Effects of Advertised Receive Buffer Size and Timer Granularity on TCP Performance over Erroneous Links in a LEO Satellite Network

Marc Emmelmann

emmelmann@ieee.org
Fraunhofer Institut for Open Communication Systems (FOKUS)
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

***Abstract*–This paper presents a preliminary investigation into the impact of TCP's advertised receive buffer size and timer granularity on TCP performance over erroneous links in a LEO satellite environment. Conducted simulations include over 200 different combinations of TCP flavor, advertised receive buffer size, timer granularity, and bit error rate. Results show that TCP can only approximate the variable propagation delay for unsaturated links and that the minimum timer granularity which prevents a premature expiration of the RTO depends on the advertised receive buffer size. Low BERs do not influence TCP's capability to track the variable propagation delay in contrast to high BERs. Final results indicate that the relative performance degradation of TCP over erroneous links does not depend on the ability to estimate the variable propagation delay.**

I. Introduction

This paper investigates the performance of the Transmission Control Protocol (TCP) [1] in a variable delay satellite environment, esp. how it is effected by different bit error rates (BERs), values of TCP's timer granularity, and advertised receive buffer sizes. FhI FOKUS, DLR (German Aerospace Agency), and Tesat-Spacecom (former Bosch SatCom) have completed the system design for an ATM-based, LEO satellite multimedia communication network (ATM-Sat); a prototype based demonstrator is currently under development. The project considers TCP as the primary protocol for a reliable, point-to-point data transfer and includes its evaluation under project specific network parameters (i.e: orbit altitude and bandwidth restrictions). In addition to satellite related TCP research considering rather static long-delay networks (see [2], [3], and [4]), a previous investigation reveals TCP's ability to estimate variable, inter-satellite propagation delays on error-free links [5].

TCP offers a reliable data transfer by detecting and retransmitting lost segments [6]. It estimates the round trip time (RTT) of the propagation path with a timer granularity G and calculates on this basis the retransmission timeout (RTO). If an acknowledgment has not been received before the RTO expires, TCP assumes the segment to be lost and starts its retransmission. The closer the RTO approximates the real segment RTT, the earlier a lost segment can be retransmitted. A rather coarse timer granularity degrades this approximation but, in turn, prevents needless retransmissions due to single spikes in the RTT [7].

The preliminary investigation presented in this paper is structured as follows: Section II. describes the simulation environment including the propagation delay model and network parameters. The effects of different receive buffer sizes and timer granularities on TCP's ability to estimate the variable propagation delay are analysed in section III. and, in section IV. , reevaluated in the presence of BERs. Section V. gives the conclusion and future work in this area.

II. Simulation Setup

All experiments are conducted with the OPNET network simulator [8] using an adapted model of its point-to-point duplex link to support variable delays of the project specific LEO satellite network (orbit altitude of 1350 km). We divide the formula for the slant range given in [9] by the speed of light and include the result in OPNET's pipeline stage for the propagation delay. The analytically gained slant range is verified with the Satellite Took Kit (STK, version 4.2). Fig. 1. illustrates how the experienced delay is periodically repeated for simulations lasting longer than the visibility of a single satellite. This repetition illustrates the delay for the envisioned ATM-based LEO satellite system whenever a handover between two satel-

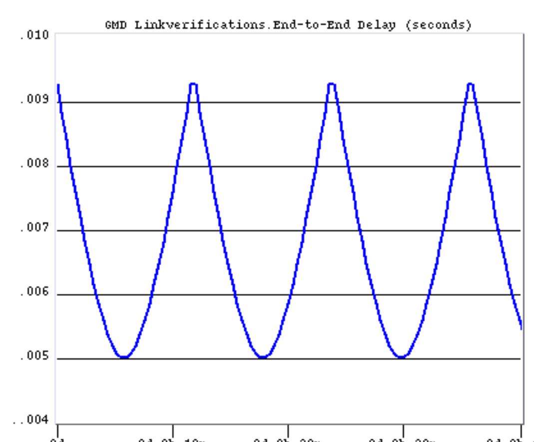

Fig. 1. LEO satellite one-way delay, periodically repeated

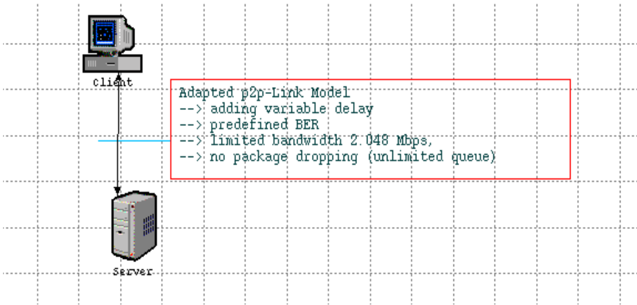

Fig. 2. Simulation Setup

lites occurs as soon as the minimum elevation angle falls below 20^{o}.

