2T_0}{E(D)}} \tag{11}$$

During the slow start, if the receiver adopts delayed acknowledgment, the sender's congestion window will grow by half of the window size in the previous round according to the following rule:

$$cwnd_{j+1} = cwnd_j + \left\lceil\left(\frac{cwnd_j}{2}\right)\right\rceil \quad \text{with } cwnd_1 = 1, \quad j = 1,2,3\cdots \tag{12}$$

which can be approximated as:

$$cwnd_j = \left(\frac{3}{2}\right)^j \quad j = 1,2,3\cdots \tag{13}$$

End of the slow start stage at $E(SST)$ after v rounds implies that $cwnd_v = E(SST)$; the number of rounds needed to complete this stage is approximately expressed as:

$$v = \left\lceil \log_{3/2}^{(E(SST))} \right\rceil \approx \lceil 1.71\log\left(E\left(SST\right)\right)\rceil \tag{14}$$

Determination of E(K): The number of segments sent in each round of the slow start stage in Fig. 1 is given in Eqn. (13). So the number of segments sent during slow start can be approximated by the sum of the segments sent during these v rounds:

$$E(K) = \sum_{j=1}^{v}\left(\frac{3}{2}\right)^j \approx 3\left(\frac{3}{2}\right)^{\lceil 1.71\log(E(SST))\rceil} - 3 \tag{15}$$

By substituting $E(R_D)$, v, and $E(K)$ from Eqns. (10), (14), and (15) into Eqns. (3) and (4), we can obtain the number of segments sent and the duration of one LDP.

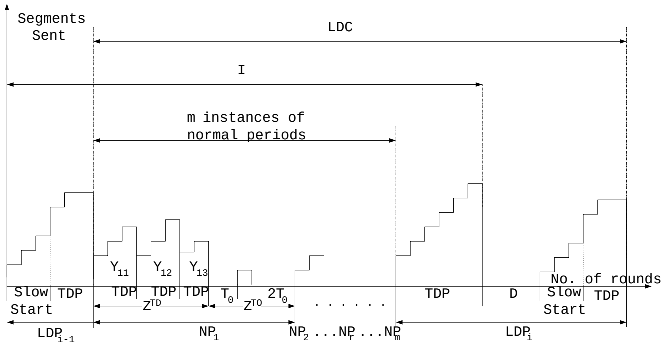


Fig. 3. Sender window evolution during one Long Delay Cycle (LDC).

2) Analysis of one Long Delay Cycle (LDC): In Eqn. (2), Z^{NP} can be obtained from [13] as given in Eqn. (16), $E(Z^{LDP})$ has already been developed in Sec. III-D.1, and $E(G)$ depends on m, $E(U)$, $E(M_r)$, $E(R)$. $E(M_r)$ and $E(R)$ can be obtained from [13] as given in Eqns. (16) and (17) .

$$\begin{aligned} E(Z^{NP}) &= E(n)E(Z^{TDP}) + E\left(Z^{TO}\right) \\ &= \left(\frac{b}{2}E(W)+1\right)E(n)RTT + T_0\frac{f_p}{1-p} \qquad (16) \\ \text{where } f_p &= 1+\sum_{i=1}^{6}2^{i-1}p^i \end{aligned}$$

$$E(M_r) = E(n)E(Y) + E(R) = \frac{\frac{1-p}{p} + \sqrt{\frac{8}{3bp}}}{\min\left(1, 3\sqrt{\frac{3bp}{8}}\right)} + \frac{1}{1-p} \qquad (17)$$

$E(n)$ and $E(R)$ in Eqns. (16) and (17) can be determined from Eqns. (18) and (19), which can be obtained from the PFTK model [13] as given below.

$$E(n) = \frac{1}{\min\left(1, \frac{3}{E(W)}\right)} \qquad (18)$$

$$E(R) = \frac{1}{1-p} \qquad (19)$$

$E(U)$ has already been developed in Sec. III-D.1, which leaves us with only determining m.

We define another term, called *LDC* (as shown in Fig. 3), which starts with the end of the previous *LDP*. An *LDC* consists of several instances of ”normal periods” (*NP*) at the beginning and an *LDP* at the end. Here, the ”normal period” denotes the time interval with no long delays, which is equal to the sum of Z^{TD} and Z^{TO}; values of Z^{TD} and Z^{TO} are obtained from [13] as given in Eqn. (16).

