UW/EE Technical Report: Performance Modelling of TCP Enhancements in Terrestrial-Satellite Hybrid Networks

Jing Zhu[1], Sumit Roy[2], Jae Kim[3]

jing.z.zhu@intel.com

roy@ee.washington.edu

jae.h.kim@pss.boeing.com

1. Communications Technology Lab, Intel Corp., 2111 NE 25th Ave., JF3-206, Hillsboro, OR 97124

2. Univ. of Washington, Box 352500 Seattle, WA 98195-2500

3. Phantom Works, The Boeing Co., Box 3999 MC 3W-51, Seattle, WA 98124

Abstract

In this paper, we focus on the performance of TCP enhancements for a hybrid terrestrial-satellite network. While a large body of literature exists regarding modeling TCP performance for the wired Internet, and recently over a single-hop wireless link, the literature is very sparse on TCP analysis over a hybrid wired-wireless (multi-hop) path. We seek to make a contribution to this problem (where the wireless segment is a satellite uplink) by deriving analytical estimates of TCP throughput for two widely deployed approaches - *TCP splitting* and *E2E(End-to-End) TCP with link layer support* as a function of key parameters such as terrestrial/satellite propagation delay, segment loss rate and buffer size. Our analysis is supported by simulations; throughput comparisons indicate superiority of TCP splitting over E2E scheme in most cases. However, in situations where end-to-end delay is dominated

Contact Author: Sumit Roy, (Phone/FAX: (206)221-5261/543-3842, Email: roy@ee.washington.edu, Address: Dept. of Electrical Engineering, Univ. of Washington Box 352500, Seattle, WA 98195-2500)

by terrestrial portion and buffering is very limited at intermediate node, E2E achieves higher throughput than TCP splitting.

Keywords: satellite networks, TCP/IP, ARQ.

Abbreviations

SACK : Selective Acknowledgement

ACK : Acknowledgement

NAK : Negative Acknowledgement

FACK : Forward Acknowledgement

LEO : Low Earth Orbit

GEO : Geosynchronous Orbit

TCP : Transport Control Protocol

RLP : Radio Link Protocol

ARQ : Automatic Retransmission reQuest

FEC : Forward Error Correction

I. INTRODUCTION

The need for *global broadband access* to the Internet for airborne/seaborne nodes with high mobility has led to expansion of the terrestrial Internet backbone by incorporating satellite communication links. Examples include proprietary networks by Teledesic, GlobalStar Inc. (and others) to provision for new data services via terrestrial-satellite hybrid networks based on a constellation of LEO satellites [1]. TCP which continues to be the primary transport protocol, is well known to face new challenges in a satellite networking environment, including the long propagation delay (e.g. one-way delay is $10 \sim 100$ ms for LEO satellite and 250 ms for GEO satellite) and significant packet losses on the satellite link (e.g. for typical satellite links, average BER ranges from 10^{-5} to 10^{-8}, and higher - 10^{-2} to 10^{-6} - in land mobile satellite channels [10]). [2] demonstrated significant performance degradation of TCP in a lossy network with large bandwidth-delay product (BDP) (e.g. satellite) due to its limited loss-recovery capability. Since TCP's congestion control mechanism regards link layer losses (erroneously) as indicative of congestion, it invokes unnecessary rate control leading to low bandwidth utilization. Thus many enhancements have been proposed to improve TCP performance, which can be conveniently classified into three broad categories - *TCP Protocol Enhancements* (e.g. TCP-Peach [4][3], TCP-SACK [5], etc.), *TCP Splitting* (e.g. I-TCP [6], Skyx [7], etc.) and *End-to-End(E2E) TCP with link layer support* ([9], [10], etc.). TCP Protocol Enhancements preserve end-to-end semantics and do not require complicated configuration and control in the core network; however its main drawback is the need to replace current TCP protocol stack implementations at end-user devices with the new versions that can be cumbersome. On the other hand, both *TCP Splitting* and *E2E TCP with link layer support* do not require any modifications in TCP protocol stack at the end-systems and have found wide acceptance by industry (e.g. Skyx [7], Flash [8] etc.) in product deployment. Accordingly, in this work we focus on *analysis* of *TCP Splitting* and *E2E with link layer support* approaches.

TCP splitting uses a performance enhancing proxy at the satellite channel access node that divides the end-to-end TCP connection between a (terrestrial) source and (airborne) destination pair (see Fig. 1) into two (or possibly more) segments. On the satellite portion, advanced schemes are employed to combat wireless channel losses - usually some combination of enhanced link layer ARQ/FEC approaches or specialized TCP versions(SACK, FACK, etc.). This results in improved throughput without costly upgrades to the TCP stacks at the end systems and any system optimization to hide the impact of the link losses is therefore local to the satellite segment. Nevertheless, performance sensitivity issues arise due to the interaction among path segments and different layers for any particular solution. For example, in TCP Splitting, the intermediate node (the spoofer) sends back a *spoofing* ACK packet to the TCP sender immediately upon receiving a TCP data packet instead of waiting for the ACK from the final TCP destination. [11] studied the performance of TCP spoofing by simulation and showed the problem of data accumulating at the spoofer, potentially leading to an additional bottleneck. We also note that RFC 3135 [31] has identified many issues related to the TCP splitting approach, such as robustness and security. One of the well known problems of TCP splitting is that by breaking the end-to-end connection, a split TCP connection is no longer reliable or secure, and a failure of the satellite ground station may cause the sender to believe data has been successfully received when it has not.

The other alternative - E2E scheme with link layer support - makes packet loss completely transparent to TCP layer by using reliable link layer protocol such as selective repeat ARQ on the satellite portion. While this approach preserves original TCP end-to-end semantics and does not suffer from the security weaknesses of TCP splitting, it does potentially contribute a new problem - the interaction between TCP and link layer protocol, both of which offer reliable data transfer. This may impact end-to-end performance significantly due to the possibility for greater variability in (end-to-end) round trip time caused by link layer retransmissions. [20] demonstrated through simulation that using selective-repeat ARQ at the link layer rather than Stop-Wait or Go-Back-N, the problem of competitive retransmissions between TCP and link layer is much less serious than previously reported.

The primary significance of our work is *our contribution towards modelling of TCP performance* in the context of the relative lack of such (analytically inspired) results for hybrid networks. Of the few earlier studies, [12] 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. [13] and [14] considered the effect of forward error correction (FEC), and [15] 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 [16] [17], but they focused on the impact of burst errors in a fading channel while ignoring wireless propagation delay (and the resulting interaction with TCP congestion control algorithm) which is not feasible for TCP-over-satellite. [21] took segmentation at link layer into consideration and modelled TCP over ARQ using a Markov method; however, the propagation delay at the link layer was again neglected. [18] evaluated performance of hybrid ARQ in LEO satellite networks, but did not study TCP performance. [19] proposed an analytical model to evaluate the performance of TCP over Go-Back-N ARQ in UMTS environments. Although [19] took the wireless propagation delay into consideration, Go-Back-N is less effective than selective repeat ARQ (see [20]), which limits the application of the model proposed in [19].

In summary, there does not exist any reliable analytical estimate of TCP throughput for E2E with LL SR-ARQ or TCP Splitting in a lossy hybrid network - our work provides the first comprehensive analysis. Further, the analysis is validated by simulation with ns-2^{TM} simulator. Our main conclusions are that TCP splitting generally outperforms E2E scheme; however in the case where the end-to-end delay is dominated by terrestrial portion (and not the satellite link, such as in LEO network where the round trip time is 10ms) and buffer size is limited at intermediate node, E2E scheme is preferred. The only metric investigated in this paper is *throughput* and *delay* performance is not considered; this limits the utility of the analysis to data services such as email and FTP that are not very delay sensitive.

The paper is organized as follows. In Section 2, we describe the terrestrial-satellite hybrid satellite network scenario and introduce a theoretical system model as the basis of our analysis. Throughput expressions for E2E with LL SR-ARQ and TCP splitting are obtained in Section 3 and Section 4, respectively supported by numerical results by way of model validation. Section 5 presents some observations based on our results as well as model extensions by considering more realistic factors, such as fading channel, limited retransmission attempts and multiple connections. Section 6 concludes the paper.

II. System Model

Fig.1 shows a generic network model with terrestrial and satellite portions for both TCP splitting and E2E with link layer support. Generally, the bandwidth on the terrestrial portion is much larger than on the satellite portion so that the intermediate node (gateway) is a congestion point.

Therefore, provisioning of sufficient buffer space at the satellite gateway plays a key role in influencing TCP performance. We assume a bent-pipe satellite model which can be regarded as a lossy point-to-point link; thus no flow and congestion control is needed in principle on the satellite portion and should be avoided for optimizing overall system efficiency.

In TCP splitting, a connection is divided into two separated sub-connections at the intermediate node. A normal version of TCP (Reno) is used in the terrestrial portion while an improved link-layer protocol (ARQ, FEC, etc.) or some advanced version of TCP (SACK, FACK, etc.) is suggested for the satellite portion. In this paper, we assume a fully reliable selective repeat ARQ over the satellite link, where a data packet is not cleared from the send buffer until the arrival of corresponding acknowledgment.

A suitable reliable protocol (e.g. SR-ARQ) is used in the E2E scheme, but only at link layer. Further, they are completely transparent to TCP layer so that TCP end-to-end semantics is unchanged (see Fig.1). Note there exists a maximum limit on retransmission attempts at link layer of a real system. As is well known, TCP throughput is sensitive to loss; therefore, the retransmission limit should be sufficiently large to achieve very low residual segment loss rate. This was confirmed in [19] which also concluded that the price for this reduced residual loss rate is added latency; this was considered a worthwhile trade-off since without corrupted segments, TCP window will not be backed off (reduced by half when "congestion" losses occurs) that typically leads to throughput degradation. For this reason, we assume fully reliable SR-ARQ at the link layer.

Fig.2 shows a system model for our following theoretical analysis that defines the key system parameters listed below.

B: Buffer size of intermediate node (in units of TCP segments);

T_1: Round Trip Time (RTT) of terrestrial portion;

T_2: RTT of satellite portion;

μ: Transmission rate of satellite portion (TCP segments per second);

p: TCP segment loss rate of the satellite link.

Note that the link capacity on the terrestrial part is not specified as it is assumed to be significantly larger than the (average) wireless link capacity and its specific value does not impact our analysis. The above model was also used in [22] for modelling TCP performance in a network with high bandwidth-delay product and random loss. However [22] did not consider any enhancements such as link layer SR-ARQ or TCP splitting and only used end-to-end RTT without differentiating between the respective RTTs on the terrestrial and satellite segments. Intuitively, since the random loss on the satellite channel will lead to retransmissions, RTT variation on the satellite segment is expected to have a greater impact on the TCP throughput than that on the terrestrial part.

Like earlier works [22] [16] [28], the model proposed in this paper assumes a "constant" terrestrial RTT T_1, including all queuing, propagation and processing delays in the paths constituting the connection. The underlying basis for this assumption is that although the RTT in the terrestrial segment is time-varying, the variations are slow compared to that in the satellite portion - hence the quasi-static nature can be approximated by its local mean value during a simulation run (order of hundreds of seconds) without much impairment to the accuracy of the analysis.

The satellite RTT T_2 is also variable in principle; *changing network topology* and *routing* in MEO/LEO networks (it is, of course, constant in GEO networks) can lead to abrupt delay variation [1], which has a great impact on TCP transient performance - [29] provides a detailed model for this scenario. However, as shown in [30], the mean time between such abrupt delay changes can be several hundred seconds in (Teledesic) LEO satellite network, which is long enough for TCP to enter steady state. In this work, we thus only consider TCP performance during steady state where the satellite RTT may be reasonably modelled as constant.

III. End-to-End TCP with Link Layer SR-ARQ Support

The key assumptions of our model for end-to-end TCP with link layer SR-ARQ support are described next.

1) It was concluded in [23] that any link layer protocol (e.g. SR-ARQ) in a wireless link with large bandwidth-delay product can lead to significant reordering (out-of-order delivery) of packets on the link, leading to duplicate acknowledgments by the TCP receiver, which causes the sender to invoke fast retransmission and recovery. This can potentially degrade throughput; therefore in-order packet delivery is necessary for achieving high performance with TCP over

[1] The delay variation caused by satellite motion is slower relative to those caused by route changes.

SR-ARQ in a terrestrial-satellite network, and a link layer buffer is needed for reordering at the receiver. We assume sufficiently large receive buffer to avoid any buffer overflow at receiver.

