

Modeling TCP over Selective Repeat ARQ in Wireless Networks with Non-Negligible Propagation Delay

Jing Zhu, Sumit Roy

{zhu, roy}@ee.washington.edu

Department of Electrical Engineering, University of Washington, Seattle WA 98195.

Abstract- In this paper, we propose an analytical model for TCP over selective repeat ARQ in wireless networks, in which random loss is the primary impediment to reliable, high-speed data services. We focus on the issue of modeling RTT (round trip time) variation due to ARQ retransmission at the link layer with non-negligible propagation delay. With the proposed model, we present upper and lower bounds for TCP end-to-end throughput. Simulation with ns2 is performed to validate our analysis.

Key Words: Modeling, ARQ, TCP/IP, satellite networks

I. INTRODUCTION

Global networks that provide high-speed data services invariably will contain a wireless segment (e.g. cellular, satellite, etc.). The next generation Internet is anticipated to provide support for various popular applications such as web browsing, file transfer etc. on-board airplanes or ships by using an appropriate TCP/IP based wireless network. However, wireless loss is known to adversely affect TCP end-to-end performance[1]. In the literature, there have been many proposals to improve TCP performance in wireless networks [2] among which "TCP splitting" has attracted some attention due to its ease of implementation (since it does not require modification to TCP stack in the end systems). On the other hand, a reliable protocol (such as selective repeat ARQ) can be used at link layer to prevent TCP packets losses in the wireless channel. The Wireless IP Suite Enhancer (WISE) [10] developed at the Lincoln Laboratory is one such scheme that adopts both "TCP splitting" technique and SR-ARQ to enhance TCP performance for satellite networks. [11] proposed an efficient ARQ scheme for GEO satellite networks, by adding more information about receiver status into ACK packets. Nevertheless, such approaches lead to new problems - the interaction between TCP and link layer protocol, both of which offer reliable data transfer, may impact the end-to-end performance significantly due to the possibility for greater variability in (end-to-end) round trip time due to (random number) link layer retransmissions.

An added significance our work accrues due to the relative lack of analytically inspired research on modeling and analyzing TCP performance over link layer protocols in wireless networks. Of the few earlier studies, [3] investigated TCP/RLP

performance with CDMA wireless link; as FER (frame error rate) increases, it suggested increasing the number of retransmissions at link layer to alleviate TCP throughput degradation. [4] and [5] considered the effect of forward error correction (FEC), and [6] studied the interaction between TCP and ARQ as well. However all of them relied primarily on simulation, and did not propose any substantive analytical model. Some useful analytical models were proposed in [7] [8], but they focused on the presence of burst errors in a fading channel and ignored wireless propagation delay which is not feasible when the propagation delay is not negligible such as in a satellite network. [12] evaluated performance of hybrid ARQ in LEO satellite networks, but did not study TCP performance. [13] proposed an analytical model to evaluate the performance of TCP over Go-Back-N ARQ in UMTS environments, whereas Go-Back-N ARQ is not efficient compared with SR-ARQ.

The paper is organized as follows, In Section 2, we describe our model assumptions. Delay and delay variations caused by retransmission of SR-ARQ is analyzed in Section 3. We provide our analytical model for throughput analysis for TCP over SR-ARQ in wireless networks in Section 4. Section 5 shows simulation results validating our model. We conclude the whole paper in Section 6.

II. MODEL ASSUMPTION

Fig. 1 shows the system model for the following analysis. We assume that the terrestrial propagation delay is short enough to be ignored, when compared with wireless propagation delay (such as in satellite networks). Other important assumptions on which our analysis is based are shown as follows.