Referring to Fig. 3, the interval between long delays (I) consists of a slow start phase following the previous long delay, m instances of NP and a TDP. We can calculate m as:

$$m = \frac{E(I) - 2E\left(Z^{TDP}\right) - v\cdot RTT}{E\left(Z^{NP}\right)} \qquad (20)$$

Since one LDC consists of m instances of NP and ends with one LDP, the duration of one LDC can be obtained as: $mE\left(Z^{NP}\right) + E\left(Z^{LDP}\right)$

The total number of segments sent during one LDC is the sum of segments sent during m instances of NP period and an LDP period:

$$E(G) = \sum_{r=1}^{m} M_r + E(U) = mE(M_r) + E(U) \qquad (21)$$

By substituting $E(G)$ from Eqn. (21) into Eqn. (2), we can obtain the steady state sending rate of the TCP sender.

E. Modelling TCP throughput

We determine the TCP throughput by subtracting the spuriously retransmitted and lost segments from the sending rate (derived in Sec. III-D). Referring to Fig. 1, the delayed segments in the X_i and X_{i+1}-th rounds of the first TDP are subsequently spuriously retransmitted during the slow start stage. Therefore, we need to subtract one window of segments ($E(W)$) from $E(Y)$.

$$E(Y_1') = E(Y) - E(W) = \frac{1-p}{p} \qquad (22)$$

In the second TDP of the LDP period, the lost segments (marked ”x”) need to be subtracted from the sending rate, i.e. on the average, we need to subtract $\frac{E(W)}{2}$.

$$E(Y_2') = E(Y) - \frac{E(W)}{2} = \frac{1-p}{p} + \frac{E(W)}{2} \qquad (23)$$

Because the segments retransmitted during the timeout period are discarded by the receiver, we can replace $E(R)$ in Eqn. (17) with $E(R') = 1$. Similarly, we have $E(R_D') = 1$. Replacing $E(Y)$, $E(R)$ and $E(R_D)$ in Eqns. (3) and (17) with $E(Y')$, $E(R')$ and $E(R_D')$, we obtain:

$$\begin{aligned} E(U') &= E(Y_1') + E(R_D') + E(K) + E(Y_2') \qquad (24) \\ E(M_r') &= E(n)E(Y_2') + E(R') \qquad (25) \end{aligned}$$

Therefore, the average TCP throughput during one LDC can be calculated as the total number of segments delivered to the receiver divided by the duration of one LDC. The segments delivered can be obtained by replacing $E(U)$ and $E(M_r)$ in Eqn. (21) with $E(U')$ and $E(M_r')$. Although we subtract the spuriously retransmitted and lost segments from the total number of segments received at the receiver, the duration of an LDC remains unchanged. We can write the throughput of the TCP connection as:

$$T(p, I, D) = \frac{mE(M_r') + E(U')}{mE(Z^{NP}) + E(Z^{LDP})} \qquad (26)$$

IV. Simulation setup

In order to validate the accuracy of our model presented in Sec. III, we compare the results obtained from the analytical model against simulation results obtained from the *ns-2* [20] network simulator in Sec. V. The long delays are simulated using an *ns-2* module called ”hiccup” [21] which holds all the arriving segments for time D, before releasing them into the link. This ”hiccup” module enables us to accurately control the start and end of long sudden delays. The simulation topology is shown in Fig. 4, where a TCP Reno sender sends FTP traffic to a destination via a link equipped with a hiccup module and an error module. The queue size of the link is set to be large enough (800 packets) to remove the possibility of packet drops

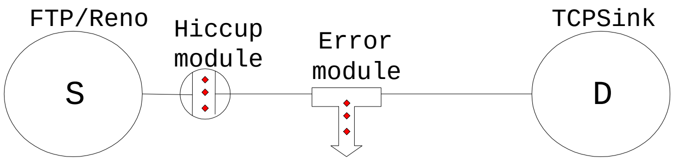


Fig. 4. *ns-2* simulation topology with hiccup and error modules.

TABLE I
SIMULATION PARAMETERS FOR THE TOPOLOGY OF FIG. 4.

Protocol	TCP Reno
Header size	40 bytes
Payload size	536 bytes
rwnd limit	800 segments
Initial *cwnd*	1 segment
Initial *ssthresh*	800 segments
link bandwidth	300Mbps
link propagation delay	50 - 200 ms
link loss rate	0.001 - 0.5
link buffer size limit	800 packets

caused by link queue overflows. The packet error rate can thus be accurately controlled by the error module.

We insert the hiccup module to simulate different delay patterns ($E(I)$, $E(D)$) (see Sec. III-C), and also simulate the packet error rate p using a 2-state Markov error module. We measure the sending rate (B) and the throughput (T) of TCP, and compare them with the PFTK model and our proposed analytical model in Sec. V. Values of relevant simulation parameters are summarized in Table I. Note that we set the *rwnd* limit to a large value of 800 segments to avoid any effect of the advertised receiver window on the sending rate and throughput. We also set the link bandwidth to a large value of 300Mbps to simulate the sender behavior of probing for available network bandwidth. The sending rate and throughput are, therefore, only limited by the values of p, I, and D.