2) Wireless channel losses are modelled as independent and identically distributed (i.i.d), which is reasonable for most fixed (static) satellite terminals. Even for a land mobile satellite channel characterized by correlated packet losses, the correlation can be dramatically reduced by using sufficient interleaving at physical layer. At any fading rate, results based on an i.i.d.loss model capture trends of TCP performance that are similar to that for correlated loss models.

3) For i.i.d. channel models, E2E RTT variations caused by retransmission are statistically independent; in such cases, the probability of unnecessary timeout occurrence due to RTT variance is typically negligible with the current Retransmission TimeOut estimator (RTO = $X + hY$, where X is an estimator of the current RTT, Y is a smoothed estimator of the mean deviation, and h is the weight [2] (currently set as 4).) in TCP protocol. Neglecting the impact of timeout and considering only congestion losses, allows us to assume that TCP remains in congestion avoidance in steady state, thereby simplifying throughput estimation considerably.

4) We assume only standard ACK scheme (no delayed ACKs), i.e., one TCP ACK is generated for each received TCP data packet and returned to TCP sender with no delay.

5) At link layer, retransmissions have higher priority than new packet arrivals; the latter are sent only when there are no retransmit packets in queue.

6) ACK/NAKs are used at link layer; for each received link layer packet, ACK is sent for success and NAK for failure.

7) Both TCP ACK packets and link layer ACK/NAK packets are assumed to be error-free. This is reasonable in most cases since their length is much smaller when compared with data packets. Furthermore, they constitute control traffic with higher priority so that more powerful forward error correction (FEC) schemes should be used to protect them from losses.

8) Link Layer (LL) SR-ARQ is assumed fully reliable such that a LL data packet will not be released until it is successfully acknowledged.

9) Saturation traffic is assumed such that the TCP source always has packets to send.

10) Compared with satellite RTT (SRTT), a packet transmission time $\frac{1}{\mu}$ is small enough to be ignored.

[2] The larger the weight h, the higher RTT variance the RTO algorithm can tolerate.

A. *TCP Window Transfer Time*

In the congestion avoidance phase, TCP window increases by one for successful ACK of *all* packets in current window. We define the duration between the arrival of ACK for the last packet in the previous window and that for the ACK for the last packet in the current window as the TCP window transfer time, denoted as $\tau(w)$ where w is current window size. This can be described as the sum of three components, i.e,

$$\tau(w) = T_1 + Q(w) + D(w), \tag{1}$$

where T_1 is fixed terrestrial RTT, $Q(w)$ is queuing delay, and $D(w)$ is the total transmission delay on the satellite portion. The total transmission delay for a packet is the duration from beginning of first transmission attempt to the arrival of TCP ACK for that packet. Fig.3 shows the sequence of events in a TCP window transfer.

Characterizing the variables $Q(w)$ and $D(w)$ via their pdf (probability density function) is exceedingly complex; instead, we will attempt a mean-value analysis wherever possible (resorting to conservative upper bounds at other times) that yields simpler closed-form relations and consequent insight as to how end-to-system performance depends on key system parameters.

We assume that both Link Layer (LL) and TCP segments have fixed lengths, and each TCP segment is segmented into S LL packets. If n successive TCP segments await transmission, nS LL packets reside in the buffer at the intermediate node after segmentation. A TCP segment is assumed successful only upon receipt of the ACK for the last LL packet constituting the TCP segment.

1) SR-ARQ Retransmission Delay $D(w)$: The *total transmission delay* is the duration from the beginning of transmission to the arrival of TCP ACK (corresponding to final LL ACK). For in-order link layer delivery to upper layers, the delay in correctly receiving all the previously sent LL packets must be considered. The probability mass function (pmf) of total transmission delay d normalized by T_2 for any reference packet on the satellite link with independent link layer packet loss rate r $(= 1 - (1-p)^{1/S})$ is given by the well-known geometric distribution

$$\mathrm{P}(d/T_2 = i) = r^{(i-1)}(1-r) \rightarrow \mathrm{P}(d = iT_2) = r^{(i-1)}(1-r). \tag{2}$$

Let $\tilde{d}(k)$ denote the total transmission delay for in-order delivery given that k LL packets with sequence number lower than the reference packet are in flight on the satellite link when the first transmission starts. It follows that since the delay for each packet is i.i.d with pmf given by Eq.(2), the distribution of $\tilde{d}(k)$ is given by the pmf of the *maximum* of k i.i.d. geometric random variables. Thus

$$\begin{aligned}
\mathrm{P}(\tilde{d}(k) = iT_2) &= [\mathrm{P}(d \leq iT_2)]^k - [\mathrm{P}(d \leq (i-1)T_2)]^k \\
&= [\sum_{j=1}^{i} \mathrm{P}(d = jT_2)]^k - [\sum_{j=1}^{i-1} \mathrm{P}(d = jT_2)]^k \\
&= (1 - r^i)^k - (1 - r^{(i-1)})^k. \qquad (3)
\end{aligned}$$

The mean of $\tilde{d}(k)$ can be shown to be well approximated by (after some tedious steps given in Appendix *A*)

$$E_k = \mathrm{E}(\tilde{d}(k)) \approx \frac{T_2}{1-r}(1 + \nu \ln(\frac{k+1}{2})), \quad \nu = -\frac{1-r}{\ln r} \qquad (4)$$

For satellite links, typically $k >> 1$, leading to

$$E_k \approx \frac{T_2}{1-r}(1 + \nu \ln(\frac{k}{2})). \qquad (5)$$

showing that $\mathrm{E}(\tilde{d}(k))$ is a logarithmic function of k.

Now clearly $k \leq \mu T_2$ (BDP of satellite link). Furthermore, k cannot exceed the buffer size B, as a copy of each unacknowledged in-flight LL packet is required in the buffer. In addition, the TCP window size w controls the total number of in-flight TCP segments; as a result,

$$k \leq \min(B, w, \mu T_2) S. \qquad (6)$$

From the above, the total transmission delay for a TCP segment $D(w)$ is upperbounded by

$$D(w) \leq \tilde{d}(\min(B, w, \mu T_2) S). \qquad (7)$$

with high probability, since $\tilde{d}(k)$ is a monotonic function of its argument.

The mean delay $D(w)$ is then bounded by

$$E(D(w)) \leq E(\tilde{d}(\min(B, w, \mu T_2) S) \approx \frac{T_2}{1-r}(1 + \nu \ln(\frac{\min(B, w, \mu T_2) S}{2})). \qquad (8)$$

2) Modelling Queuing Delay: At the link layer, a TCP segment will be segmented into S LL packets, implying an effective LL transmission rate of μS packets/sec. Next we consider the queuing delay at the sender's LL buffer (see Fig. 4) for a *new* TCP segment, defined as the duration from arrival to the first transmission of a LL packet.

With *assumptions* 5) $\sim$ 8), a reliable satellite LL SR-ARQ system can be described as a transmission pipe with bandwidth delay product $\mu T_2 S$ (see Fig.4). Since a transmitted LL packet will be removed from the pipe only when it is successfully acknowledged, it can be modelled as G/G/k *multi-server* queue where each of the $k = \mu T_2 S$ servers serve one LL packet, as shown in Fig.5. We introduce the following key notations:

$q_1(w)$:The number of queued LL packets in buffer that will be served with the rate $\mu S(1-r)$.

$q_2(w)$:The number of queued LL packets in buffer that will be served with the rate μS.

$q_3(w)$:The number of transmitted LL packets awaiting acknowledgement.

The service rate of queued LL packets depends on the current state of the pipeline and determines the queuing delay. If the pipeline is fully occupied, i.e., $q_3(w) = \mu T_2 S$, the rate of packet removal from the system is $\mu S(1-r)$, incorporating the success probability of $1-r$ for any transmission. If the pipeline is under-used, i.e., $q_3(w) < \mu T_2 S$, the new arrival can enter the pipeline immediately so that the service rate is approximately[3] μS.

The new TCP segment sees total $q_1(w)$ + $q_2(w)$ + $q_3(w)$ packets in queue on arrival, among which $q_1(w)$ + $q_2(w)$ packets are in the congestion window and must be served first before transmission of any corresponding LL packets. The queuing delay is then given by

$$Q(w) = \frac{q_1(w)}{\mu S(1-r)} + \frac{q_2(w)}{\mu S}. \tag{9}$$

Since the maximum queue size is min$(B, w)S$, the number of packets to be released at rate $\mu S(1-r)$ is bounded by

$$q_1(w) \leq (\min(B, w) - \mu T_2)S. \tag{10}$$

If the maximum queue length is less than the BDP, i.e. min$(B, w) < \mu T_2$, the pipeline does not reach capacity and all LL packets are served with the rate μS; i.e.,

$$q_1(w) = 0, \text{ if } (\min(B, w) < \mu T_2). \tag{11}$$

[3]More precisely, the input rate is μS only if the pipeline is *empty*, i.e., $q_3(w) = 0$, and should be in the range $(\mu S(1-r), \mu S)$ for $0 < q_3(w) < \mu T_2 S$. Here, we simply employ the upper-bound as an approximation which is reasonable when the packet error rate r is small.

Hence,

$$E(q_1(w)) \leq [\min(B, w) - \mu T_2]^+ S, \tag{12}$$

where

$$[x]^+ = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} . \tag{13}$$

When the transmission pipeline is under-utilized, i.e. $q_3(w) < \mu T_2 S$, the packets in the queue can be served continuously. Therefore, all $q_1(w) + q_2(w)$ packets in queue arrive at sink almost at the same time as the new packet within the same burst. Let $L(w)$ denote the *maximum* burst length and assume that i) a burst length is uniformly distributed on the range [1, $L(w)$] and ii) that a reference packet is uniformly positioned in the burst. Then,

$$E(q_2(w)) \leq E(q_1(w) + q_2(w)) = \frac{1}{L(w)} \sum_{x=1}^{L(w)} (\frac{1}{x} \sum_{i=1}^{x} (i-1)) = \frac{L(w)-1}{4}. \tag{14}$$

Note that the main reason for burst arrival is the requirement for in-order delivery; a link layer packet arriving 'earlier' at the receiver must wait for the slower packets. Over error free links, TCP segments corresponding to a transmission window arrive continuously at the receiver. Continuous transmission of a TCP window is segmented by link layer retransmissions due to loss, since a TCP segment is only delivered (to the application) when all the link layer frames for that TCP segment as well as previously sent packets (due to in-order delivery policy) are received successfully. Once the retransmitted packet is received correctly, all subsequent successfully received TCP segments are up delivered to the TCP receiver as a burst. Consequently, TCP ACK packets are generated in bursts, and so are TCP data packets.

We estimate the maximum burst length in TCP segments, which is then scaled by S to obtain the maximum LL burst length $L(w)$. Consider two TCP segments sent with a separation interval equal to one satellite RTT T_2. Let x_1 and x_2 denote the random variable for their respective transmission delays, each given by the maximum of S (mutually indep.) i.i.d. geometric random variable[4] (Eq.(3)). The probability of receiving the two TCP segments out of order is therefore

$$\begin{aligned} & \mathrm{P}(x_2 - x_1 > 1) \\ = \; & \mathrm{P}(x_2 - x_1 = \{2, 3, 4, ...\}) \end{aligned}$$

[4] The transmission delay of a TCP segment is determined by the corresponding link layer packet with maximum delay.

$$
\begin{aligned}
&= \sum_{x_1=1}^{\infty}[((1-r^{x_1})^S-(1-r^{x_1-1})^S)\sum_{x_2=x_1+2}^{\infty}((1-r^{x_2})^S-(1-r^{x_2-1})^S)] \\
&= \sum_{x_1=1}^{\infty}[((1-r^{x_1})^S-(1-r^{x_1-1})^S)(1-(1-r^{x_1+1})^S)], \quad (r=1-(1-p)^{1/S}) \qquad (15)
\end{aligned}
$$

The function

$$f(S,x_1)=((1-r^{x_1})^S-(1-r^{x_1-1})^S)(1-(1-r^{x_1+1})^S),\ (x_1\geq 1, S\geq 1, r=1-(1-p)^{1/S})$$

can be shown to be a monotonically decreasing function of S for any $p < 0.5$. Consequently, Eq.(15) ($=\sum_{x_1=1}^{\infty} f(S,x_1)$) is also a monotonically decreasing function of S. An intuitive explanation is that larger S implies smaller link layer packet size for fixed TCP segment size, which increases transmission reliability, therefore reducing the probability of out-of-order delivery.