The link that connects server and client may corrupt traffic with a preset BER and limits the bandwidth to 2.048 Mb/s. Conducted experiments include bit errors of 2*10e-8 and 2*10e-5 in addition to an error-free connection. The link's maximum segment size equals 1500 bytes. Fig. 2. illustrates the setup. It should be noted that OPNET's point-to-point link model does not drop any traffic due to overload situations and is therefore acting as an unlimited queue.

For the experiments, a FTP connection is established between client and server on top of OPNET's TCP/IP model to transfer a 250 MB large file. We conduct the experiments with two TCP flavors, i.e., TCP Reno and TCPmin.[1] The file size guarantees that TCP's congestion window (CWND) grows beyond the bandwidth delay product (BDP). The client advertises a receive buffer of either 64 kB, 3.300 kB, or 1.46 kB. These values assure experiments where the buffer is never a limiting factor for the transmission, holds approx. 50 percent of the BDP, or equals to the segment size minus 40 bytes consumed by the IP header.

The smallest possible TCP timer granularity avoiding false retransmissions due to a premature expiration of the retransmission timeout (RTO) is determined. The RTO's lower bound is set to 1 ms. Additionally, file transfers are conducted with preset timer granularities G in between 1 ms and 500 ms and lower-bounded RTOs of 100 ms, 200 ms, and (for OPNET's default settings) 500 ms. Table 1 numbers the experiments for a preset triple of TCP flavor, BER, and buffer size.

III. Effects Of Receive Buffer Size And Timer Granularity On SRTT And RTO

First experiments check on TCP's ability to accurately track the RTT and compare the latter to the RTO. TCP periodically clocks the RTT of transmitted segments and computes a smoothed version (SRTT) and the sample's variance (RTTVAR). The accuracy of the RTT measurement depends on the

1. TCPmin denotes the flavor where all TCP enhancements are disabled. TCP Reno differs only by adding the Fast Retransmit and Recovery algorithms.

TABLE 1
EXPERIMENT NUMBERS FOR A GIVEN TCP FLAVOR, BUFFER SIZE, AND BER

Lower Bound Of RTT Timer Granularity G	None	100 ms	200 ms	Default
G=1 ms		2	7	
G=50 ms		3	8	
G=100 ms		4	9	
G=250 ms		5	10	
G=500 ms		6	11	12
Minimal G to avoid false retransmissions (varying)	1			

granularity G by which TCP's clock is increased. Finally, TCP calculates the RTO upon the current SRTT and RTTVAR. (A detailed discussion of the formulas used in this experiment as well as in most TCP implementation is given in [10].) TCP assumes a timed segment to be lost and triggers its retransmission if the RTO expires before the segment is acknowledged. The cause may either be an inaccurate RTO estimator, i.e., the RTO graph falls below the actual RTT, or a packet loss/delay due to network congestion. As TCP always assumes the network to be congested, the congestion window is reduced in either case.

Experiments show that for the chosen LEO satellite delay pattern, TCP's timer is vulnerable to both, the advertised receiver buffer size and the timer granularity. The advertised receive buffer is directly proportional to the queuing delay and has no effect on an accurate RTT estimation for a timer granularity of "zero" and buffer values limiting the link's utilization to less than 100% for the entire FTP transmission. Fig. 3. illustrates this case for a receive buffer size of 8760