We vary the interval between the long delays (I) with an expected value ranging from 30 to 240 seconds, and long delay duration (D) with an expected value ranging from 2 to 12 seconds. For each p, I, D, RTT combination, we run the simulation for 100 times, with 300 seconds for each run to make the simulation results statistically trustable.

V. RESULTS

In this section, we evaluate the effectiveness of our proposed model by comparing the sending rate and throughput predicted by our model and the PFTK model against the values obtained from simulation. To find out the sensitivity of these two models to different values of $E(D)/E(I)$ ratio, RTT, and p, we also compared the mean square estimation error and the 95% confidence interval error range of the two models for these parameters.

A. Comparison of sending rate estimation

We compare the predicted sending rate from our proposed model and PFTK model against simulation results. Figs. 5 and 6 show the scenarios where RTT = 200ms, $E(I)$ = 30 and 240 seconds, and $E(D)$ ranges from 6 to 12 seconds. Fig. 5 shows that the proposed model can predict the sending rate more accurately than the PFTK model. It is also shown that when $E(D)$ increases, as expected, the gap between the PFTK model and the simulation result increases, but the proposed model accommodates the increase of $E(D)$ well. When $E(I)$ increases to 240 (Fig. 6), implying that the long delays are much more sparse than the $I = 30$ scenario, the estimations from the proposed model and the PFTK model are rather close. We repeated the above experiments for RTT = 400ms, and obtained similar results as shown in Figs. 7 and 8.

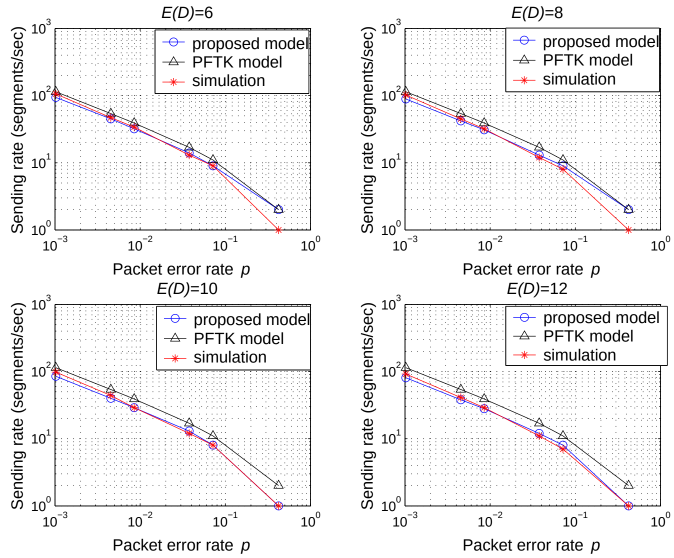


Fig. 5. Sending rate estimation for RTT=200ms and $E(I)$=30 sec.

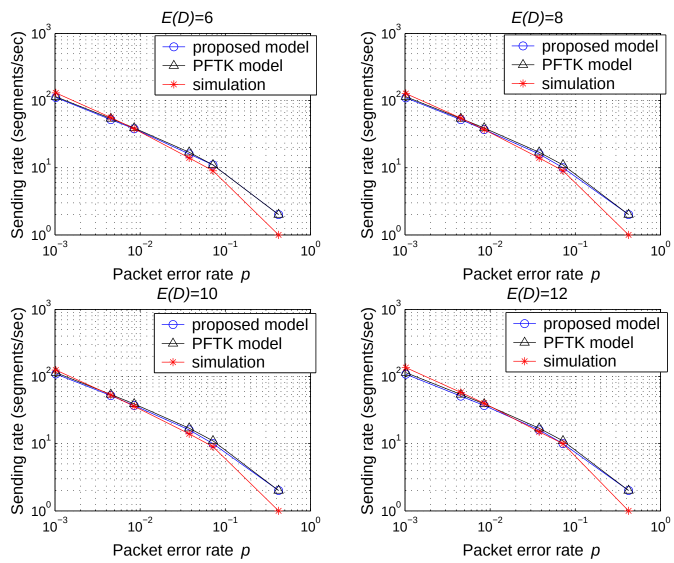


Fig. 6. Sending rate estimation for RTT=200ms and $E(I)$=240 sec.