If $S=1$, Eq.(15) simplifies to

$$
\begin{aligned}
& \mathrm{P}(x_2-x_1>1)\mid_{S=1} \\
=& \sum_{x_1=1}^{\infty}[((1-p^{x_1})-(1-p^{x_1-1}))(1-(1-p^{x_1+1}))] \\
=& \sum_{x_1=1}^{\infty}[(p^{x_1-1}-p^{x_1})p^{x_1+1}] \\
=& \sum_{x_1=1}^{\infty}[(1-p)p^{2x_1}] \\
=& p^2/(1+p). \qquad (16)
\end{aligned}
$$

which thus gives an upper-bound for cases $S\geq 1$. In the following simulation, we only consider the case $S=1$.

From the above, it follows that for any reasonable scenario ($p\sim 10^{-1}$), two packets sent with separation interval longer than ONE satellite RTT are received out of order with sufficiently low probability ($\sim 1\%$). The out-of-order delivery probability, however, increases with segment loss rate p, e.g., about 16.7% for $p=0.5$. For such high loss rate, the transmission delay $E(D(w))$ will dominate in calculating the average TCP window transfer time $E(\tau(w))$, and therefore the impact of underestimating $L(w)$ by assuming that any two TCP packets with the separation longer than ONE satellite RTT will be received in order is small[5], especially when $\mu T_2 >> 1$,

[5]For example, if $\mu T_2 = 10$, $p=0.5$, $S=1$, and $min(B,w) > 2\mu T_2$, we have $E(D(w)) = 4.32T_2$ using Eq.(8). Compared to using one satellite RTT, the difference in the queueing delay estimate with Eq.(19) by using "$L(w) = \min(B, 2\mu T_2, w)S$" is $0.25T_2$ ($\approx 6\% E(D(w))$), and the out-of-order delivery probability given by $P(x_2-x_1 > 2)\mid_{S=1}= p^3/(1+p) \approx 8\%$).

which is usually true in broadband satellite networks.

The maximum number of LL packets transmitted in duration T_2 is $\mu S T_2$. Furthermore, any burst can never be larger than TCP window wS. As a result, the maximum burst length is given by $\min(B, \mu T_2, w)S$, i.e.,

$$L(w) = \min(B, \mu T_2, w)S. \tag{17}$$

For satellite links, typically $\min(B, w, T_2\mu)S >> 1$, leading to

$$E(q_2(w)) \leq \frac{L(w)-1}{4} = \frac{\min(B, w, T_2\mu)S - 1}{4} \approx \frac{\min(B, w, T_2\mu)S}{4}. \tag{18}$$

Insert Eq.(12) and (18) into Eq.(9) to get

$$E(Q(w)) = \frac{E(q_1(w))}{\mu S(1-r)} + \frac{E(q_2(w))}{\mu S} \leq \frac{[\min(B, w) - \mu T_2]^+}{\mu(1-r)} + \frac{\min(B, w, T_2\mu)}{4\mu}. \tag{19}$$

Finally , we estimate the average TCP window transfer time $E(\tau(w))$ by the upper-bound

$$\begin{aligned} E(\tau(w)) &= T_1 + E(Q(w)) + E(D(w)) \\ &\leq T_1 + \frac{[\min(B, w) - \mu T_2]^+}{\mu(1-r)} + \frac{\theta}{4\mu} + \frac{T_2}{1-r}(\nu \ln(\frac{\theta S}{2}) + 1) \end{aligned} \tag{20}$$

$$(\theta = \min(B, w, \mu T_2),\ \nu = \frac{1-r}{\ln(1/r)}).$$

B. Congestion Analysis

In this section we study the problem of buffer overflow at the intermediate node; we ignore the terrestrial propagation delay (i.e. $T_1 = 0$) at first so that packets from TCP source arrive at the intermediate node instantaneously. We define the notations used in the following analysis.

$b_R(t)$: Number of packets waiting for reordering in receive buffer at time t;

$b_T(t)$: Number of packets in send buffer at time t;

$w(t)$: TCP congestion window size at time t;

Note that $b_R(t)$ and $b_T(t)$ are link layer packets measured in units of TCP segment size. Obviously, overflow occurs when $b_T(t) > B$.

For any time t_0, ACK packets already in flight will arrive at the sender before $t_0 + \frac{T_2}{2}$. Then, copies of all packets counted in $b_R(t_0)$ will be cleared from the send buffer. Packets arriving at the receiver during this period $(t_0, t_0 + \frac{T_2}{2})$ still have their copies in the send buffer, and will be counted in $b_T(t_0 + \frac{T_2}{2})$. As a result, $b_T(t_0 + \frac{T_2}{2}) + b_R(t_0)$ indicates the total number of unacknowledged packets in flight, sender buffer and receiver buffer at time $t_0 + \frac{T_2}{2}$ that must equal the congestion window size, i.e.,

$$w(t) = b_T(t) + b_R(t - \frac{T_2}{2}). \tag{21}$$

Since $w(t)$ is a constant for the duration of a window transfer period, which is at least one E2E RTT long ($> T_1 + T_2$), it is reasonable to assume that the TCP window size at t and $t - \frac{T_2}{2}$ are the same (the difference will be no more than 1 when TCP is in congestion avoidance stage). It implies that $b_T(t)$ reaches a local maximum when $b_R(t - \frac{T_2}{2})$ reaches a local minimum. Using $\Phi^{(i)}$ and $\Psi^{(i)}$ to denote the maximum (minimum) queue length of the send(receive) buffer during the ith TCP window transfer with size $W^{(i)}$, we have from Eq.(21) that

$$W^{(i)} = \Phi^{(i)} + \Psi^{(i)}, \tag{22}$$

with

$$\Phi^{(i)} \geq 0 \quad \text{and} \quad \Psi^{(i)} \geq 0. \tag{23}$$

It is easily seen from Eq.(22) that buffer overflow will never happen if $W^{(i)} \leq B$. Otherwise, single or multiple losses may occur; let n denote the number of such losses. We can model $\{W^{(i)}, 1 \leq i \leq \infty\}$ as a Markovian process with transition probability given as follows:

$$\begin{cases} P\{W^{(i+1)} = x+1 | W^{(i)} = x\} = 1 & (x \leq B) \\ P\{W^{(i+1)} = x+1 | W^{(i)} = x\} = P\{\Psi^{(i)} \geq x - B | W^{(i)} = x\} & (x > B) \\ P\{W^{(i+1)} = \frac{x}{2^n} | W^{(i)} = x\} = P\{\Psi^{(i)} = x - B - n | W^{(i)} = x\} & (x > B) \end{cases}. \tag{24}$$

The first two equations are for window increase (linear increase), and the third equation is for window deflation (exponential decrease).

Accurate solution of the above Markovian process depends on the conditional probability distribution of $\Psi^{(i)}$, which is very difficult to solve. We thus approximate $\Psi^{(i)}$ with the following distribution

$$P\{x\} = \begin{cases} 1, & x = 0 \\ 0, & \text{else} \end{cases}, \tag{25}$$

which means the minimum queue length in receiver buffer is zero at every window transfer. Consequently, Eq.(24) simplifies to

$$\begin{cases} P\{W^{(i+1)} = x+1 | W^{(i)} = x\} = 1, & (x \le B) \\ P\{W^{(i+1)} = \frac{x}{2} | W^{(i)} = x\} = 1, & (x = B+1) \end{cases} . \tag{26}$$

Only one packet is dropped at overflow when the maximum TCP window size is $B+1$. After overflow, TCP window is reduced by half and the TCP window size oscillates between $B+1$ and $\frac{B+1}{2}$. We use this simplification (Eq.(26)) to estimate the average throughput.

To find the maximum TCP window w_{max} and its transfer time taking terrestrial propagation delay into consideration, we note that $w_{max} \ \ge \ B+1 > B$; hence

$$\min(B, w_{max}, \mu T_2) = \min(B, \mu T_2) \tag{27}$$

From Eq.(20), we have the average transfer time of the maximum window

$$E(\tau(w_{max})) \ = \ T_1 + T_2\mathcal{T}, \tag{28}$$

where

$$\mathcal{T} = \begin{cases} \frac{B}{\mu T_2(1-r)} + \frac{1}{(1-r)}(\nu \ln(\frac{\mu T_2 S}{2})) + \frac{1}{4}, & (B > \mu T_2) \\ \frac{B}{4\mu T_2} + \frac{1}{(1-r)}(\nu \ln(\frac{BS}{2}) + 1), & (B \le \mu T_2) \end{cases} . \tag{29}$$

To obtain the maximum window size for $T_1 > 0$, we introduce a new concept - virtual transfer time for any partial number of TCP segments within a window size w , denoted as $\tau'(x)$ where x is the number of packets. Given the TCP window size w and the window transfer time $\tau(w)$, we define $\tau'(x)$ as

$$\tau'(x) = \tau(w)\frac{x}{w}. \tag{30}$$

Let $\lambda(w_{max})$ be the average throughput in the transfer period of the maximum TCP window. The average virtual transfer time for $B+1$ packets is given by

$$E(\tau'(B+1)) = \frac{B+1}{w_{max}/E(\tau(w_{max}))} = \frac{B+1}{\lambda(w_{max})}. \tag{31}$$

By definition, we have

$$\lambda(w_{max}) = \frac{w_{max}}{E(\tau(w_{max}))} = \frac{w_{max}}{T_1 + T_2\mathcal{T}} \ge \frac{B+1}{T_1 + T_2\mathcal{T}}. \tag{32}$$

Intuitively, the throughput $\lambda(w_{max})$ during the maximum window transfer should decrease as the terrestrial RTT (T_1) increases, i.e., it is upper bounded by value at $T_1 = 0$:

$$\lambda(w_{max}) \leq \frac{B+1}{T_2 \mathcal{T}}. \tag{33}$$

Combining Eq.(31) $\sim$ (33) results in

$$T_2 \mathcal{T} < E(\tau'(B+1)) < T_1 + T_2 \mathcal{T}. \tag{34}$$

For approximation, we choose the value $T_1/2 + T_2\mathcal{T}$ as the estimate of the average virtual transfer time $E(\tau'(B+1))$. Hence, the average throughput in the maximum TCP window transfer is approximated by

$$\lambda(w_{max}) \approx \frac{B+1}{\frac{T_1}{2} + T_2 \mathcal{T}}, \tag{35}$$

Combining Eq.(28) and Eq. (35), we obtain

$$w_{max} = \lambda(w_{max}) E(\tau(w_{max})) = (B+1) \frac{T_1 + T_2 \mathcal{T}}{\frac{T_1}{2} + T_2 \mathcal{T}} = (B+1) \frac{2}{1+\rho}, \tag{36}$$

where ρ is defined as

$$\rho = \frac{T_2 \mathcal{T}}{T_1 + T_2 \mathcal{T}}. \tag{37}$$

Note that $\rho \leq 1$; when $T_2 >> T_1$ typically, $\rho \approx 1$.

The average throughput is computed as follows

$$\lambda = \frac{\frac{3}{8} w_{max}(w_{max}+2)}{\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w))}. \tag{38}$$

Defining $\beta = \frac{B}{\mu T_2}$, we present only the final results; for details please see Appendix *B*.

$$\frac{\lambda}{\mu} = \frac{\frac{3}{2}\beta(\frac{1-r}{1+\rho})}{\frac{T_1(1-r)}{T_2} + A_1 \nu \ln(\min(1,\beta)\mu T_2) + A_2 \frac{1-r}{4} + A_3 + \nu \ln(S/2)}, \tag{39}$$

where A_1, A_2, and A_3 are given by

$$\text{Case } (\beta \geq 1+\rho): \quad \begin{cases} A_1 = 1 \\ A_2 = 1 \\ A_3 = (\frac{1+\rho-\frac{1}{2}\rho^2}{1+\rho})\beta \end{cases}$$

$$\text{Case } (1 < \beta < 1+\rho): \quad \begin{cases} A_1 = 2 - \frac{(1+\rho)}{\beta} \\ A_2 = 2 - \frac{(1+\rho)}{2\beta} - \frac{\beta}{2(1+\rho)} \\ A_3 = 2\beta - 1 + \frac{(1+\rho)}{2\beta} - \frac{\beta(1+\rho)}{2} + \frac{(1+\rho)}{B}\nu \ln(\frac{(\mu T_2-1)!}{[\frac{B}{1+\rho}-1]!}) \end{cases} \tag{40}$$

$$\text{Case } (\beta \leq 1): \quad \begin{cases} A_1 = 1 - \rho \\ A_2 = (\frac{1+\rho-\frac{1}{2}\rho^2}{1+\rho})\beta \\ A_3 = 1 + \frac{(1+\rho)}{B}\nu \ln(\frac{(B-1)!}{[\frac{B}{1+\rho}-1]!}) \end{cases} .$$

C. Numerical Results and Discussion

The ns2 simulator was used to obtain results to validate the model. A 1 Mbps satellite link is assumed that drops TCP segments independently; the terrestrial bandwidth is set at 100 Mbps. The TCP segment length is fixed at 500 bytes (4000 bits) and TCP receiver window size is set large enough to eliminate its effect on throughput. The link layer packet length is also fixed at 500 bytes, leading to $S = 1$ and $p = r$. We generate an i.i.d. lossy channel by using Bernoulli random variable with the loss probability p. The segment loss rate used in simulation ranges from 0.1 to 0.5, corresponding to $[10^{-5}, 10^{-4}]$ in terms of BER (Bit Error Rate), i.e. BER $= 1-(1-p)^{\frac{1}{4000}}$.

