

# 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 are generally considered to represent a transient state in wired networks, previous research did not consider the effects of ST on the steady state performance 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 proposed model has been validated against simulation results and has been found to be more accurate than previous models in the presence of spurious timeouts.

## I. INTRODUCTION

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]. A sudden increase of the instantaneous round trip time ( $RTT$ ) beyond the sender's retransmission timeout value ( $RTO$ ) causes *Spurious Timeouts*<sup>1</sup> (ST) which produces serious end-to-end TCP performance penalty [2], [3], [5]. Causes of delay spikes in a wireless mobile environment include [1]: (1) *handoff* of a mobile host between cells, (2) *physical disconnection* of the wireless link, (3) excessive retransmission by Radio Link Control (RLC) layer in wireless mobile networks (such as GPRS and CDMA2000), (4) higher-priority traffic, such as circuit-switched voice, can preempt (block) the data traffic temporarily.

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, the Eifel algorithm [5], TCP-D-SACK option [6], and conservative management of TCP's retransmission timer [4]. 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* [7], [8], [9]. The models by Lakshman et. al. [7] and Mathis et. al. [8] only considered slow start and congestion avoidance, but not timeouts. The model proposed by Padhye et. al. [9] improves the one in [8] 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*. In wireless mobile environments, STs should be considered explicitly in order to accurately model the steady state throughput of TCP. This

<sup>1</sup>**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  $RTO$  [5].

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. We use the result of Padhye et. al. [9] 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 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 [10]. 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 [11], [2]. The settings of lower layer protocols, such as handoff schemes in Mobile IP and retransmission schemes at the link layer, have a non-trivial 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 al-

ways has sufficient data to send. 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 [9]).
- The time required to send a window of data is smaller than an RTT. This assumption was also used in the PFTK model [9].
- 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  $RTT$ .

### III. ANALYTICAL MODEL

In this section, our objective is to develop an analytical model for the sending 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 [9].

- $I$  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  $TDP$ s in one  $NP$ .
- $LDC$  long delay cycle, which consists of  $m$   $NP$ s and one  $LDP$  (see Fig. 3).
- $m$  number of  $NP$ s 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 [9] instead of  $Z^{TDP}$  and  $Z^{NP}$ .

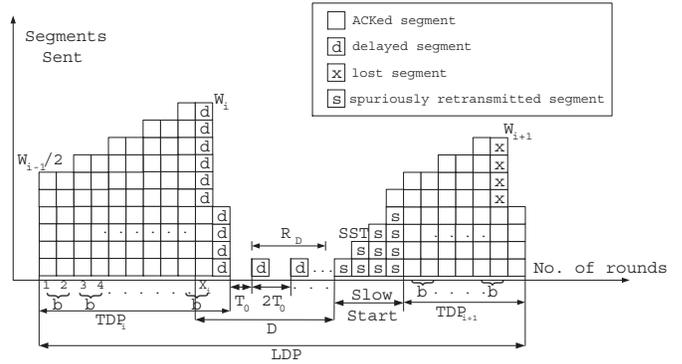


Fig. 1. Segments sent during one Long Delay Period ( $LDP$ ).

- $Z^{TDP}$  duration of  $n$  instances of  $TDP$ s 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  $NP$ ,  $r = 1, 2 \dots m$ .
- $R$  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

We first analyze the dynamics of the sender's window around a long delay. 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  $SST$ ).

TCP Reno starts fast retransmit after receiving three duplicate ACKs, which are called *triple-duplicate loss indications*. Triple Duplicate Period (TDP) is defined in [9] 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

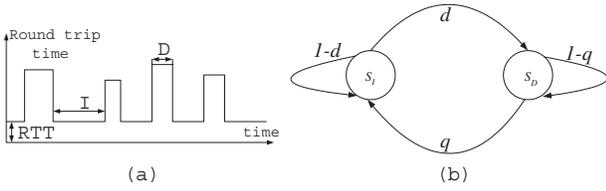


Fig. 2. (a) Variation of RTT showing four long delays, and (b) model of long delays.