B. Comparison of throughput estimation

Next, we compare the predicted throughput from the proposed model and the PFTK model against the values obtained

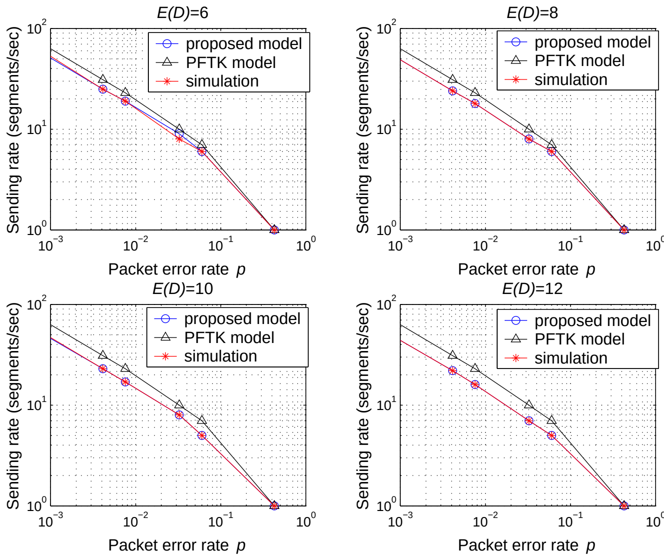


Fig. 7. Sending rate estimation for RTT=400ms and $E(I)$=30 sec.

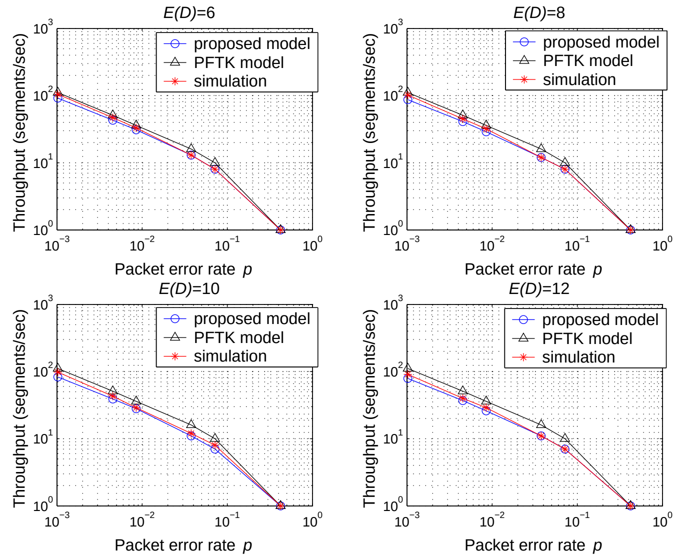


Fig. 9. Throughput estimation for RTT=200ms and $E(I)$=30 sec.

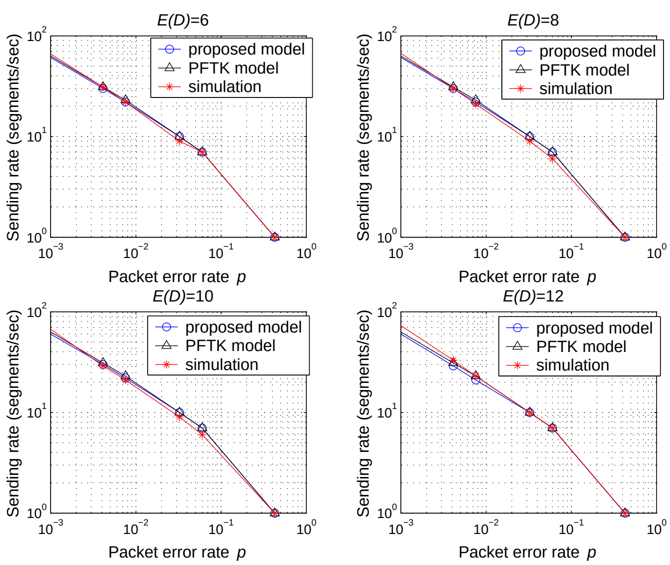


Fig. 8. Sending rate estimation for RTT=400ms and $E(I)$=240 sec.

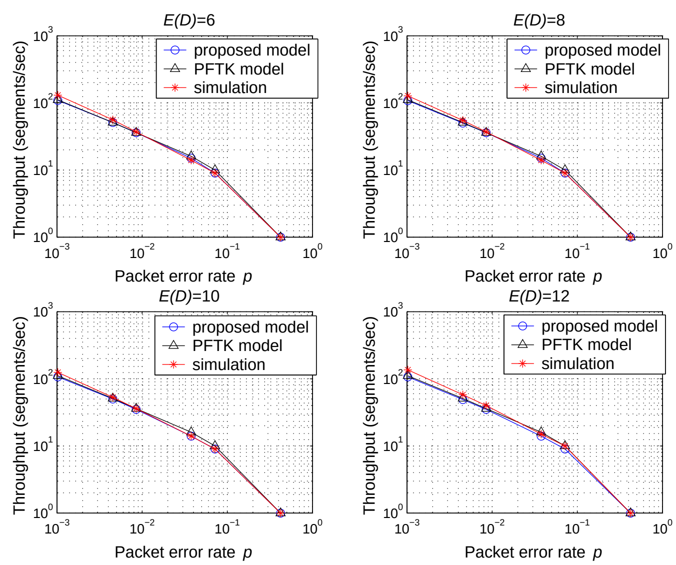


Fig. 10. Throughput estimation for RTT=200ms and $E(I)$=240 sec.