We investigate the impact of B, p, T_1, and T_2 on TCP throughput. For simplicity, we only consider the case without segmentation (worst case scenario). A large variety of configurations are used in order to validate the predicted value from Eq. (39); Figs. 6 - 9 show that our analysis matches the simulation results well.

Fig.6 shows the effect of segment loss rate p for different satellite RTTs (i.e. 100ms, 250ms, and 500ms). Figs. 7,8 demonstrate the effect of satellite round trip T_2 and terrestrial round trip time T_1, respectively for $p = 0.1, 0.3$. As we can see, increasing T_2 results in much faster degradation of the throughput than increasing T_1 as can be anticipated since retransmissions on the satellite portion are more costly. Fig.9 illustrates the effect of buffer size, showing a *logarithmic* relation between throughput and buffer size. T_1 and T_2 are set equal to 100ms. for two values $p = (0.1, 0.3)$.

IV. TCP SPLITTING

In this section, we will study the performance of TCP splitting. In TCP splitting, TCP source is 'spoofed' by ACKs generated by intermediate node for packets that have not yet reached the destination. These ACK packets carry the receiver window advertisement (RWA), which indicates the remaining buffer size at the gateway TCP receiver. Note that in 'normal' TCP operation, the sender's rate is essentially determined by the congestion window (since receive buffer is assumed to be large). However, in our case, buffer overflow at gateway receiver is prevented by use of RWA which in turn controls the rate of the TCP sender. Assuming that the terrestrial link is loss-free, TCP sender's rate will be dominated by the remaining buffer space at the receiver (although the sender's congestion window will continue to grow beyond the RWA value).

Let $x(t)$ denote the number of packets sent by TCP sender in one terrestrial RTT (equivalently, called the TCP window size in the round), where t is the time when the burst arrives at the gateway. Since the terrestrial bandwidth is much higher than the satellite bandwidth and TCP segments arrive at the gateway in bursts, it is reasonable to assume that $B - q(t) - x(t)$ is the remaining buffer size after the arrival of the burst, given the burst length $x(t)$ and queue length $q(t)$ respectively. The RWA in the ACK for the ith packet in the burst is set to $B - q(t) - i$ $(1 \leq i \leq x(t))$. As these ACKs are received by TCP sender, the TCP window size decreases from $B - q(t) - 1$ to $B - q(t) - x(t)$. Since the rate of ACK generation equals the rate of transmission of TCP segments (as the terrestrial portion is assumed loss free), it follows that one TCP segment is sent out per received ACK. The TCP sender stops transmitting when the number of TCP segments sent reaches the TCP window size which is determined by the current RWA. As illustrated in Fig.10, the dashed-dotted line shows the RWA while the solid line indicates the number of TCP segments sent as a function of in-burst ACK index[6] that goes from 1 to $x(t)$. It is

[6] The x-axis in Fig.10 may be interpreted as a time scale, i.e. the time TCP sender receives each of the in-burst ACKs; therefore the slope of the solid line actually equals TCP transmission rate. After receiving all $x(t)$ ACKs, TCP sender will continue sending segments till reaching the latest TCP window size $B - q(t) - x(t)$ if $x(t) < \frac{1}{2}(B - q(t))$.

clearly seen from Fig.10 that the number of sent TCP segments for the next burst after terrestrial round trip time T_1 is given by

$$x(t+T_1) = \begin{cases} B - q(t) - x(t), & x(t) < \frac{1}{2}(B - q(t)) \\ \frac{1}{2}(B - q(t)), & x(t) \geq \frac{1}{2}(B - q(t)) \end{cases} . \tag{41}$$

Average over time to yield the time-averaged burst arrival length:

$$\overline{x} = \frac{B - \overline{q}}{2}, \tag{42}$$

where $\overline{q}$ is the average queue length.

In the above, we have assumed that $< x(t+T_1) >=< x(t) >= \overline{x}$ since T_1 is much less than the averaging interval[7]. The average TCP traffic arrival rate at the buffer of intermediate node is then well-approximated by

$$\overline{\nu} = \frac{\overline{x}}{T_1} = \frac{B - \overline{q}}{2T_1}. \tag{43}$$

On the satellite portion, fully reliable SR-ARQ at the link layer implies that each TCP segment is retransmitted till success. Given the segment loss rate p, the average persistence time $\overline{T_p}$ for a packet on the satellite portion is given by

$$\overline{T_p} = \frac{T_2}{1-p}. \tag{44}$$

Using Little's theorem, we have

$$\overline{q} = \overline{T_p}\overline{\nu}. \tag{45}$$

Substituting Eq.(43),(44) into Eq.(45), we get

$$\overline{q} = \frac{B - \overline{q}}{2T_1}\frac{T_2}{1-p}. \tag{46}$$

Thus, the average queue length is

$$\overline{q} = \frac{BT_2}{T_2 + 2(1-p)T_1}, \tag{47}$$

and

$$\overline{\nu} = \frac{B(1-p)}{T_2 + 2(1-p)T_1}. \tag{48}$$

[7] $< . >$ indicates the time averaging operation.

In steady state, the average TCP traffic arrival rate $\overline{\nu}$ at the buffer of intermediate node should be equal to the average throughput λ.

$$\lambda = \overline{\nu} = \frac{B(1-p)}{T_2 + 2(1-p)T_1}. \tag{49}$$

Normalized by the maximum throughput of $\mu(1-p)$, Eq.(49) reduces to

$$\frac{\lambda}{\mu(1-p)} = \min(1, \frac{\beta}{(1+2\alpha)}), \qquad (\beta = \frac{B}{\mu T_2}, \alpha = \frac{T_1}{T_2/(1-p)}). \tag{50}$$

Fig.11 studies two scenarios with different satellite round trip time (0.01s and 0.5s) along with analytical results from Eq.(50) that show a good match to simulation. The saturation throughput is 0.9, which is also the maximum throughput at segment loss rate of 0.1. The results also imply that maximum throughput is linearly related to buffer size.

Fig.12 shows the effect of segment loss rate on throughput. It is clearly shown that throughput degrades with segment loss rate increasing. Furthermore, the throughput for longer satellite RTT is more sensitive to segment loss rate.

With Eq.(50) and Eq.(39), we can theoretically compare TCP splitting with E2E with LL SR-ARQ in terms of throughput. Fig.13 compares results of two schemes at different α which can be interpreted as the average persistence time of a packet on the terrestrial portion (T_1) to that on the satellite portion ($T_2/(1-p)$). It is clearly seen that generally TCP splitting outperforms E2E scheme. Nevertheless, at a very high ratio (say $\alpha = 10$) with very limited buffer size (say $\frac{\beta}{1+2\alpha} < 0.4$), E2E scheme performs better than TCP splitting-the main reason being that in TCP splitting, TCP window is always limited by the remaining buffer size at the intermediate node (congestion node) due to RWA (receiver window advertisement) while E2E scheme has no such problem, therefore the very limited buffer size of intermediate node has greater impact on TCP splitting than E2E scheme; On the other hand, the main benefit of using TCP splitting compared with E2E with LL SR-ARQ is that the added latency caused by retransmission on the satellite portion has slight impact on the performance of the terrestrial portion. However this advantage vanishes as the terrestrial portion becomes more and more dominant in the end-to-end delay (i.e. α increases).

V. Model Extensions and Discussions

A. Observations

The results obtained support the following summary observations regarding TCP in terrestrial-satellite hybrid networks.

i) In E2E with LL SR-ARQ support, "$B > (1+\rho)\mu T_2$" is necessary for achieving high end-to-end TCP performance (see Fig.13). Generally, in a terrestrial-satellite hybrid network, the end-to-end delay is dominated by the satellite portion (i.e. $T_2 >> T_1$) and the satellite link has large bandwidth-delay product ($\mu T_2 >> 1$), leading to $\rho \approx 1$. Consequently, to achieve high end-to-end TCP throughput by using link layer SR-ARQ to resist wireless loss, the buffer size must satisfy

$$B > 2\mu T_2, \tag{51}$$

a very familiar result in the context of achieving high utilization with TCP.

ii) With $T_2 >> T_1$ and $B > 2\mu T_2$, the normalized throughput for E2E with LL SR-ARQ given by Eq.(39) is

$$\lambda/\mu = \frac{\frac{3}{4}B(1-p)}{\frac{3}{4}B + T_2\mu(\nu \ln(\frac{\mu T_2 S}{2}) + \frac{1}{4}(1-p))}. \tag{52}$$

Assuming $\nu \ln(\frac{\mu T_2 S}{2}) + \frac{1}{4}(1-p) \approx \nu \ln(\mu T_2)$ given a large enough μT_2 (i.e., $\ln(\mu T_2) >> \ln(\frac{S}{2})$), Eq.(52) simplifies to

$$\lambda/\mu \approx \frac{B}{B + \frac{4}{3}\nu[\mu T_2 \ln(\mu T_2)]}(1-p). \tag{53}$$

It is well known that in a lossless (wireline) link, the buffering required at the bottleneck link must scale linearly with the bandwidth-delay product, while the key lesson from Eq.(53) is that buffering required to hide the impact of link layer retransmission from TCP over a lossy (wireless) link with LL SR-ARQ support must scale as "$x \ln x$" where x denotes the bandwidth-delay product.

iii) In TCP splitting, $T_2 >> T_1$ leads to $\alpha \approx 0$, thus reducing Eq.(50) to

$$\frac{\lambda}{\mu(1-p)} \approx \min(1, \frac{B}{\mu T_2}), \tag{54}$$

which is exactly the throughput of SR-ARQ over a link with BDP of μT_2 and average segment loss rate of p. Eq.(54) also implies that TCP splitting only requires a linearly increased buffering with the bandwidth-delay product. This is the main reason why TCP splitting outperforms E2E with LL SR-ARQ support.

B. Partially Reliable SR-ARQ due to Limited Retransmission Attempts

Our model for E2E TCP with LL SR-ARQ support assumed *fully reliable SR-ARQ with unlimited retransmission attempts*; this is now relaxed to partially reliable SR-ARQ with *fixed* maximum retransmission number. As we see from Fig.14, sufficiently large maximum retransmission number leads to high throughput in general that remains almost unchanged with further increase in the maximum retransmission number. Accordingly, our model of infinite retransmissions (fully reliable SR-ARQ) provides a good approximation in such cases.

C. Correlated Fading Channel

We next extend our model to include correlated fading. A widely accepted loss model for correlated fading channels is two-state (i.e. *good* and *bad*) Markov model. For simplicity, we assume that the bit error rates in good(bad) states are $0(1)$ respectively, corresponding to a simplified scenario that nevertheless suffices to expose the key issues. Since the duration of each state is exponentially distributed, two parameters - mean duration of bad state and bad state time-sharing parameter (denoted as as m and X, respectively) can fully characterize a two-state Markov model. The mean duration of good state is given by $m\frac{1-X}{X}$; accordingly, the fading rate (denoted as f) can be defined as

$$f = \frac{X}{m}. \tag{55}$$

We do not attempt a detailed model for analysis of the correlated channel and instead seek a simpler one that captures the essential effects. Generally, the average segment loss rate p increases with the fading speed [24]. Let T_p be the persistence time of a packet on satellite link layer, which is defined as the duration from beginning of LL packet transmission to the arrival of LL ACK for the packet ($E[T_p] = \frac{T_2}{1-p}$, for i.i.d. channel). For the correlated fading channel, we have

$$p > X \text{ and } E[T_p] > \frac{T_2}{1-p}. \tag{56}$$

For E2E TCP with link layer SR-ARQ support, the essential effect introduced by correlated fading is on the total transmission delay $D(w)$ via the two components - T_p and reordering delay (denoted as T_r). Intuitively, the correlation among in-flight packets should help reduce the reordering delay at receiver. Let us consider an extreme case where in-flight packets are either *all* correctly received or lost; in either case, all packets are delivered in-order leading to zero reordering delay. A lower bound of $E[D(w)]$ then follows

$$E[D(w)] = E[T_p] + E[T_r] > E[T_p] > \frac{T_2}{1-p} > \frac{T_2}{1-X}. \tag{57}$$

Then, we can derive upper bounds for the normalized throughput in E2E by using Eq.(39) with $\nu = 0$ and $p = X$ and similarly, one for TCP splitting by using Eq.(50) with $p = X$. Note however that the assumptions of no TCP timeouts is no longer valid for a correlated channel, therefore the upper bound from Eq.(39) is only an optimistic. The detailed discussion of modeling TCP over a correlated channel is out of the scope of this paper; see [21] for a detailed analytical model.