1. In-order packet delivery policy is assumed for the SR-ARQ data link, Otherwise, spurious retransmissions due to out-of-order packets will take place, degrading performance significantly. Therefore, a buffer is needed for reordering at the link layer of the receiver, denoted by B_{Rv} (see Fig.1). Furthermore, we assume B_{Rv} is big enough so that no received packets will be lost at the receiver.
2. For the purpose of protecting TCP from unreliable wireless channel, fully reliable SR-ARQ is used at link layer, implying that each packet is retransmitted

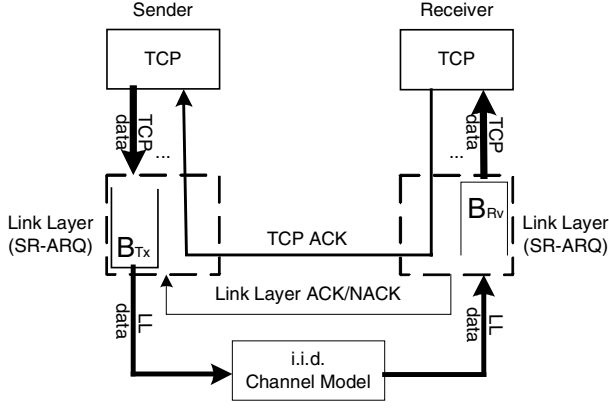


Fig. 1. System Model

till success. Therefore, the link layer of the sender needs a buffer to store packets, denoted by B_{Tx} .

3. Compared with propagation delay, transmission time of individual link layer packets is ignored.
4. Wireless channel losses are modeled as i.i.d, which is reasonable if sufficient interleaving is assumed in the physical layer. Thus, variations of round trip time caused by retransmission can be assumed to be independent; in such cases, timeouts do not occur. Without timeout but only congestion losses, TCP sender always remains in congestion avoidance in steady state.
6. We assume only standard ACK scheme (no delayed ACKs) by which one TCP ACK is generated for each received TCP data packet and sent back to TCP sender with no delay.
7. ACK/NACK method is also used at link layer; for each received link layer packet, ACK is sent for success and NACK for failure.
8. We assume that both TCP ACK packets and Link layer ACK/NACK packets are error-free. This is reasonable in most cases since their length is much smaller when compared with data packets. Furthermore, since they constitute control traffic with higher priority, powerful forward error correction (FEC) schemes can be used to protect them from losses.

III. MODELING DELAY CAUSED BY SR-ARQ RETRANSMISSION

We assume that link layer (LL) packet loss rate in a wireless channel is p , the round trip time of the wireless channel is T , the transmission time of one LL packet is δ , and each TCP packet is segmented to S LL packets. The probability distribution of total delay d for transmission of a LL packet is then given by

$$P(d = iT) = p^{(i-1)}(1 - p), \quad (1)$$

i.e. a geometric distribution. Assuming that n successive TCP packets await transmission, we have nS LL packets in buffer B_{Tx} after segmentation. As follows, we will derive the probability distribution of total delay $d^{(nS)}$ (see Fig.2), which is the time to successfully transmit all nS LL packets.

At first, we ignore the transmission time of each packet so that nS LL packets can be regarded as essentially being sent at the same time. Since the probability distribution of d for each LL packet is identical, independent and given by Eq.(1), we have

$$\begin{aligned} P(d^{(nS)} = iT) &= [P(d \leq iT)]^{nS} - [P(d \leq (i-1)T)]^{nS} \\ &= \left[\sum_{j=1}^i P(d = jT) \right]^{nS} - \left[\sum_{j=1}^{i-1} P(d = jT) \right]^{nS} \\ &= (1 - p^i)^{nS} - (1 - p^{(i-1)})^{nS}. \end{aligned} \quad (2)$$

The mean of $d^{(nS)}$ is given by

$$\begin{aligned} E(d^{(nS)}) &= \sum_{j=1}^{\infty} jTP(d^{(nS)} = jT) \\ &= T \sum_{i=1}^{nS} \binom{nS}{i} \frac{1}{1 - p^i} (-1)^{i-1}. \end{aligned} \quad (3)$$

If nS equals 1, Eq.3 simplifies to

$$E(d^{(1)}) = \frac{1}{1 - p} T, \quad (4)$$

as expected.