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 ( $S_I$ ) 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:

$$E(D) = \frac{dE(I)}{q} \quad (1)$$

Given a model for the lower layer events (such as link layer retransmission, mobile handoff, etc. [12]) 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

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

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 LDC 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):

$$E(U) = 2E(Y) + E(R_D) + E(K) \quad (3)$$

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  $TDP_i$ :

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

$E(Y)$  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 [9] as follows:

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

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

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

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:

$$\begin{aligned} Pr(T_0 < D \leq 2T_0) &= Pr(D \leq 2T_0) - Pr(D \leq T_0) \\ &= e^{-\frac{T_0}{E(D)}} - e^{-\frac{2T_0}{E(D)}} \end{aligned} \quad (8)$$

Similarly, the probability of  $j$  timeouts is:

$$Pr(2^{j-1}T_0 < D \leq 2^jT_0) = e^{-\frac{2^{j-1}T_0}{E(D)}} - e^{-\frac{2^jT_0}{E(D)}} \quad (9)$$

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:

$$\begin{aligned} E(R_D) &= 1Pr(T_0 < D \leq 2T_0) + 2Pr(2T_0 < D \leq 4T_0) + \\ &\quad \dots + 6Pr(32T_0 < D \leq 64T_0) \\ &= \sum_{j=0}^5 \left( e^{-\frac{2^jT_0}{E(D)}} - e^{-\frac{64T_0}{E(D)}} \right) \end{aligned} \quad (10)$$

**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. The probability that there are two or more timeouts is:

$$Pr(D > 2T_0) = e^{-\frac{2T_0}{E(D)}} \quad (11)$$

Therefore, the expected value of  $SST$  after the long delay is:

$$E(SST) = \max(W_i/2, 2) \left( e^{-\frac{T_0}{E(D)}} - e^{-\frac{2T_0}{E(D)}} \right) + 2e^{-\frac{2T_0}{E(D)}} \quad (12)$$

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:

$$cwnd_{j+1} = cwnd_j + \left\lceil \left( \frac{cwnd_j}{2} \right) \right\rceil \quad \text{with } cwnd_1 = 1, \quad j = 1, 2, 3 \dots \quad (13)$$

which can be approximated as:

$$cwnd_j = \left( \frac{3}{2} \right)^j \quad j = 1, 2, 3 \dots \quad (14)$$

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:

$$v = \left\lceil \log_{3/2}(E(SST)) \right\rceil \approx \lceil 1.71 \log(E(SST)) \rceil \quad (15)$$

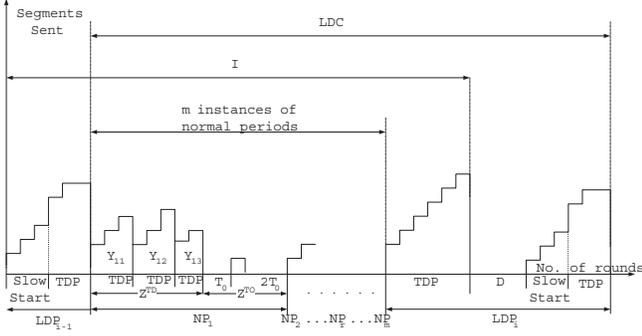


Fig. 3. Sender window evolution during one Long Delay Cycle (LDC).

**Determination of  $E(K)$ :** The number of segments sent in each round of the slow start stage in Fig. 1 is given in Eqn. (14). So the number of segments sent during slow start can be approximated by the sum of the segments sent during these  $v$  rounds:

$$E(K) = \sum_{j=1}^v \left(\frac{3}{2}\right)^j \approx 3 \left(\frac{3}{2}\right)^{\lceil 1.71 \log(E(SST)) \rceil} - 3 \quad (16)$$

By substituting  $E(R_D)$ ,  $v$ , and  $E(K)$  from Eqns. (10), (15), and (16) into Eqns. (3) and (4), we can obtain the number of segments sent and the duration of one LDP.