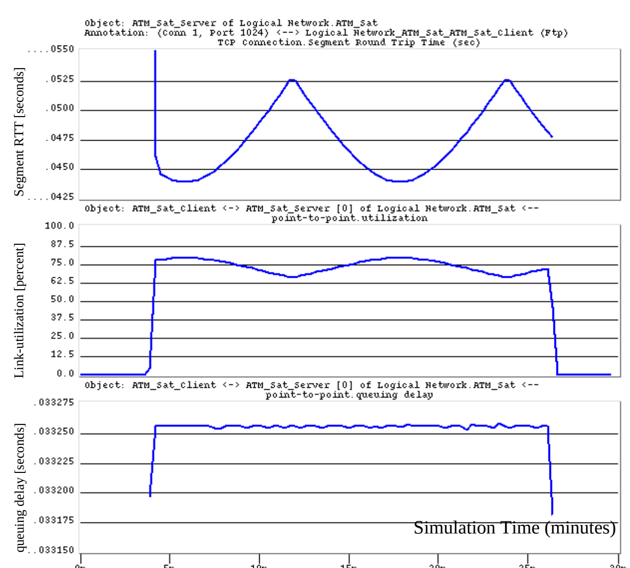

Fig. 3. SRTT, Link Utilization, and Queuing Delay
(Receive Buffer Size of 8760 bytes)

TABLE 2
Lowest Possible Value Of TCP Timer Granularity To Avoid Needless Retransmissions

Advertised Receive Buffer	Min. Timer Granularity (G)
64 kB	1 ms
6.6 kB (BDP)	1 ms
3.3 kB (1/2 BDP)	4 ms
1.46 kB (MSS)	26 ms (250 ms)[a]

a. Only consecutive values larger than 200 ms avoid needless retransmissions. G = 26 ms is a single occurrence.

bytes. The RTO (not shown) follows the SRTT so closely that the two plots are not distinguishable. TCP's SRTT and RTO timers only reflect the predominant, constant queuing delay for receive buffers resulting in 100% link utilization.

The TCP connection reveals needless retransmissions due to a premature expiration of the RTO as soon as we introduce a timer granularity. Smaller advertised receiver buffers require a rather coarse timer granularity to avoid retransmissions whereas buffer sizes saturating the link allow to reduce G even to 1 ms (Table 2) without performance degradation. (Note, that for a saturated link, TCP cannot approximate the shape of the propagation delay but reflects the queuing delay.) Experiments with an advertised window size equal to the MSS have to be evaluated separately. TCP uses delayed ACKs and confirms only after a timeout of 200 ms the single datagram that could be sent at a time due to buffer limitations. SRTT values are perfectly constant or slightly vary around 200 ms depending on the chosen timer granularity. For a RTTVAR of zero, the RTO equals the SRTT and results in false retransmissions shortly before the ACK arrives. Only granularities coarser than 200 ms cause a variation of the RTT and guarantee a perfect CWND growth.

IV. Introduction Of Bit Error Rates

The experiments conducted to analyze effects of buffer size and timer granularity are repeated with additionally introduced uniformly distributed bit error rates of 2*10e-5 and 2*10e-8. Project constrains require a permanent line-of-sight to the satellite; therefore the considered BERs represent free space loss and link degragation caused by rainfall.The former BER is the maximum required by project constrains and is achieved by the link layer's forward error correction (FEC). The latter derives from a worst case approximation of the considered LEO system for a QPSK-modulated signal and a minimum elevation angle of ten degrees [9][11]. The time to successfully transfer the 250 MB file (FTP response time) over the erroneous link is measured for each experiment and normalized by the time observed in the absence of BERs.

A BER of 2*10e-8 does not influence TCP's capability to estimate the link's delay characteristic whereas, for a BER of 2*10e-5, TCP's timer cannot reflect the variable propagation delay. Needless retransmissions due to a premature expiration of the RTO do not occur for either error rate. The "jagged" shape of the RTO curve in Fig. 4. is caused by the timer granularity and not by the BER and is hardly distinguishable from the error-free case. SRTT and RTO samples for an error rate of 2*10e-5 are all well above the longest segment round trip time as detected in the error-free case and do not even allude to the variable propagation delay (Fig. 5.).