Fig.15 shows that TCP splitting is more robust to performance degradation caused by increased fading rate than TCP over SR-ARQ. Note that for TCP over SR-ARQ, we also include the analytical result based on i.i.d. channel model (i.e. $\nu = -(1-p)/\ln(p)$) with $p = X$. As expected, it provides a good approximation to the lower bound. We note that the accuracy of our analytical model can be improved by using a more precise value of average segment loss rate [24] instead of X. Also, a more accurate $E[T_p]$ derived from a Markov analysis as in [25] may be helpful for further improvements.

D. Multiple Connections

Up to this point, attention was restricted to a single TCP connection as in [16]. When multiple connections share the bottle-neck link, fair queuing and/or appropriate buffer management can be used to provide isolation among different connections, as proposed in [26], [27]. Thus, given the resources (bandwidth and the buffer size) allocated to various TCP flows at the bottleneck, the analytical approach presented here enables the estimation of achievable throughput. Even if a simple FIFO buffer is used at the bottleneck without isolation, our model can still provide a reasonable estimate of throughput.

We studied the following scenario - multiple TCP *Reno* connections with the same terrestrial and satellite paths sharing a common bottleneck with a simple FIFO queue. The results for total throughput were obtained for both *TCP over SR-ARQ* and *TCP Splitting*, as shown in Fig.16. The main observations are:

1) TCP over SR-ARQ: The aggregate throughput increases with the total number of connections due to fewer in-flight packets per connection. Since only packets from the same connection require in-order delivery, the re-ordering delay at receiver is dramatically reduced. Our model provides a conservative lower bound for the performance with multiple connections.

In a TCP over SR-ARQ system, the well known behaviour of synchronized TCP window evolution for the case with multiple connections does not exist. As we have mentioned earlier, due to the in-order delivery policy, a link layer packet arriving 'earlier' at the receiver must wait for the slower packets. Consequently, the ACK packets are generated in bursts, and so are the TCP data packets. Therefore, the packets from a TCP connection tend to stay together as a burst instead of being interleaved with packets from other TCP connections. When buffer overflow occurs, the discarded packets thus may come from only *some* of the active TCP connections. Fig.17 shows the traces of TCP window variation for two TCP connections sharing a satellite link, obtained from ns2 simulation. Clearly, at point A both TCP 1 and TCP 2 lost one packet each, while at point B all lost packets are from TCP 2.

2) TCP Splitting: The number of connections has much less effect on the total throughput (see Fig.16b) since the reordering delay on the satellite link has little impact on TCP splitting due to the separation. However, the total TCP window size of all connections is now limited by the *advertised receive window* and hence the number of connections cannot be too large. As is known, TCP fast retransmission/recovery scheme is triggered by triple duplicate ACK packets, which requires a TCP window of at least 4 packets. Let N be the total number of connections, then we must have

$$\frac{B}{N} \geq 4. \tag{58}$$

The total throughput for the case with $B = 30$ and $N = 10$ is approximately zero in our simulation, conforming with the above.

In summary, TCP Splitting does not scale well as the number of sharing connections increasing. Furthermore, for E2E TCP over SR-ARQ, the system efficiency can be significantly improved by allowing more TCP connections. Therefore, although most of current satellite gateway products are based on TCP splitting, E2E TCP with reliable link layer protocol (e.g. SR-ARQ) is a potential alternative in a large scale broadband satellite IP network.

VI. Conclusion

In this paper, we focused on TCP performance modeling in terrestrial-satellite hybrid network, where the propagation delay on the terrestrial or satellite portion must be considered. Two prevailing approaches: E2E TCP with link layer SR-ARQ support and TCP splitting, were studied and compared. Analytical estimates for TCP throughput were derived. Simulation results with a large variety of realistic parameter settings were used to prove the validity of the analysis.

For a single connection, our performance comparison establishes the case for using TCP splitting vis-a-vis E2E TCP with link layer support in a long satellite round trip time (compared with terrestrial round trip time) environment. Nevertheless, with limited buffer and short satellite round trip time such as in LEO systems (SRTT=10ms), the E2E scheme is preferred.

On the other hand, TCP splitting does not scale well as the number of TCP flows are increased. Furthermore, the efficiency of TCP over SR-ARQ(e.g. total E2E throughput) can be significantly improved by allowing more TCP connections and must be carefully considered for future broadband satellite IP networks that must serve many users.

References

[1] N. Ghani and S. Dixit, "TCP/IP Enhancements for Satellite Networks", IEEE Commun. Magazine, July 1999, pp.64-72.

[2] P. Charalambos, V. S. Frost and J. B. Evans, "Performance of TCP Extensions on Noisy High BDP Networks", IEEE Commun. Letters, vol. 3, no. 10, pp. 294-296, Oct. 1999.

[3] I. F. Akyildiz, X. Zhang, and J. Fang, "TCP-Peach+: Enhancement of TCP-Peach for Satellite IP Networks", IEEE Comm. Letter, vol. 6, no. 7, pp. 303-305, July 2002.

[4] I. F. Akyildiz, G. Morabito and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite IP networks" , IEEE/ACM Trans. Networking, pp. 307-321, June 2001.

[5] M. Mathis and J. Mahdavi, "Forward Acknowledgment: Refining TCP Congestion Control", Proc. ACM Sigcomm 1996, pp. 281-291, Stanford, CA, Aug. 1996.

[6] A. V. Bakre and B. R. Badrinath, "Implementation and Performance Evaluation of Indirect TCP", IEEE Trans. Computer, vol. 46, no. 3, pp. 260-278, Mar. 1997.

[7] http://www.mentat.com/skyx/skyx-gateway.html

[8] http://www.flash-networks.com

[9] 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.

[10] J. L. Mineweaver, J. S. Stadler, S. Tsao, M. Flanagan, "Improving TCP/IP Performance For The Land Mobile Satellite Channel", MILCOM 2001, vol. 1, pp.711-718, Oct. 2001.

[11] J. Ishac, M. Allman, "On The Performance of TCP Spoofing in Satellite Networks", MILCOM 2001, vol. 1, pp.700-704 , Oct. 2001.

[12] G. Bao, “Performance Evaluation of TCP/RLP Protocol Stack over CDMA wireless link”, ICCT’96, 1996, vol.2, pp. 710-713.

[13] 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.

[14] A. Chockalingam, M. Zorzi and V. Tralli, “Wireless TCP Performance with Link Layer FEC/ARQ”, ICC’99, 1999, vol.2, pp. 1212-1216.

[15] Y. Bai, A. T. Ogielski, G. Wu, “Interactions of TCP and Radio Link ARQ Protocol”, VTC’99, 1999, vol.3, pp. 1710-1714.

[16] 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.

[17] 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.

[18] 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.

[19] A-F. Canton and T.Chahed, “End-to-End Reliability in UMTS : TCP over ARQ”, Globecom 2001, vol. 6, pp. 3473-3477, Nov. 2001.

[20] Jackson W. K. Wong and Victor C. M. Leung, “Improving End-to-End Performance of TCP Using Link-Layer Retransmission over Mobile Internetworks”, ICC’99, vol. 1, pp. 324-328, June 1999.

[21] C. F. Chiasserini and M. Meo, “Modeling Interactions between Link Layer and Transport Layer in Wireless Networks”, Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on , 2001, pp. A-73-A-77 vol.1

[22] T.V. Lakshman, U. Madhow, “The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss”, IEEE/ACM Trans. on Networking, vol. 5, no. 3, pp. 336-350 June 1997.

[23] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, R. H. Katz, “A Comparison of Mechanism for Improving TCP Performance over Wireless Links”, IEEE/ACM Trans. on Networking, vol. 5, no. 6, pp. 756-769, Dec. 1997.

[24] J. Zhu, Z. Niu, Performance Evaluation of TCP over SR-ARQ in Correlated Fading Channel”, In Proceedings International Seminar on Teletraffic and Network 2000, pp. 217-224, Nov. 15-17 2000, Hangzhou, China.

[25] J. Zhu, Y. Wu, Z. Niu, A Delayed Multiple Copy Retransmission Scheme for Data Communication in Wireless Networks, the 9th IEEE International Conference on Networks, pp. 310-315, Oct. 10-12 2001, Bankok, Tailand.

[26] B. Suter, T. V. Lakshman, D. Stiliadis, and A. Choudhury, Design considerations for supporting TCP with per-flow queueing, in Proc. IEEE INFOCOM, 1998, pp. 299-306.

[27] R. Guerin, S. Kamat, V. Peris, and R. Rajan, Scalable QoS provision through buffer management, in Proc. ACM SIGCOMM 1998, pp. 29-40, 1998

[28] T. V. Lakshman, U. Madhow, and B. Suter, TCP/IP Performance with Random Loss and Bidirectional Congestion, IEEE/ACM Transactions on Networking, vol. 8, no. 5, pp. 541-555, Oct. 2000.

[29] A. Abouzeid and S. Roy, Stochastic Modeling of TCP in Networks with Abrupt Delay Variations, ACM/Kluwer Wireless Networks (WINET), 9(6), 509-24, September 2003.

[30] Y. Chotikapong, H. Cruickshank, and Z. Sun, Evaluation of TCP and Internet Traffic via Low Earth Orbit Satellites, IEEE Personal Communications, vol. 8, issue 3, pp. 28-34, June 2001.

[31] IETF RFC 3135 - Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations, http://www.faqs.org/rfcs/rfc3135.html

APPENDIX

A. The Derivation of $E(\widetilde{d}(k)|k=K)$

Eq.(3) gives the probability distribution function of total transmission time $\widetilde{d}(k)$ under the condition of "$k = K$". Thus we can compute the mean value of $\widetilde{d}(k)$ as follows.

$$\begin{aligned} \mathrm{E}(\widetilde{d}(k)|k=K) &= \lim_{J->\infty} \sum_{j=1}^{J} jT_2[(1-r^j)^K - (1-r^{(j-1)})^K] \\ &= T_2 - T_2(\sum_{i=1}^{K} \begin{pmatrix} K \\ i \end{pmatrix} (\frac{r^i - \lim_{J->\infty}(r^{iJ})}{1-r^i})(-1)^i) \quad (59) \\ &= T_2 \sum_{i=1}^{K} \begin{pmatrix} K \\ i \end{pmatrix} \frac{1}{1-r^i}(-1)^{i-1}. \quad (60) \end{aligned}$$

J, the maximum limit on retransmission of SR-ARQ must be $>> 1$ for achieving sufficiently low residual loss rate as mentioned in Sec.II. Therefore, we have

$$r^i - r^{iJ} \approx r^i, \quad (J >> 1,\ i \geq 1,\ 0 < r < 1). \quad (61)$$

From Eq.(59) and (61), we conclude that the delay estimates for SR-ARQ between fully reliable ($J = \infty$) and partially reliable ($1 << J < \infty$) case are negligible.

Note that for $K = 1$, Eq.(60) simplifies to

$$E(\widetilde{d}(k)|k=1) = \frac{T_2}{1-r}, \quad (62)$$

as expected.

We use E_K to denote $\mathrm{E}(\widetilde{d}(k)|k=K)$; Eq.(60) can be re-written as

$$E_K = T_2 \sum_{i=1}^{K} \begin{pmatrix} K \\ i \end{pmatrix} \frac{1}{1-r^i}(-1)^{i-1} = T_2 \sum_{j=0}^{\infty}[1-(1-r^j)^K]. \quad (63)$$

It is clearly seen from Eq.(63) that E_K is a monotonously increasing function of K because $(1-r^j) < 1$ for $0 < r < 1$ and $j > 0$ (i.e. $-a^x$ is a monotonously increasing function of x iff $0 < a < 1$).