2) *Analysis of one Long Delay Cycle (LDC):* In Eqn. (2),  $Z^{NP}$  can be obtained from [9] as given in Eqn. (17),  $E(Z^{LDP})$  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 [9] as given in Eqns. (17) and (18).

$$\begin{aligned} E(Z^{NP}) &= E(n)E(Z^{TDP}) + E(Z^{TO}) \\ &= \left(\frac{b}{2}E(W) + 1\right) E(n)RTT + T_0 \frac{f_p}{1-p} \end{aligned} \quad (17)$$

$$\text{where } f_p = 1 + \sum_{i=1}^6 2^{i-1} p^i$$

$$E(M_r) = E(n)E(Y) + E(R) = \frac{\frac{1-p}{p} + \sqrt{\frac{8}{3bp}}}{\min\left(1, 3\sqrt{\frac{3bp}{8}}\right)} + \frac{1}{1-p} \quad (18)$$

$E(n)$  and  $E(R)$  in Eqns. (17) and (18) can be determined from Eqns. (19) and (20), which can be obtained from the PFTK model [9] as given below.

$$E(n) = \frac{1}{\min\left(1, \frac{3}{E(W)}\right)} \quad (19)$$

$$E(R) = \frac{1}{1-p} \quad (20)$$

$E(U)$  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 [9] as given in Eqn. (17).

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:

$$m = \frac{E(I) - 2E(Z^{TDP}) - v \cdot RTT}{E(Z^{NP})} \quad (21)$$

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

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:

$$E(G) = \sum_{r=1}^m M_r + E(U) = mE(M_r) + E(U) \quad (22)$$

By substituting  $E(G)$  from Eqn. (22) 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  $E(Y)$ .

$$E(Y'_1) = E(Y) - E(W) = \frac{1-p}{p} \quad (23)$$

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  $\frac{E(W)}{2}$ .

$$E(Y'_2) = E(Y) - \frac{E(W)}{2} = \frac{1-p}{p} + \frac{E(W)}{2} \quad (24)$$

Because the segments retransmitted during the timeout period are discarded by the receiver, we can replace  $E(R)$  in Eqn. (18) with  $E(R') = 1$ . Similarly, we have  $E(R'_D) = 1$ . Replacing  $E(Y)$ ,  $E(R)$  and  $E(R_D)$  in Eqns. (3) and (18) with  $E(Y')$ ,  $E(R')$  and  $E(R'_D)$ , we obtain:

$$E(U') = E(Y'_1) + E(R'_D) + E(K) + E(Y'_2) \quad (25)$$

$$E(M'_r) = E(n)E(Y'_2) + E(R') \quad (26)$$

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. (22) 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:

$$T(p, I, D) = \frac{mE(M'_r) + E(U')}{mE(Z^{NP}) + E(Z^{LDP})} \quad (27)$$

## 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* network simulator in Sec. V. The long delays are simulated using

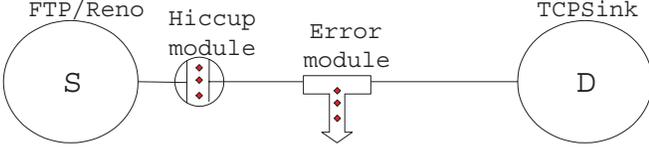


Fig. 4. *ns-2* simulation topology with hiccup and error modules.

TABLE I

SIMULATION PARAMETERS FOR THE TOPOLOGY OF FIG. 4.

| Protocol                | TCP Reno     |
|-------------------------|--------------|
| Header size             | 40 bytes     |
| Payload size            | 536 bytes    |
| <i>rwnd</i> limit       | 800 segments |
| Initial <i>ssthresh</i> | 800 segments |
| link bandwidth          | 300Mbps      |
| link propagation delay  | 50 - 200 ms  |
| packet error rate       | 0.001 - 0.5  |

an *ns-2* module called "hiccup" which delays all the arriving segments for time  $D$ . The simulation topology is shown in Fig. 4, where the link equipped with a hiccup module and an error module. The packet error rate can thus be accurately controlled by the error module.