from simulation. Figs. 9 and 10 show the results for RTT = 200ms, $E(I)$ = 30 and 240 seconds, and $E(D)$ ranging from 6 to 12 seconds. Fig. 9 shows that the proposed model can also predict the actual throughput more accurately than the PFTK model. It is also shown that when $E(D)$ increases, the difference between the PFTK model and the simulation result increases, but the proposed model accommodates this increase well. When $E(I)$ is increased to 240 (Fig. 10), the estimations from the proposed model and the PFTK model are close. We then repeated the comparisons for $RTT = 400$ms, and obtained the similar results shown in Figs. 11 and 12.

C. *Mean square estimation error and error range*

To investigate and compare the sensitivity of our proposed model and the PFTK model to $E(D)/E(I)$, RTT, and p, we define *Squared Estimation Errors* of sending rate and throughput as $\epsilon_B^2 = \left(\frac{B_a - B_s}{B_s}\right)^2$ and $\epsilon_T^2 = \left(\frac{T_a - T_s}{T_s}\right)^2$, respectively. Here, B_a and B_s are the sending rate obtained from analytical models and simulations, respectively; similarly, T_a, and T_s are the throughput obtained from analytical models and simulations, respectively. The *Mean Squared Estimation Error* (MSEE) is defined as the mean of the Squared Estimation Errors of sending rate and throughput as $\overline{\epsilon_B^2}$ and $\overline{\epsilon_T^2}$, respectively. The 95% confidence interval for ϵ_B^2 and ϵ_T^2 are represented by $\widetilde{\epsilon_B^2}$ and $\widetilde{\epsilon_T^2}$, respectively. We compute and plot $\overline{\epsilon_B^2}$ and $\overline{\epsilon_T^2}$ for both the proposed model and the PFTK model. We also investigated $\widetilde{\epsilon_B^2}$ and $\widetilde{\epsilon_T^2}$ which indicate the oscillation of the estimation error around its mean value, of the two models.

First, we investigate the impact of the ratio $E(D)/E(I)$ on

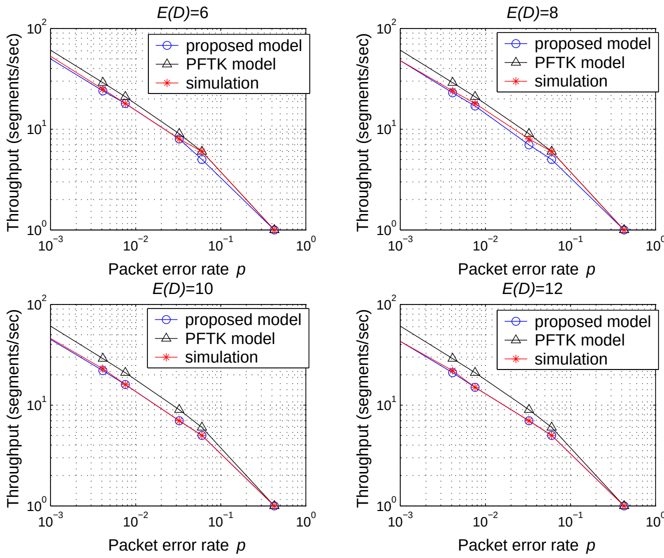


Fig. 11. Throughput estimation for RTT=400ms and $E(I)$=30 sec.

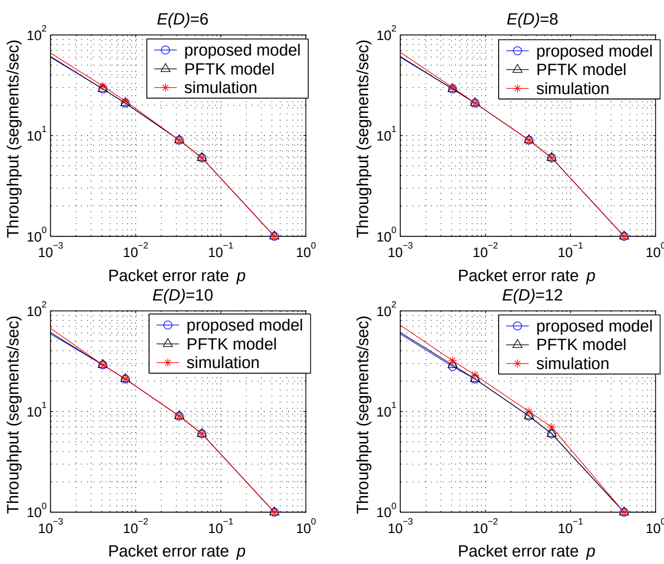


Fig. 12. Throughput estimation for RTT=400ms and $E(I)$=240 sec.