TPC's ability to approximate the variable delay has no direct influence on its performance. The queuing delay dominates in all experiments with an advertised receive buffer of 64 kB whereas a buffer of 3.300 kB (50% of the BDP) allows TCP to reflect the temporal change in both, the SRTT samples and the RTO. Fig. 6. shows that the relative performance degradation caused by a BER of 2*10e-8 is in the same order for both buffer sizes and lays always below two percent (middle graphs). Higher normalized response times for "TCP with minimum settings" and a 64 kB receive buffer (upper graph) are caused by the TCP flavor itself: The congestion window is closed to one segment for each retransmission whereas TCP Reno employs fast retransmit and recovery setting the CWND to one-half of the minimum of the current CWND and the receiver's advertised

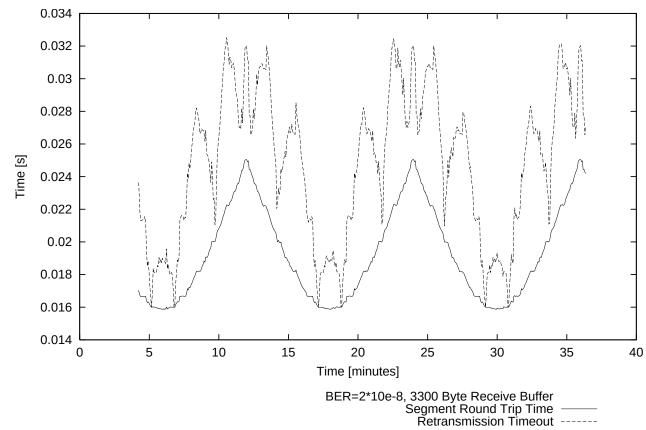

Fig. 4. SRTT and RTO estimations for a BER of 2*10e-8

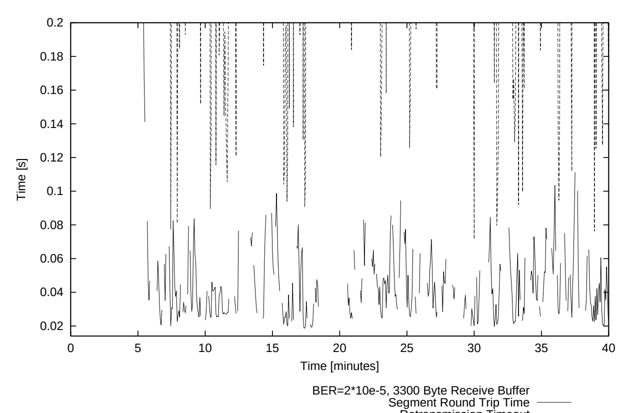

Fig. 5. SRTT and RTO estimations for a BER of 2*10e-5

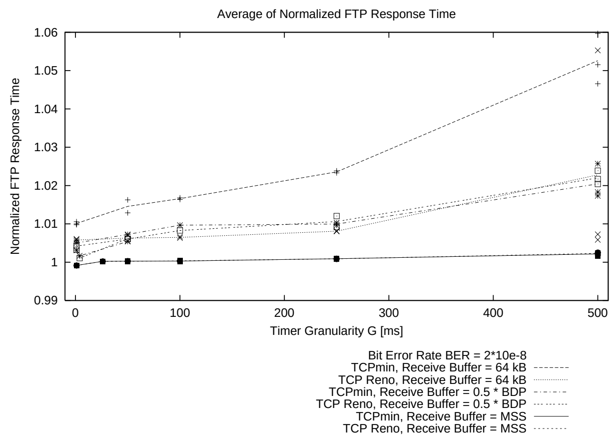

Fig. 6. Average of normalized FTP response time for a BER of 2*10e-8

window. The lower average CWND for TCP_{min} results in higher FTP response times. All experiments with the higher BER of 2*10e-5 end prematurely as TCP closes the connection after six unsuccessful retransmissions. The results in Fig. 7. are based on measurements with a time based retransmission scheme (connection aborts after retransmitting for 400 seconds). The high number of corrupted packets prevents TCP from taking advantage of the fast retransmit and recovery algorithm. Relative performance gains are primarily driven by the advertised receive buffer. Neither of the two BERs degrades the FTP response time for a receiver buffer of 1460 bytes as only one segment is inject at a time into the link and retransmissions are only triggered by the RTO for both TCP flavors.