From Eq.(63), we can also get a closed-form approximation to Eq.(60) when K is small.

$$E_K = T_2 \sum_{j=0}^{\infty}[1-(1-r^j)^K] \approx T_2(1 + \sum_{j=1}^{\infty}[1-(1-Kr^j)]) = T_2(1 + K\frac{r}{1-r}). \quad (64)$$

The total number of packets in the link is (upper) bounded by the bandwidth-delay product (BDP) of the link. Since satellite links have large bandwidth-delay product (BDP), using Eq.(64) to approximate Eq.(60) will result in significant error. Hence we next present another closed-form approximation with higher accuracy.

First, consider the increments $\Delta E_K = E_{K+1} - E_K$ where

$$\Delta E_K \;=\; T_2 \sum_{j=0}^{\infty}[1-(1-r^j)^{K+1}] - T_2 \sum_{j=0}^{\infty}[1-(1-r^j)^{K}] = T_2 \sum_{j=0}^{\infty}[(1-r^j)^K r^j]. \quad (65)$$

Eq.(65) shows that the increments ΔE_K are monotonously decreasing function of K (i.e. a^x is a monotonously decreasing function of x iff $0 < a < 1$). By approximating the summation with integration, Eq.(65) turns into

$$\Delta E_K \;\approx\; T_2 \int_0^{\infty}[(1-r^x)^K r^x]dx = -\frac{T_2}{\ln(r)}(\frac{1}{K+1}). \quad (66)$$

Thus

$$E_K \approx \int \Delta E_K dK = \int -\frac{T_2}{\ln(r)}(\frac{1}{K+1})dK = -\frac{T_2}{\ln(r)}\ln(K+1) + C, \quad (67)$$

where C is a constant. Since $E_1 = \frac{T_2}{1-r}$, we have

$$\frac{T_2}{1-r} = -\frac{T_2}{\ln(r)}\ln(2) + C. \quad (68)$$

Therefore,

$$C = T_2(\frac{1}{log_2(r)} + \frac{1}{1-r}) \quad (69)$$

and finally,

$$E_K \approx T_2(\frac{1}{log_2(r)} + \frac{1}{1-r} - log_r(K+1)) = \frac{T_2}{1-r}(1+\nu\ln(\frac{K+1}{2})), \quad (\nu = \frac{1-r}{\ln(1/r)}). \quad (70)$$

B. The Derivation of λ

The average throughput of E2E TCP with Link Layer SR-ARQ support is given as

$$\lambda = \frac{\frac{3}{8}w_{max}(w_{max}+2)}{\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w))}. \tag{71}$$

Eq.(20) gives the value of $E(\tau(w))$, which has different forms for different parameter configurations. We demonstrate the exact result for every case as follows.

Case I: $B \geq \mu T_2$

i. $B \geq w \geq \mu T_2$

$\theta = \mu T_2$, $\min(B, w) = w$, and $E^{(1)}(\tau(w)) = T_1 + \frac{T_2}{4} + \frac{w-\mu T_2}{\mu(1-r)} + \frac{T_2}{1-r}(\nu \ln(\mu T_2 S/2)+1)$.

ii. $w \geq B \geq \mu T_2$

$\theta = \mu T_2$, $\min(B, w) = B$, and $E^{(2)}(\tau(w)) = T_1 + \frac{T_2}{4} + \frac{B-\mu T_2}{\mu(1-r)} + \frac{T_2}{1-r}(\nu \ln(\mu T_2 S/2)+1)$.

iii. $B \geq \mu T_2 \geq w$

$\theta = w$ and $E^{(3)}(\tau(w)) = T_1 + \frac{w}{4\mu} + \frac{T_2}{1-r}(\nu \ln(wS/2) + 1)$.

Case II:$B \leq \mu T_2$

i. $B \leq w$

$\theta = B$ and $E^{(4)}(\tau(w)) = T_1 + \frac{B}{4\mu} + \frac{T_2}{1-r}(\nu \ln(BS/2) + 1)$.

ii. $B > w$

$\theta = w$ and $E^{(5)}(\tau(w)) = T_1 + \frac{w}{4\mu} + \frac{T_2}{1-r}(\nu \ln(wS/2) + 1)$.

In the following, we first compute the denominator $\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w))$ in Eq.(71) using the above results of $E(\tau(w))$, then use Eq.(71) to calculate the average throughput.

Case I: $B > \mu T_2$ (i.e., $\beta > 1$)

1) $w_{max}/2 \geq \mu T_2$ (i.e., $\beta \geq 1+\rho$):

$$\begin{aligned}\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w)) &= \sum_{w=\frac{w_{max}}{2}}^{B} E^{(1)}(\tau(w)) + \sum_{w=B+1}^{w_{max}} E^{(2)}(\tau(w)) \\ &= (\frac{w_{max}}{2}+1)(T_1 + T_2\mathcal{T}) - (\frac{\rho}{1+\rho})^2 \frac{(B+1)^2}{2\mu(1-r)}. \end{aligned} \tag{72}$$

Hence

$$\lambda = \frac{\frac{3}{4}w_{max}(\frac{w_{max}}{2}+1)}{(\frac{w_{max}}{2}+1)(T_1+T_2\mathcal{T}) - (\frac{\rho}{1+\rho})^2\frac{(B+1)^2}{2\mu(1-r)})}$$

$$\approx \frac{\frac{3}{2}\frac{(B+1)}{1+\rho}(1-r)}{T_1(1-r)+\frac{B}{\mu}+\frac{T_2}{4}(1-r)+T_2(\nu\ln(\frac{\mu T_2 S}{2}))-\frac{(\frac{\rho}{1+\rho})^2\frac{(B+1)^2}{2\mu}}{\frac{B+1}{1+\rho}+1}}$$

$$= \frac{\frac{3}{2}\frac{B}{T_2}(\frac{1-r}{1+\rho})}{\frac{T_1(1-r)}{T_2}+T_2\nu\ln(\frac{\mu T_2 S}{2})+\frac{(1-r)}{4}(\frac{1+\rho-\frac{1}{2}\rho^2}{1+\rho})\frac{B}{\mu}} \quad . \tag{73}$$

2) $w_{max}/2 \leq \mu T_2$ (i.e., $\beta \leq 1+\rho$):

$$\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w)) = \sum_{w=\frac{w_{max}}{2}}^{\mu T_2 -1} E^{(3)}(\tau(w)) + \sum_{w=\mu T_2}^{B} E^{(1)}(\tau(w)) + \sum_{w=B+1}^{w_{max}} E^{(2)}(\tau(w))$$

$$= \sum_{w=\frac{w_{max}}{2}}^{\mu T_2 -1} (T_1+\frac{w}{4\mu}+\frac{T_2}{1-r}(\nu ln(wS/2)+1)) + (w_{max}-\mu T_2+1)(T_1+T_2\mathcal{T}) - \frac{(B-\mu T_2+1)(B-\mu T_2)}{2\mu(1-r)}$$

$$\approx (\frac{w_{max}}{2}+1)(T_1+\frac{T_2}{1-r}(1+\nu ln(S/2))) + \frac{T_2}{1-r}(\nu ln(\frac{(\mu T_2-1)!}{[\frac{w_{max}}{2}-1]!}) + (2\frac{B}{1+\rho}-\mu T_2)(\nu ln(\mu T_2)+\frac{1-r}{4}+\frac{B-\mu T_2}{\mu T_2}))$$

$$+(\frac{w_{max}}{2}+1)(\frac{\frac{(1+\rho)(\mu T_2)^2}{B}-\frac{B}{(1+\rho)}}{8\mu}+\frac{(2\mu T_2-\frac{(\mu T_2)^2}{B}-B)(1+\rho)}{2\mu(1-r)}). \tag{74}$$

Let

$$\begin{cases} X_1 = 2-\frac{(1+\rho)}{\beta} \\ X_2 = 2-\frac{(1+\rho)}{2\beta}-\frac{\beta}{2(1+\rho)} \\ X_3 = 2\beta-1+\frac{(1+\rho)}{2\beta}-\frac{\beta(1+\rho)}{2}+\frac{(1+\rho)}{B}\nu\ln(\frac{(\mu T_2-1)!}{[\frac{B}{1+\rho}-1]!}) \end{cases} \tag{75}$$

Hence

$$\lambda = \frac{\frac{3}{2}\frac{B}{T_2}(\frac{1-r}{1+\rho})}{\frac{T_1(1-r)}{T_2}+X_1\nu\ln(\mu T_2)+X_2\frac{1-r}{4}+X_3+\nu\ln(S/2)}. \tag{76}$$

Case II: $B \leq \mu T_2$ (i.e., $\beta \leq 1$)

$$\sum_{w=\frac{w_{max}}{2}}^{w_{max}} E(\tau(w)) = \sum_{w=\frac{w_{max}}{2}}^{B} E^{(5)}(\tau(w)) + \sum_{w=B+1}^{w_{max}} E^{(4)}(\tau(w))$$

$$\approx (\frac{w_{max}}{2}+1)(T_1+\frac{T_2}{1-r}(1+\nu\ln(S/2))+\frac{B(1+\rho-\frac{1}{2}\rho^2)}{4\mu(1+\rho)})+\frac{T_2}{1-r}(\nu\ln(\frac{(B-1)!}{[\frac{w_{max}}{2}-1]!})+(w_{max}-B)(\nu\ln(B))). \tag{77}$$

Hence

$$\lambda = \frac{\frac{3}{2}\frac{B}{T_2}(\frac{1-r}{1+\rho})}{\frac{T_1(1-r)}{T_2}+(1-\rho)\nu\ln(B)+(\frac{B}{\mu T_2(1+\rho)}(1+\rho-\frac{1}{2}\rho^2))\frac{1-r}{4}+1+\frac{(1+\rho)}{B}\nu\ln(\frac{(B-1)!}{[\frac{B}{1+\rho}-1]!})+\nu\ln\frac{S}{2}}. \tag{78}$$