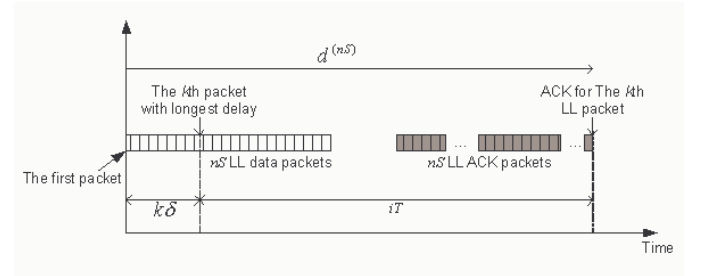


Fig. 2. Delay caused by SR-ARQ retransmission for n successively sent packets

In fact, packets are sent successively not simultaneously. Assume that the k th packet is 'slowest', i.e. requires the largest number of transmission attempts prior to success. The duration between the start time of the first transmission and that of the k th packet is δk ($1 \leq k \leq nS$). Since the delay distribution of n packets is identical and independent, they have the equal probability $\frac{1}{nS}$ to be the slowest one. Therefore, the mean duration is

$$\sum_{k=1}^{nS} \frac{1}{nS} k\delta = \frac{1 + nS}{2} \delta. \quad (5)$$

Thus a more accurate result for the mean of $d^{(nS)}$ is

$$E(d^{(nS)}) = T \sum_{j=1}^{nS} \binom{nS}{j} \frac{1}{1 - p^j} (-1)^{j-1} + \frac{(nS + 1)\delta}{2}. \quad (6)$$

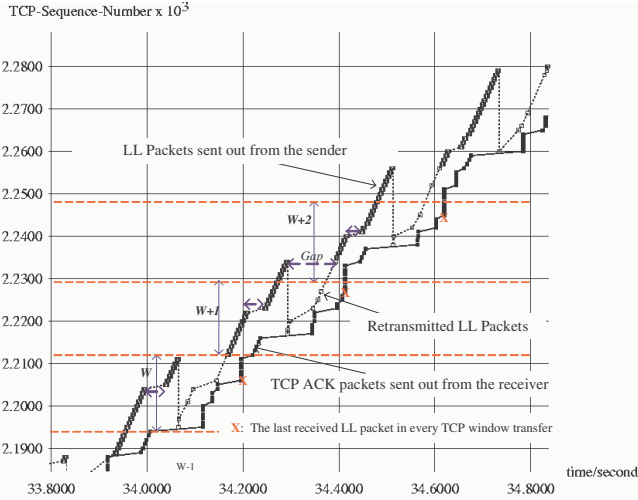


Fig. 3. Tracing Packet Transmission ($C=20$ TCP pkts, $T=100$ ms, $S=10$)

IV. ANALYSIS OF TCP OVER SR-ARQ

Fig.3 shows a typical series of packet transmissions, that show the start time of transmission of a link layer packet and the corresponding TCP ACK, and the TCP sequence number of the slowest LL packet in every window transfer. We see that the time spend for successfully sending each TCP packet on the wireless channel is different due to channel loss and retransmission. A link layer packet arriving 'earlier' at the receiver must wait for the slower packets according to in-order delivery policy. Consequently, TCP ACK packets are generated in bunches, and each TCP window transfer is segmented into several continuous parts with a gap in between (see Fig.3). In Fig.3, "X" denotes the slowest packet in each window transfer. It shows that the slowest packet belongs to the final continuous segment. Furthermore, the number of packets in the final continuous segment is a random variable and depends on the position of the slowest packet in the previous window transfer - we assume to be uniformly distributed on $[2, w]$, where w is the size of TCP congestion window in the current window transfer. The lower bound 2 corresponds to the case that the last packet in the previous window is slowest, and w corresponds the case that the first packet in the previous window is slowest.

We use $r^{(w)}$ to denote the total transmission time of the final continuous part with the TCP window size of w . The mean of $r^{(w)}$ is given by

$$\begin{aligned} E(r^{(w)}) &= \sum_{n=2}^w \frac{1}{w-1} E(d^{(nS)}) \\ &= \sum_{n=2}^w \left[\frac{1}{w-1} T \sum_{j=1}^{nS} \binom{nS}{j} \frac{1}{1-p^j} (-1)^{j-1} \right] + \frac{wS+4}{4} \delta. \end{aligned} \quad (7)$$

The above represents the approximate average time for TCP congestion window to increase by one at the window size of w during the congestion avoidance period. In the following, we investigate when the overflow of sender buffer B_{Tx} takes place. We assume that at time t the number of TCP packets in

receive buffer is $b_{Rv}(t)$, the number of TCP packets in transmit buffer is $b_{Tx}(t)$, the number of TCP ACK packets in travel is $Ack(t)$, and TCP congestion window size is $w(t)$ (note that actually both b_{Rv} and b_{Tx} are link layer packets, but measured in units of a TCP packet). Obviously, overflow occurs when $b_{Tx}(t)$ exceeds the transmit buffer capacity which is denoted by C .