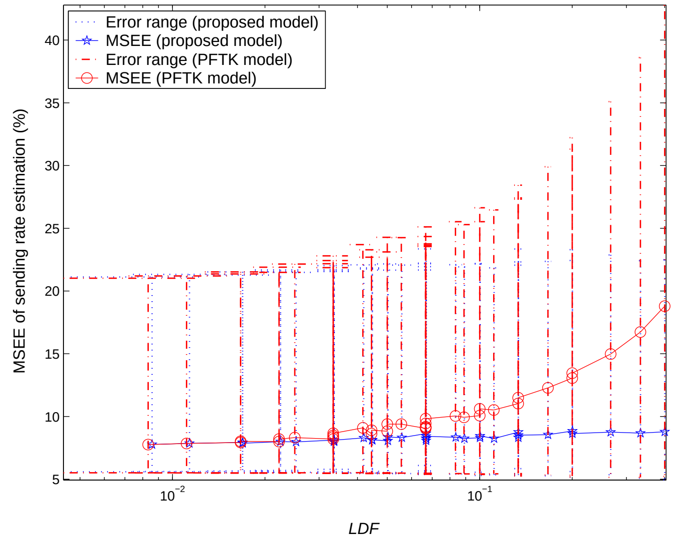


Fig. 13. Sending rate estimation error vs. LDF.

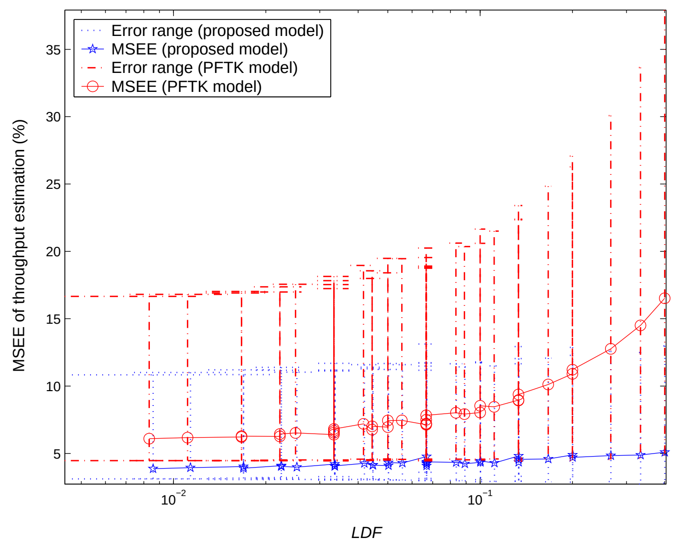


Fig. 14. Throughput estimation error vs. LDF.

ϵ_B^2 and ϵ_T^2 of the proposed and PFTK models. We define $E(D)/E(I)$ as the *Long Delay Frequency* (LDF), which represents the frequency of long delays within a period of time. Figs. 13 and 14 show $\overline{\epsilon_B^2}$, $\overline{\epsilon_T^2}$, $\widetilde{\epsilon_B^2}$, and $\widetilde{\epsilon_T^2}$ versus LDF. When LDF increases, PFTK model's ϵ_B^2 and $\overline{\epsilon_T^2}$ increase dramatically. However, we can observe that the proposed model's $\overline{\epsilon_B^2}$ and $\overline{\epsilon_T^2}$ are almost constant with increase of LDF values. This is because a higher D/I ratio means longer delays with relatively short intervals, thereby making the impact of long delays on the PFTK model more severe.

To determine the change of ϵ_B^2 and ϵ_T^2 of the two models as a function of RTT, we investigate the sensitivity of ϵ_B^2 and ϵ_T^2 versus RTT. Figs. 15 and 16 show $\overline{\epsilon_B^2}$, $\overline{\epsilon_T^2}$, $\widetilde{\epsilon_B^2}$, and $\widetilde{\epsilon_T^2}$ versus RTT. When the RTT increases, both $\overline{\epsilon_B^2}$ and $\overline{\epsilon_T^2}$ decrease. This is due to the fact that the impact of long delays on the sending rate and throughput becomes insignificant when the RTT increases.

Figs. 17 and 18 show $\overline{\epsilon_B^2}$, $\overline{\epsilon_T^2}$, $\widetilde{\epsilon_B^2}$, and $\widetilde{\epsilon_T^2}$ versus packet error rates. When p increases, both $\overline{\epsilon_B^2}$ and $\overline{\epsilon_T^2}$ increase. We can see that if we can control $p < 0.1$, we can expect the MSEE of the bandwidth and throughput of the proposed model to be under 5%.