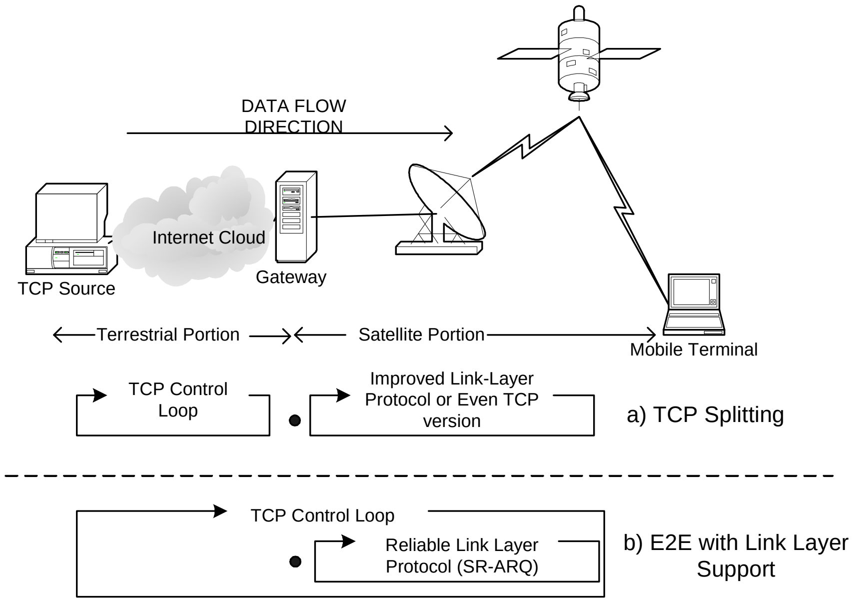


Fig. 1. Network scenarios of a) TCP splitting and b) E2E with link layer support

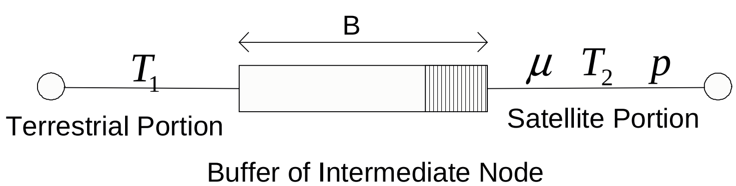


Fig. 2. System Model

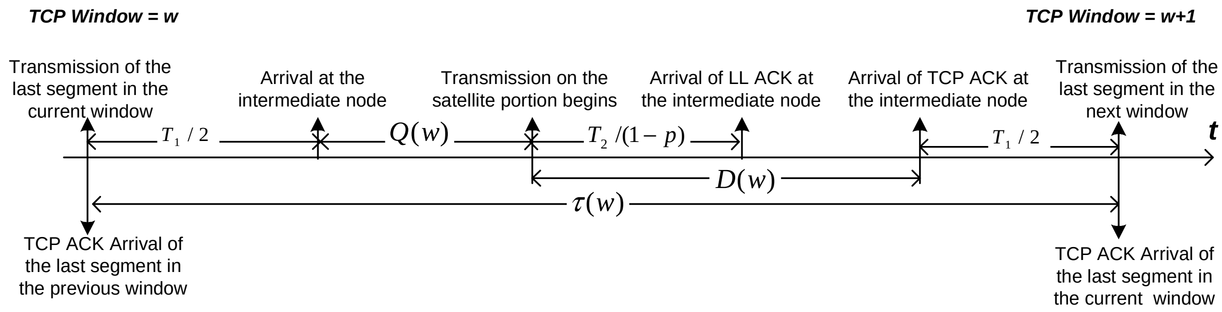


Fig. 3. Event sequencing of the last segment in a TCP window transfer

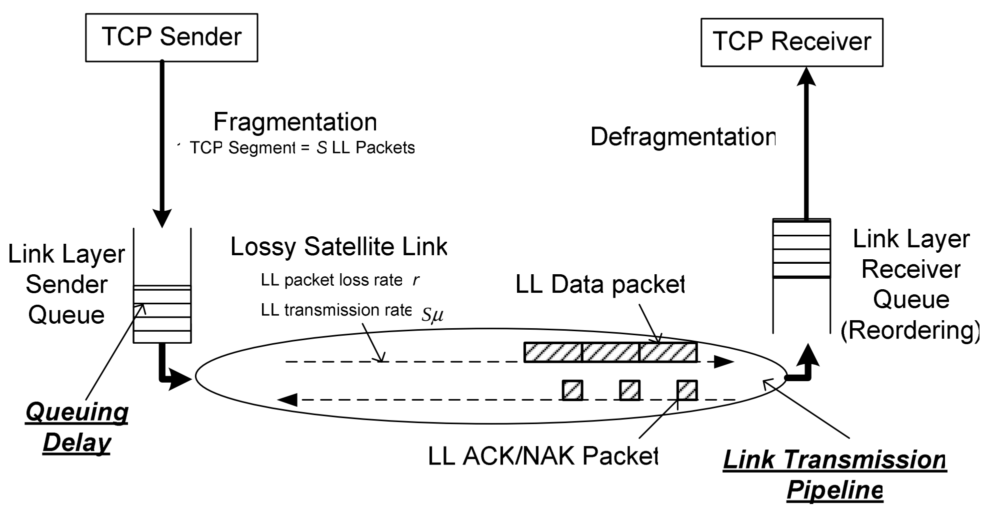


Fig. 4. Anatomy of the Link Layer (LL) queuing process and transmission pipeline

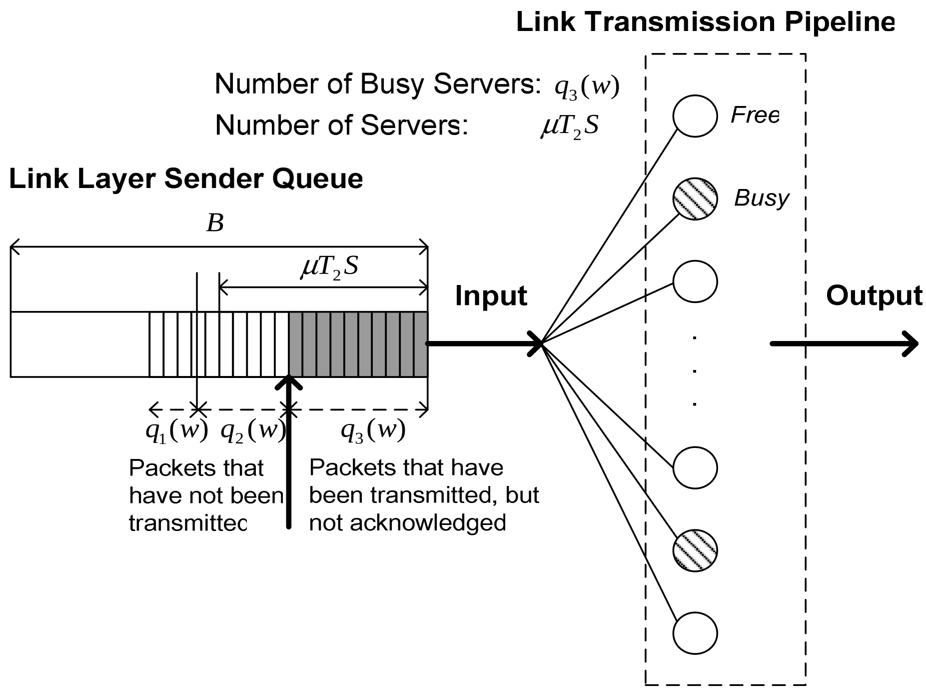


Fig. 5. Modeling LL SR-ARQ as a Multi-Server Queue

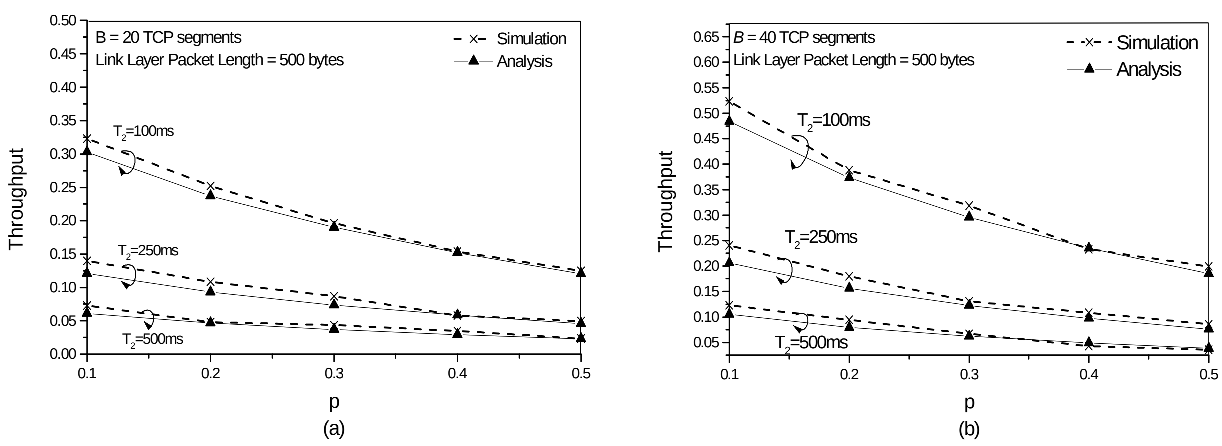


Fig. 6. Throughput vs Segment loss rate ($T_1 = 0$, analysis results from Eq.(39))

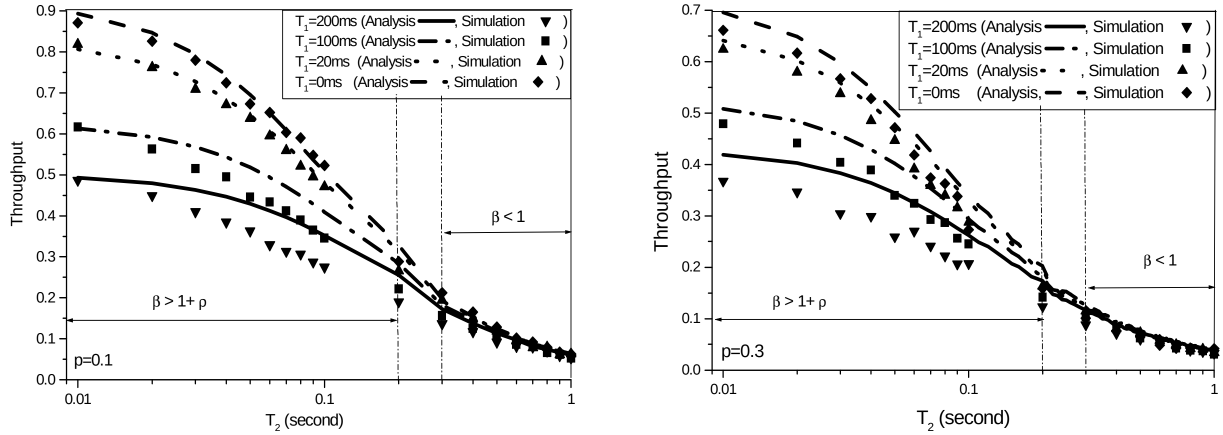


Fig. 7. Effect of Satellite Round Trip Time T_2 (B=50 TCP segments, analysis results from Eq.(39))

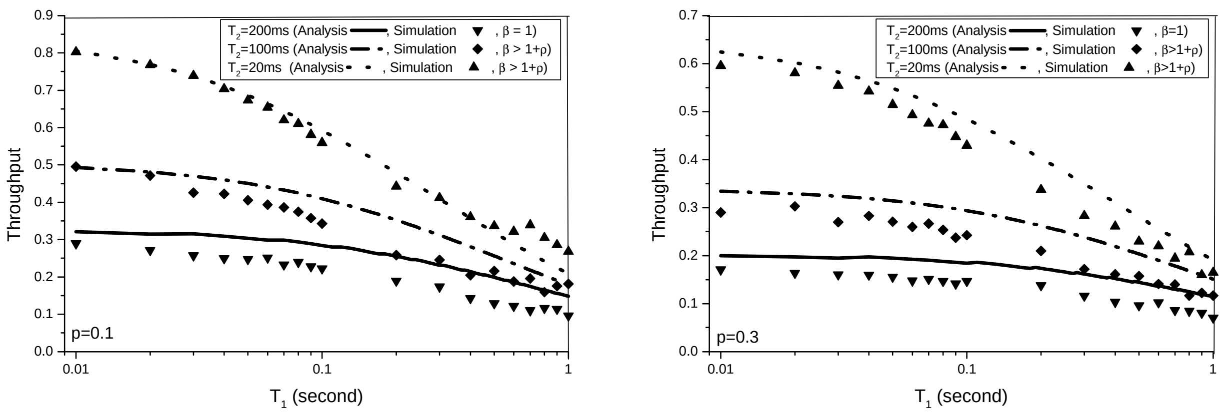


Fig. 8. Effect of Terrestrial Round Trip Time T_1 (B=50 TCP segments, analysis results from Eq.(39))

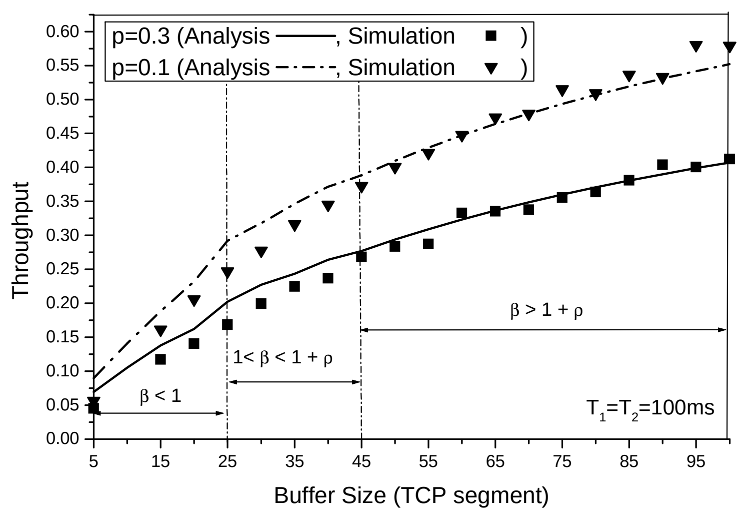
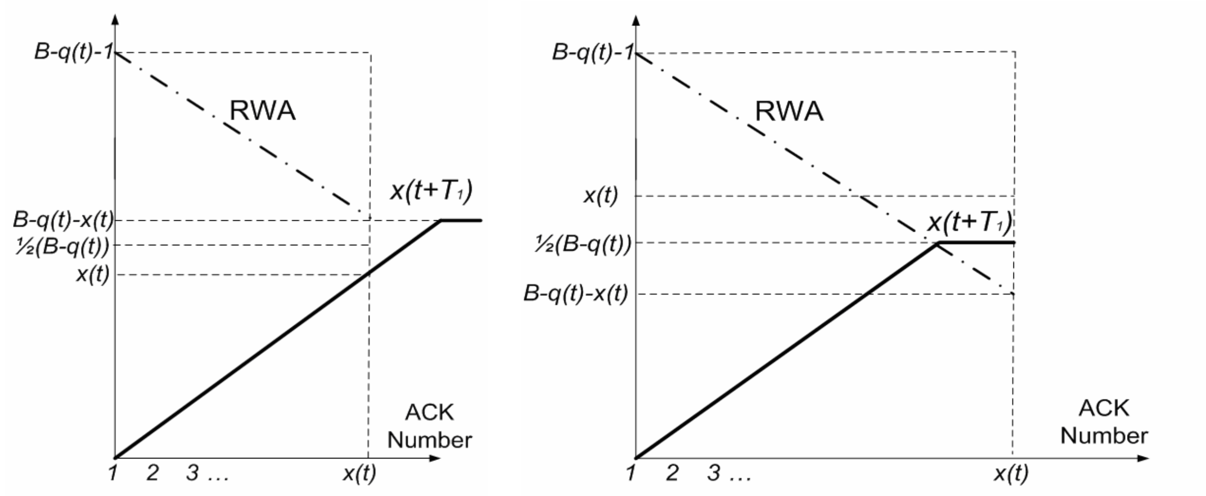
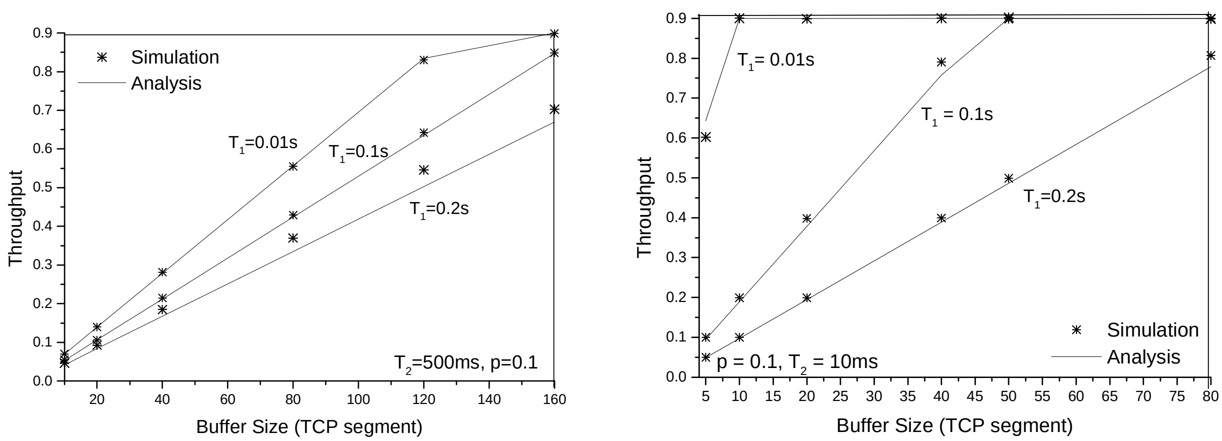