Since $w(t)$ indicates the total number of TCP packets without acknowledgement, we have

$$w(t) = b_{Tx}(t) + b_{Rv}(t) + Ack(t). \quad (8)$$

Note that both $Ack(t)$ and $b_{Rv}(t)$ do not include the packets which have not been removed from B_{Tx} . From Eq.8, we see that $b_{Tx}(t)$ reaches its local maximum $w(t)$ when both $b_{Rv}(t)$ and $Ack(t)$ fall to zero. At this time, if $w(t) > C$, overflow occurs and the number of lost packets equals $w(t) - C$.

Assuming that the reordering buffer B_{Rv} becomes empty at time t_o , we have after one-way propagation delay, all TCP ACK packets sent before t_o will arrive at the sender. Although new packets might arrive at the receiver during the period from t_o to $t_o + \frac{T}{2}$, they cannot be removed from the receiver buffer during this period due to the propagation delay. Since $b_{Rv}(t)$ and $Ack(t)$ do not include the packets still in transmit buffer, both of them are zero at time $t_o + \frac{T}{2}$ and $b_{Tx}(t)$ reaches its local maximum $w(t)$. We know $w(t)$ is constant in every window transfer. Therefore, b_{Tx} reaches its local maximum $w(t)$ iff the receive buffer becomes empty sometime within a window transfer.

Since the local maximum value of $b_{Tx}(t)$ is $w(t)$, overflow only takes place after $w(t)$ is built up to $C + 1$. Let us consider two cases. In case I, B_{Rv} becomes empty in every window transfer. Finally, as $w(t)$ reaches $C + 1$, overflow will happen and one packet will be lost. In case II, B_{Rv} never becomes empty during the window transfer with $w(t)$ equal $C + 1$, and will become empty in the next window transfer when $w(t)$ equals $C + 2$. At that time, overflow takes place and two packets are lost. Of course, it is possible for B_{Rv} to still remain nonempty in the next window transfer, leading to burst loss. But we assume that its probability is small enough to be ignored, compared to case I and II. In fact, in our simulation we only see case I and II.

Now we calculate the average TCP end-to-end throughput in both of these two cases.

Case I: Congestion loss occurs when $w(t)$ equals $C + 1$, then $w(t)$ halves to $\frac{C+1}{2}$. Therefore, in steady state $w(t)$ changes between $\frac{C+1}{2}$ and $C + 1$. The number (N) of transmitted TCP packets for $w(t)$ increasing from $\frac{C+1}{2}$ to $C + 1$ is

$$N = \frac{3}{8}(C+1)(C+3). \quad (9)$$

The duration time D of this period is

$$\begin{aligned} D &= \sum_{w=\frac{C+1}{2}}^{C+1} E(r^{(w)}) \\ &= \sum_{w=\frac{C+1}{2}}^{C+1} \sum_{k=2}^w \left[\frac{1}{w-1} T \sum_{j=1}^{kS} \binom{kS}{j} \frac{1}{1-p^j} (-1)^{j-1} \right] \\ &\quad + \frac{C+3}{32}(16+3CS+3S)\delta \end{aligned} \quad (10)$$