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  $RTT = 200$ ms,  $E(I) = 30$  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  $RTT = 400$ ms, and obtained similar results as shown in Figs. 7 and 8.

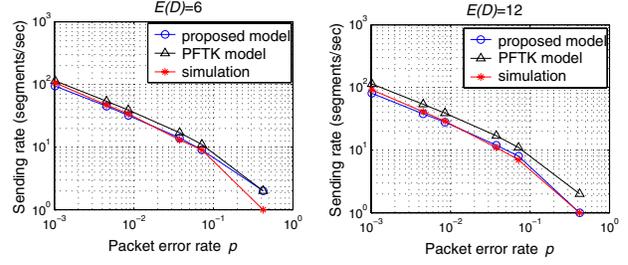


Fig. 5. Sending rate estimation for  $RTT=200$ ms and  $E(I)=30$  sec.

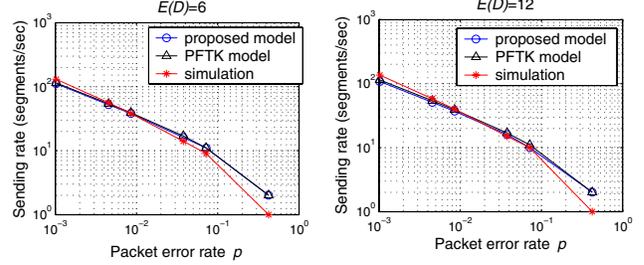


Fig. 6. Sending rate estimation for  $RTT=200$ ms and  $E(I)=240$  sec.

### B. Comparison of throughput estimation

Next, we compare the predicted throughput from the proposed model and the PFTK model against the values obtained from simulation. Here, we only show the result for  $RTT = 200$ ms,  $E(I) = 30$  and  $E(D)$  ranging from 6 to 12 seconds in Fig. 9. For other cases like shown in Sec. V-A, please refer to our full-sized version of this paper at [13]

### 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  $T_s$

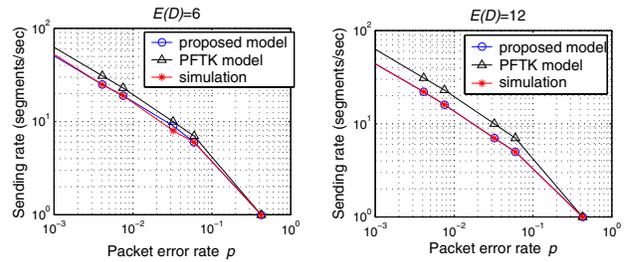


Fig. 7. Sending rate estimation for  $RTT=400$ ms and  $E(I)=30$  sec.

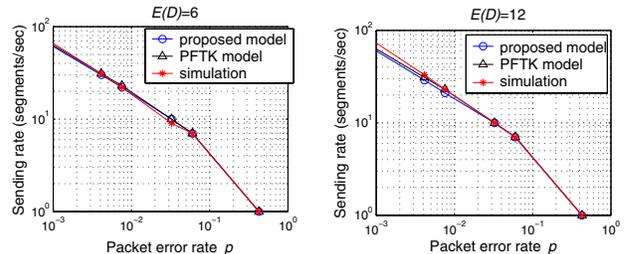


Fig. 8. Sending rate estimation for  $RTT=400$ ms and  $E(I)=240$  sec.

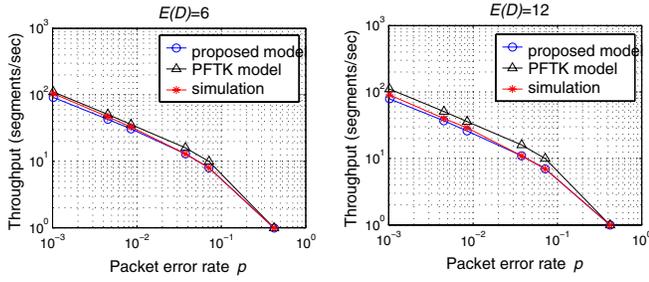


Fig. 9. Throughput estimation for  $RTT=200ms$  and  $E(I)=30$  sec.