Fig. 9. Effect of Buffer Size (analysis results from Eq.(39))



a) $x(t) < \frac{1}{2}(B - q(t))$ b) $x(t) \geq \frac{1}{2}(B - q(t))$

Fig. 10. Relation between the length of next burst $x(t + T_1)$ and the TCP window size (determined by RWA) as a function of in-burst ACK index with any given $x(t)$



a) T_2 = 500ms b) T_2=10ms

Fig. 11. Effect of Buffer Size (analysis results from Eq.(50))

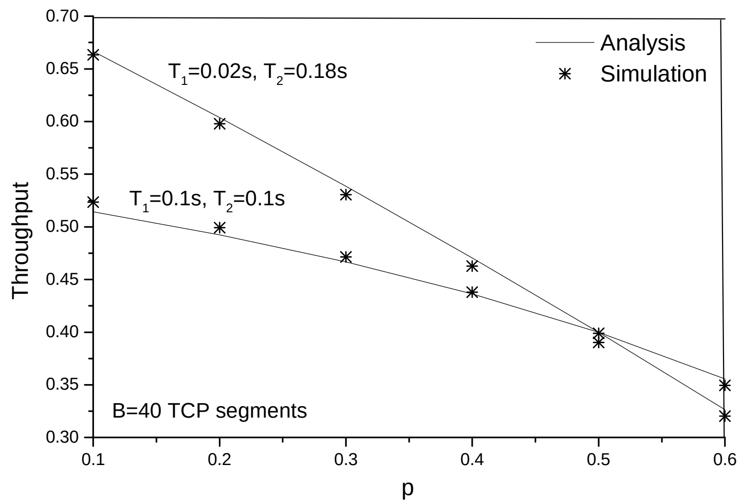


Fig. 12. Effect of segment loss rate (analysis results from Eq.(50))

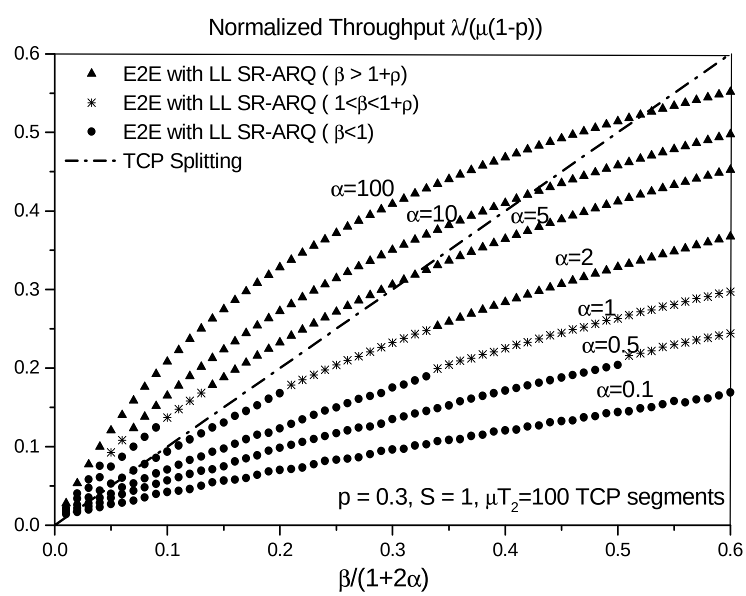
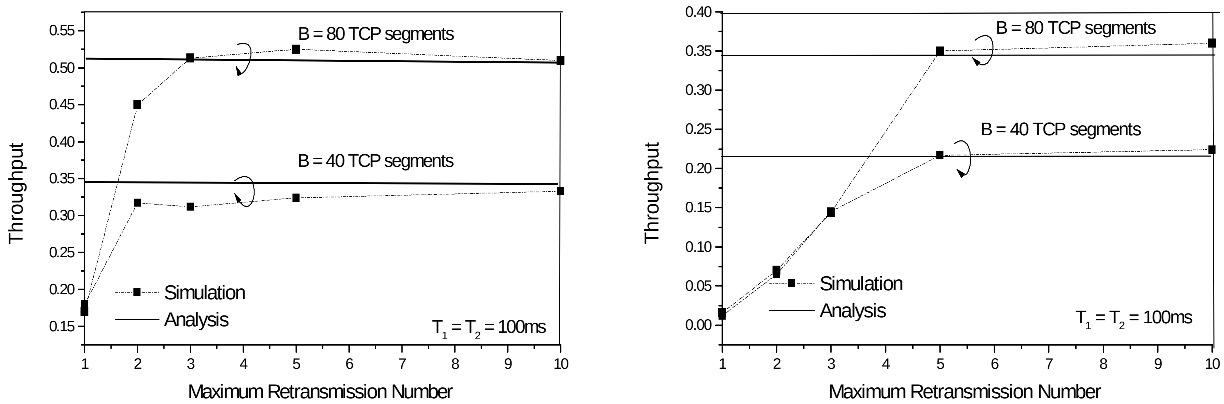
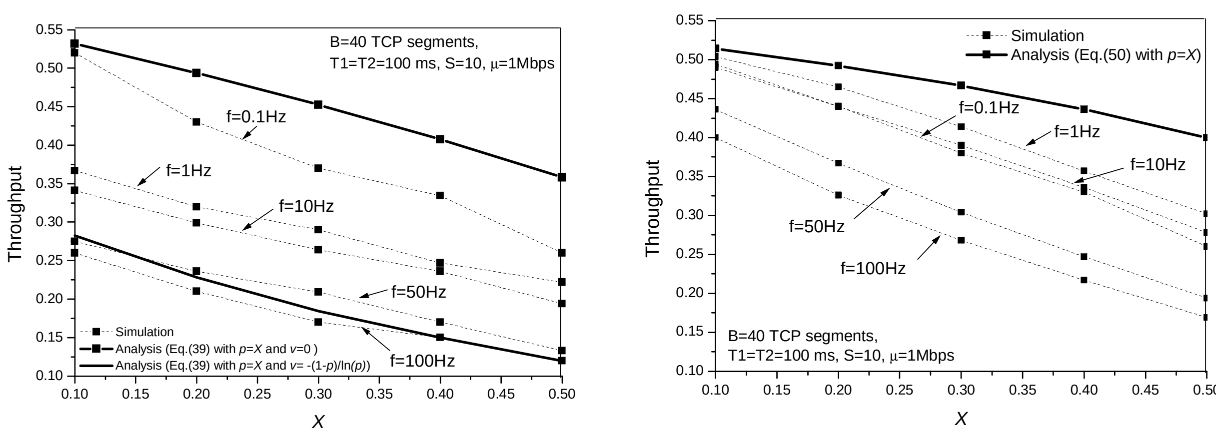


Fig. 13. Normalized Throughput Comparison with TCP Splitting to E2E with LL SR-ARQ



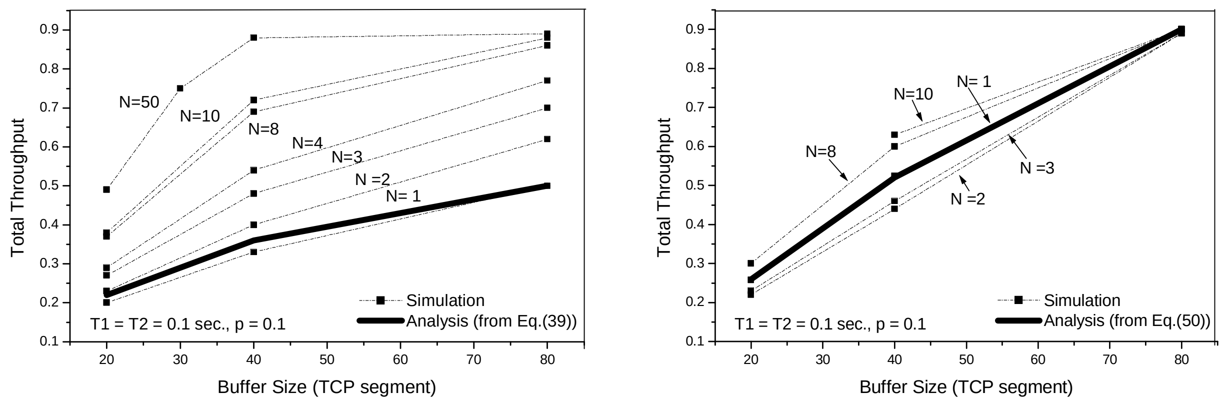
a) p=0.1 b) p=0.3

Fig. 14. Effect of Maximum Retransmission Number (analysis results from Eq.(39))



a) TCP over SR-ARQ b) TCP Splitting

Fig. 15. Effect of Correlated Fading Channel



a) TCP over SR-ARQ

b) TCP Splitting

Fig. 16. Effect of Number of Connections (N: number of connections)

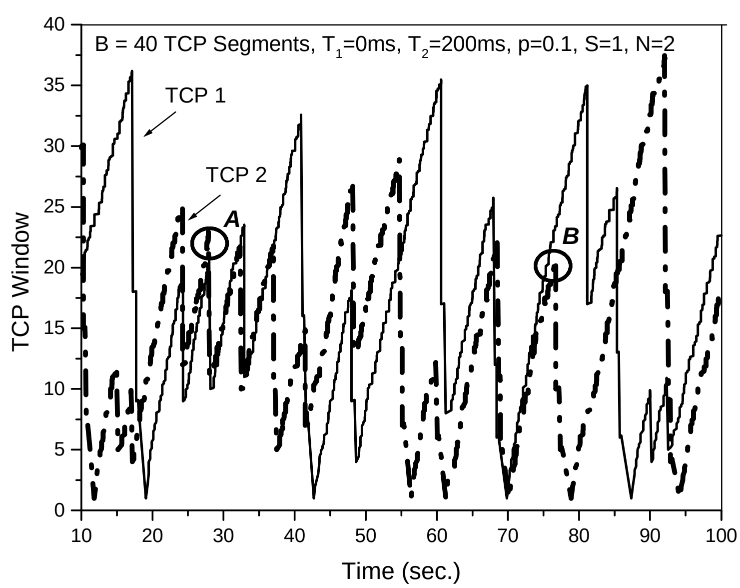


Fig. 17. Tracing TCP Window Size for Two TCP Connections