Average throughput λ_1 can be computed as

$$\lambda_1 = \frac{N}{D}. \quad (11)$$

Case II: Overflow happens at the time $w(t)$ equals $C + 2$, and two TCP packets are lost. Then $w(t)$ decreases to quarter of its previous value. In steady state, $w(t)$ changes between $\frac{C+2}{4}$ and $C + 2$. The number N of transmitted packets for $w(t)$ increasing from $\frac{C+2}{4}$ to $C + 2$ is

$$N = \frac{5}{32}(C + 2)(3C + 10). \quad (12)$$

The duration time D of this period is

$$D = \sum_{w=\frac{C+2}{4}}^{C+2} \sum_{k=2}^w \left[\frac{1}{w-1} T \sum_{j=1}^{kS} \binom{kS}{j} \frac{1}{1-p^j} (-1)^{j-1} \right] \quad (13)$$

$$+ \frac{3C+10}{4} \left(1 + \frac{5}{32} CS + \frac{10}{32} S \right) \delta.$$

Average throughput λ_2 can be computed as

$$\lambda_2 = \frac{N}{D}. \quad (14)$$

Obviously, the accurate average throughput λ should depend on both of two cases above. However, it is difficult to give out the exact probability for each of them. But we can use Eq.11 and Eq.14 to form the lower bound and upper bound, which are valuable for performance evaluation as well.

$$\lambda_2 \leq \lambda \leq \lambda_1. \quad (15)$$

In what follows, we show some simulation traces to validate our above analysis. In Fig.4, we see that the occupancy of reordering buffer often decreases to zero. Fig.5 show that in steady state, only two cases mentioned above take place.

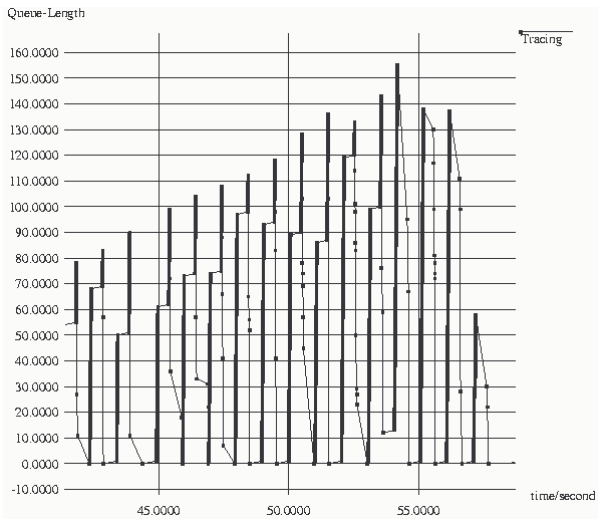


Fig. 4. Tracing the number of packets in reordering buffer B_{Rv} ($p=0.2$, $C=30$ TCP pkts, $T=500\text{ms}$, $S = 5$)

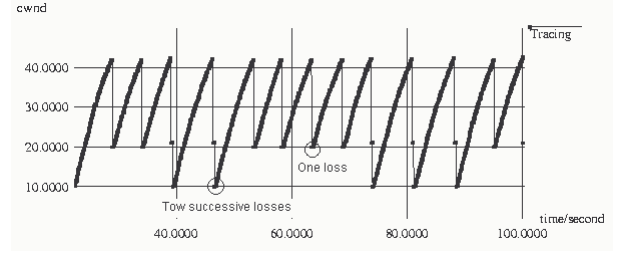


Fig. 5. Tracing TCP congestion window ($p=0.1$, $C=40$ TCP pkts, $T=100\text{ms}$, $S = 10$)

V. MODEL VALIDATION AND DISCUSSION ON NUMERICAL RESULTS

We use ns2 to simulate a network that consists of a sender-receiver pair connected via a lossy link with bandwidth 1 Mbps (bit per second). The size of TCP data packet is fixed with 500 bytes leading to the maximum output rate of 250 TCP packets per second. The loss rate is calculated in units of TCP packet no matter whether segmentation is used or not - Fig.6 does not consider segmentation whereas Fig.7 does ($S = 10$). In addition, different buffer sizes of B_{Tx} are investigated. The buffer size for Fig.6a and Fig.7a is 20 (TCP packet), and that for Fig.6b and Fig.7b is 40 (TCP packet). From Fig. 6 and Fig. 7 we can see that simulation results are bounded very well in the analytical range, showing that our model works well.