V. Conclusion And Future Work

This paper presents how the timer granularity G and BERs effect TCP over a network path imposing a LEO satellite delay pattern. TCP is able to approximate the variable delay considered in this paper only for connections which do not saturate the link. The low BER (which can still be considered as a worst case approximation for terrestrial wired links) does not effect this ability. In turn, the high BER representing an erroneous satellite link without FEC does. The minimum timer granularity preventing needless retransmissions depends on the link utilization, i.e., the advertised receive buffer. Therefore, host which advertise a limiting receive buffer size are more vulnerable to a reduced timer granularity than hosts with auto-tuned socket buffers [12] which can fully utilize the link's bandwidth. The latter are in turn effected by a rapid growth of the queuing during slow start for very large BDPs [5]. These results suggest to reduce the standard granularity to 200 ms for the considered LEO satellite network, which halves the performance degradation of the presented experiments in the best case. Further improvements are only noticeable for "hand-tuned" values of G (which seems to be impractical for widely spread TCP implementations).

Planned extensions to this work include the implementation of a more realistic error model: The BER changes over the time due to the satellites movement, i.e., the free-space-loss depends on the satellite's elevation angle. Further channel degradation caused by rain fall may not be neglected.

Adding background traffic on the satellite link shall produce RTT spikes which in turn have an impact on the appropriate choice of minimum values for G and the RTO [7]. Adding an underlying, routeable network as in [13] should supplement the results.

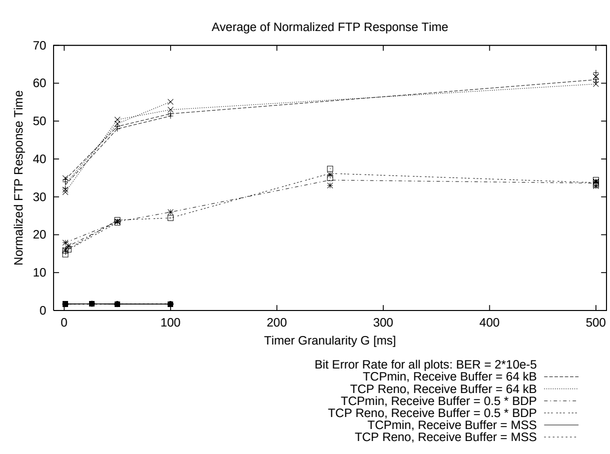

Fig. 7. Average of normalized FTP response time for a BER of 2*10e-5

Reference

[1] J. Postel, "Transmission Control Protocol," Sept. 1981, RFC 793.

[2] M. Allman, D. Glover, and L. Sanchez, "Enhancing TCP Over Satellite Channels Using Standard Mechanisms," Jan. 1999, RFC 2488.

[3] M. Allman, S. Dawkins, D. Glover, J. Griner, J. Heidemann, T. Herson, H. Kruse, S. Osterman, K. Scott, J. Semke, J. Touch, and D. Tran, "Ongoing TCP Research Related to Satellites," Feb. 2000, FRC 2760.

[4] R. Goyal, R. Jain, M. Goyal, S. Fahmy, B. Vandalore, and S. Kota, "Traffic management for TCP/IP over satellite ATM networks", *IEEE Communications Magazine*, Volume: 37 Issue: 3, March 1999.

[5] M. Allman, J. Griner, and A. Richard, "TCP Behavior in Networks with Dynamic Propagation Delay," *Proc. GLOBECOM*, Oct. 2000.

[6] M. Allman, V. Paxson, and W. R. Stevens, "TCP Congestion Control," Apr. 1999, RFC 2760.

[7] M. Allman, V. Paxson, "On Estimating End-to-End Network Path Properties," *ACM SIGCOMM*, Sept. 1999.

[8] www.opnet.com

[9] E. Lutz, M. Werner, and A. Jahn, *Satellite Systems for Personal and Broadband Communications*. Berlin, Germany: Springer, 2000.

[10] W. R. Stevens, *TCP/IP Illustrated: the protocols*. Reading, Massachusetts: Addison-Wesley, 1994.

[11] H. Bischel, "Kanalmodell Berechnungen für den ATM-Sat Demonstrator," unpublished.

[12] J. Semke, J. Mahdavi, and M. Mathis, "Automatic TCP Buffer Tuning," *ACM SIGCOMM*, Sept. 1998.

[13] L. Wood, G. Pavlou, and B. Evans, "Effects on TCP of routing strategies in satellite constellations," *IEEE Communications Magazine*, Volume: 39 Issue: 3, March 2001.