VI. Conclusion

TCP has been found to perform poorly in the presence of spurious timeouts caused by delay spikes which are more frequent in today's wireless mobile networks as compared to traditional wired network. Previous analytical models didn't consider the effect of spurious timeouts on the steady state performance of TCP. In this paper, we developed an analytical model to study TCP sending rate and throughput as a function of packet error

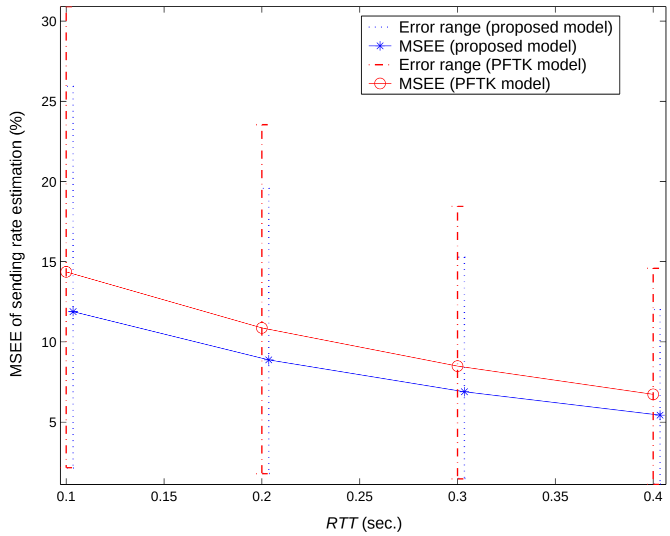


Fig. 15. Sending rate estimation error vs. RTT.

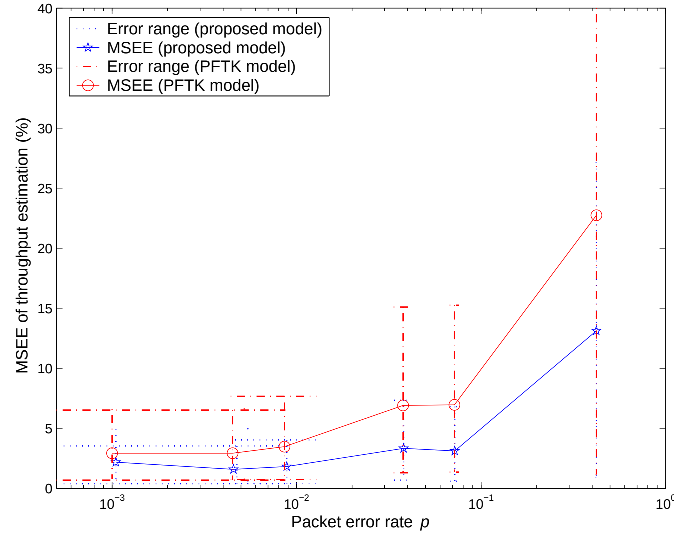


Fig. 16. Throughput estimation error vs. RTT.

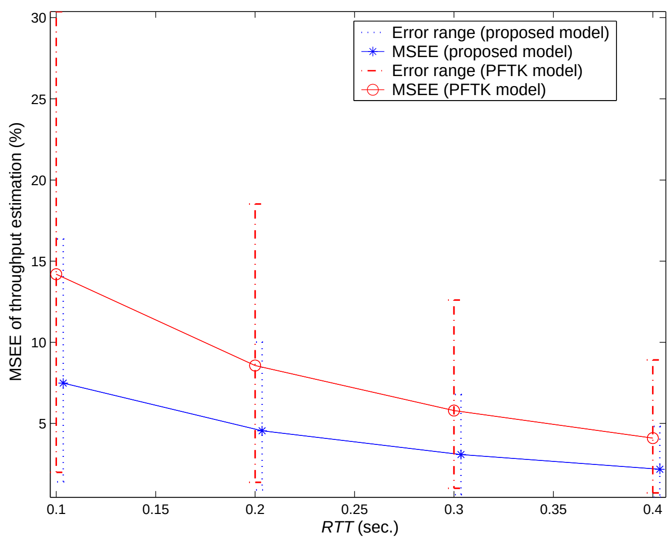


Fig. 17. Sending rate estimation error vs. p.