Fig.6b shows that lower the packet loss rate, the closer the throughput approaches the upper bound. As is known, the queue length of the reordering buffer should be dominated by the variation of the transmission time for a packet. Therefore, the greater the variation, greater the number of packets at the reordering buffer in the receiver. As a result, the time the reordering buffer stays empty in a round increases with decreasing variation. For example, if the packet rate is zero, then the variation is zero too. Every packet is delivered in order so that the reordering buffer is always empty. Therefore, in the case with less packet loss rate, the probability of Case I, in which the reordering buffer being empty at least once in every window transfer, is larger so that the throughput is much closer to the upper bound.

Moreover, there are some other factors that affect how long the reordering buffer stays empty, such as the buffer size of the transmitter. For example, if this buffer is small enough that the link will be far below the full utilization, then it is expected that the reordering buffer is empty for some duration in every window transfer. Generally, the smaller the size of the send buffer B_{Tx} , the larger the time in every window transfer when the reordering buffer B_{Rv} stays empty. The throughput in Fig.7a with the buffer size of 20 is much closer to the upper bound than that in Fig.7b with buffer size of 40. However, the difference between Fig.6 (a) and (b) is not obvious because the variation of the transmission time in Fig.6 is greatly increased because segmentation is not used. In other words, segmentation reduces the variation of the transmission time and has higher resource utilization.

On the other hand, from simulation results we see that the results are bounded more tightly in the case with the longer round trip time. In fact, this is because the absolute value of

the throughput is smaller for the longer round trip time. The ratio of the lower bound to the upper bound has little difference for different RTT. Fig.8 shows the result for the buffer size of 20 (TCP packets) and no segmentation. The total difference is below 0.01.

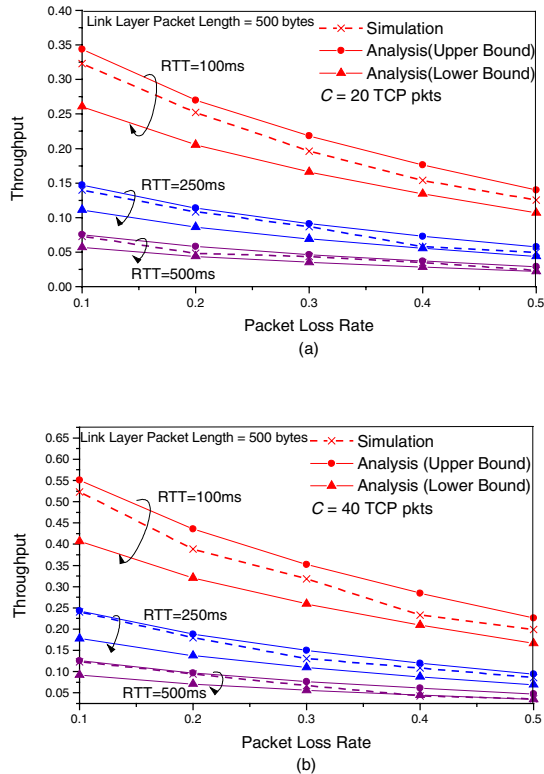


Fig. 6. Throughput vs Packet Loss Rate (without segmentation)

VI. CONCLUSION

In this paper, we studied the delay variation caused by SR-ARQ retransmission. Propagation delay and random losses in a wireless networks were concerned. Assuming a wireless channel with i.i.d model, we derived the distribution of total delay for n successively sent packets. We analyzed the problem of congestion loss due to long persistence time of full reliable SR-ARQ with in-order packet delivery policy. Simulation and analysis results show two main cases of packet loss. The upper bound and lower bound of throughput was derived and validated by simulation.

REFERENCES

- [1] P. Charalambos, V. S. Frost, and J. B. Evans, "Performance of TCP Extensions on Noisy High BDP Networks," IEEE Communications Letters, Vol. 3, No. 10, October, 1999.
- [2] N. Ghani and S. Dixit, "TCP/IP Enhancements for Satellite Networks", IEEE Commun. Magazine, July 1999, pp.64-72
- [3] G. Bao, "Performance Evaluation of TCP/RLP Protocol Stack over CDMA wireless link", ICCT'96, 1996, vol.2, pp. 710-713.
- [4] D. Hang and J. J. Shi, "Performance of TCP over Radio Link with Adaptive Channel Coding and ARQ", VTC'99, 1999, vol.3, pp. 2084-2088.
- [5] A. Chockalingam, M. Zorzi and V. Tralli, "Wireless TCP Performance with Link Layer FEC/ARQ", ICC'99, 1999, vol.2, pp. 1212-1216.
- [6] Y. Bai, A. T. Ogielski, G. Wu, "Interactions of TCP and Radio Link ARQ Protocol", VTC'99, 1999, vol.3, pp. 1710-1714.