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  $\overline{\epsilon_B^2}$  and  $\overline{\epsilon_T^2}$  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  $\overline{\epsilon_B^2}$  and  $\overline{\epsilon_T^2}$  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. 10(a) and 10(b) 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  $\overline{\epsilon_B^2}$  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.

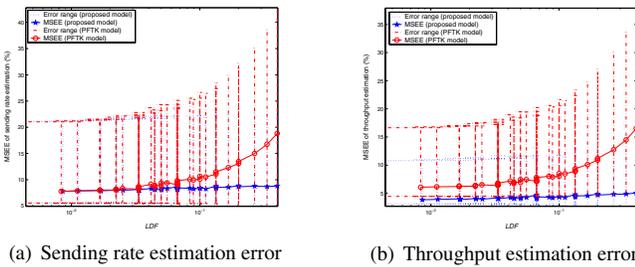


Fig. 10. Sending rate and throughput estimation error vs. LDF.

To determine the change of  $\overline{\epsilon_B^2}$  and  $\overline{\epsilon_T^2}$  of the two models as a function of  $RTT$ , we investigate the sensitivity of  $\overline{\epsilon_B^2}$  and  $\overline{\epsilon_T^2}$  versus  $RTT$ . Figs. 11(a) and 11(b) 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. 12(a) and 12(b) 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 performs poorly in the presence of spurious timeouts caused by delay spikes in today's wireless mobile networks.

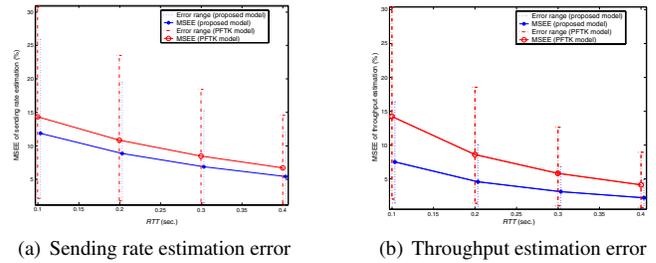


Fig. 11. Sending rate and throughput estimation error vs.  $RTT$ .

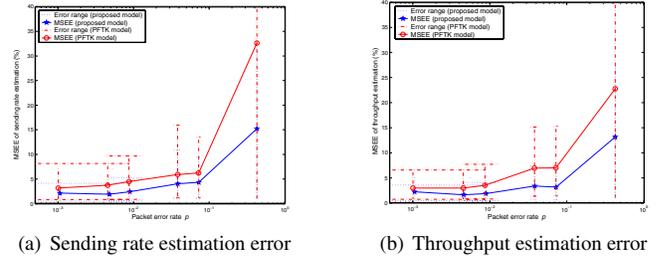


Fig. 12. Sending rate and throughput estimation error vs.  $p$ .

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 and 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 (12), and following the approach used in [9].

## 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] A. Gurtov and R. Ludwig, "Making TCP robust against delay spikes," Internet Draft, draft-gurtov-tsvwg-tcp-delay-spikes-00.txt, February 2002.
- [5] 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.
- [6] 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.
- [7] T.V. Lakshman and Ü. 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.
- [8] 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.
- [9] 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.
- [10] M. Allman and V. Paxson, "On estimating end-to-end network path properties," in *SIGCOMM*, Cambridge, MA, September 1999, pp. 263–274.
- [11] 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.
- [12] 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.
- [13] S. Fu and M. Atiquzzaman, "Modelling TCP Reno with Spurious Timeouts in Wireless Mobile Environment," Tech. Rep., Computer Science, University of Oklahoma, www.cs.ou.edu/~atiq, May 2003.