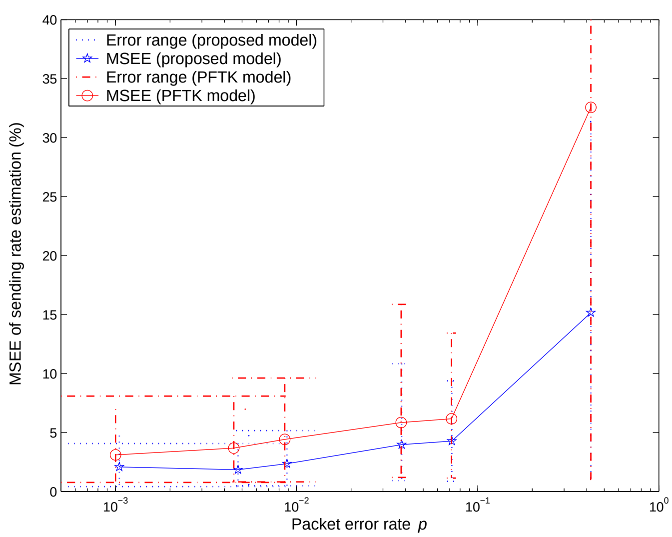


Fig. 18. Throughput estimation error vs. p.

rate and the characteristics of long delays. We have used simulation results to validate accuracy of the proposed model and compared with that of the PFTK model. We have shown that the proposed model is more accurate than the PFTK model in estimating the steady state sending rate and throughput of TCP in presence of frequent long delays.

Due to space limitations, we could not present the extension of our model to the finite receiver buffer case. However, the extension can be done by changing Eqns. (5), (6), and (11), and following the approach used in [13].

References

[1] A. Gurtov, "Effect of delays on TCP performance," in *IFIP Personal Wireless Communications*, August 2001.

[2] F. Khafizov and M. Yavuz, "Running TCP over IS-2000," in *IEEE International Conference on Communications*, New York, April 2002, pp. 3444–3448.

[3] M. Yavuz and F. Khafizov, "TCP over wireless links with variable bandwidth," in *56th IEEE Vehicular Technology Conference*, Vancouver, September 2002, pp. 1322–1327.

[4] D. S. Eom, H. Lee, and M. Sugano et. al., "Improving TCP handoff performance in mobile IP based networks," *Computer Communications*, vol. 25, no. 7, pp. 635–646, May 2002.

[5] A. Gurtov and R. Ludwig, "Making TCP robust against delay spikes," Internet Draft, draft-gurtov-tsvwg-tcp-delay-spikes-00.txt, February 2002.

[6] P. Karn and C. Partridge, "Improving Round-Trip Time estimates in reliable transport protocols," *ACM Computer Communications Review*, vol. 17, no. 5, pp. 67–73, August 1987.

[7] R. Ludwig and R. H. Katz, "The Eifel algorithm: Making TCP robust against spurious retransmission," *ACM Computer Communications Review*, vol. 30, no. 1, pp. 30–36, January 2000.

[8] R. Ludwig, "The TCP retransmit (rxt) flag," Internet Draft, draft-ludwig-tsvwg-tcp-rxt-flag-02.txt, November 2001.

[9] E. Blanton and M. Allman, "Using TCP DSACKs and SCTP Duplicate TSNs to detect spurious retransmissions," Internet Draft, draft-blanton-dsack-use-01.txt, August 2001.

[10] T.V. Lakshman and U. Madhow, "The performance of TCP/IP for networks with high bandwidth-delay products and random loss," *IEEE/ACM Transactions on Networking*, vol. 5, no. 3, pp. 336–350, June 1997.

[11] M. Mathis, J. Semke, and J. Mahdavi, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Computer Communications Review*, vol. 27, no. 3, pp. 67–82, July 1997.

[12] 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, pp. 485–498, August 1998.

[13] J. Padhye, V. Firoiu, D.F. Towsley, and J.F. Kurose, “Modeling TCP Reno performance: a simple model and its empirical validation,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–145, April 2000.

[14] M. Allman and V. Paxson, “On estimating end-to-end network path properties,” in *SIGCOMM*, Cambridge, MA, September 1999, pp. 263–274.

[15] L. Zhou, P.S.Y. Chan, and R. Radhakrishna Pillai, “Effect of TCP/LLC protocol interaction in GPRS networks,” *Computer Communications*, vol. 25, no. 5, pp. 501–506, March 2002.

[16] Maya Yajnik, Sue B. Moon, James F. Kurose, and Donald F. Towsley, “Measurement and modeling of the temporal dependence in packet loss,” in *INFOCOM’99*, New York, March 1999, pp. 345–352.

[17] K. Fall and S. Floyd, “Simulation-based Comparisons of Tahoe, Reno, and SACK TCP,” *ACM Computer Communications Review*, vol. 26, no. 3, pp. 5–21, July 1996.

[18] S. Floyd and T. Henderson, “The NewReno modification to TCP’s fast recovery algorithm,” IETF RFC 2582, April 1999.

[19] F. Khafizov and M. Yavuz, “Analytical model of RLP in IS-2000 CDMA networks,” in *56th IEEE Vehicular Technology Conference*, Vancouver, September 2002, pp. 487 –491.

[20] *The Network Simulator - ns-2*, http://www.isi.edu/nsnam/ns/.

[21] *NS TCP Eifel Page*, http://www-tkn.ee.tu-berlin.de/˜morten/eifel/ns-eifel.html.