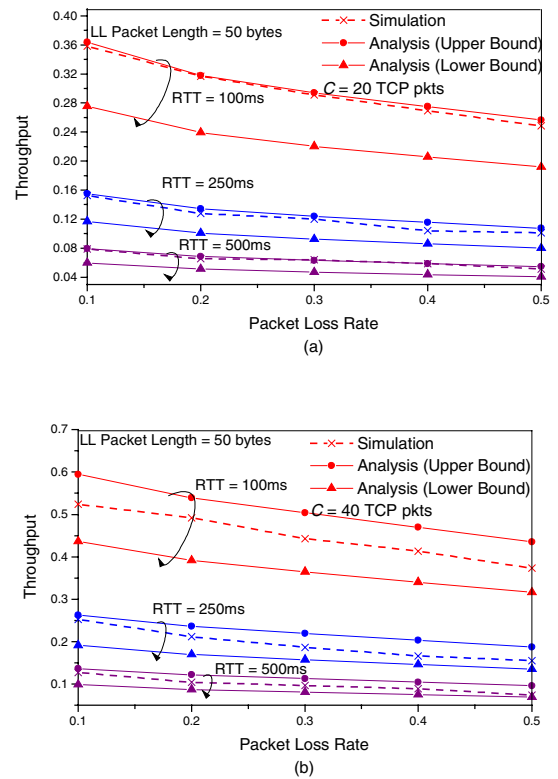


Fig. 7. Throughput vs Packet Loss Rate (with segmentation)

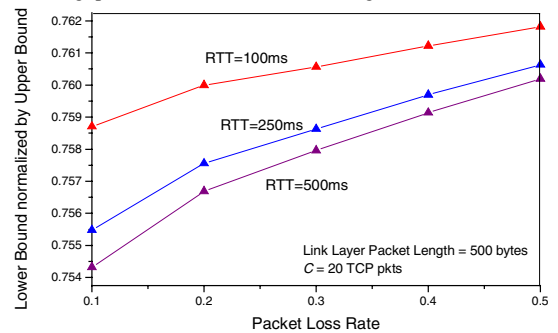


Fig. 8. The ratio of the lower bound to the upper bound

- [7] H. M. Chaskar, T. V. Lakshman, U. Madhow, "TCP over Wireless with Link Level Error Control Analysis and Design Methodology", IEEE Trans. Commu., 1999, vol. 7, no. 5, pp. 605-615.
- [8] M. Zorzi, A. Chockalingam, R. R. Rao, "Throughput Analysis of TCP on Channels with Memory", IEEE JSAC, vol. 18, no. 7, July 2000, pp. 1289-1300.
- [9] M. Zorzi, R. R. Rao, "Latency Probability of a Retransmission Scheme for Error Control on a Two-State Markov Channel", IEEE Trans. Comm., 1999, vol.47, no.10, pp. 1537-1548.
- [10] J. L. Mineweaser, J. S. Stadler, S. Tsao, M. Flanagan, "Improving TCP/IP Performance For The Land Mobile Satellite Channel", MILCOM, 2001.
- [11] L. Casone, G. Ciccarese, M. D. Blasi, L. Patrono, G. Tomasicchio, "An Efficient ARQ Protocol for a Mobile Geo-stationary Satellite Channel", IEEE GLOBECOM'01., vol.4, pp. 2692-2697, 2001.
- [12] F. Babich, E. Valentinuzzi, F. Vatta, "Performance of Hybrid ARQ schemes for the LEO satellite channel", IEEE GLOBECOM'01., vol.4, pp. 2709-2713, 2001.
- [13] A-F. Canton and T.Chahed, "End-to-End Reliability in UMTS : TCP over ARQ", 2001.