Performance of TCP extensions on noisy high BDP networks

Charalambous P. Charalambos, Victor S. Frost, Joseph B. Evans

August 26, 1998

Abstract

Practical experiments in a high bandwidth delay product (BDP) network environment assist in the design and understanding of future global networks. This paper describes the results of experiments with different implementations of TCP on a high speed ATM/SONET network over high delay and noisy channels. Comparisons are also made with host/traffic configurations over various smaller BDP systems. Tests were conducted with three different implementations of TCP; TCP Reno, TCP New Reno, and TCP SACK. In this paper we focus on experimental quantative comparisons of the three protocols on channels with varying bit error rates (BER) and round trip times (RTT).

1 Introduction

This experimental investigation covers environments that exhibit a high Bandwidth-Delay-Product (BDP). Such networks have large bit rates and inherent long propagation delay. An OC-3c (155 Mbps) Asynchronous Transfer Mode (ATM) network with a high data rate (HDR) geosynchronous earth orbit (GEO) satellite links is an example of such a network. A BDP infrastructure is the core of this investigation [12]. The NASA Advanced Communications Technology Satellite (ACTS) [10, 11], the AAI and MAGIC [9] national scale testbeds and the local ATM network were used in these experiments. A special purpose hardware [13] to insert controlled delay and bit errors was also used. Network experiments were carried out using application-level software Netspec [9].

The problem though with the above network architecture is that the reliable transport protocol - TCP - was designed for small BDP networks [5, 6]. Questions whether TCP/ATM could perform optimally over high BDP networks (like satellites) with some protocol enhancements and what those enhancements are must be answered. This paper investigates the TCP extensions on ATM over various noisy BDP networks by delivering real experimental results.

2 Transmission Control Protocol (TCP)

TCP is a transport protocol that offers connection-oriented and reliable-byte stream service. TCP is also flexible, supporting any underlying link technology [6], e.g. ATM. TCP is an end-to-end protocol with error, flow, and congestion control functions [5, 6]. Its basic implementation [7] is unsuitable for high BDP networks, and therefore modifications and additions have been added to enhance the performance [1]. The major requirement is the support of large flow control window. TCP Reno [3], TCP New Reno [4], and TCP SACK [2] over high BDP networks with errors and congestion have been investigated [13]; here the focus is on the error performance. More information on these protocols, as well as on the network architecture and systems we used can be found in [12, 13].

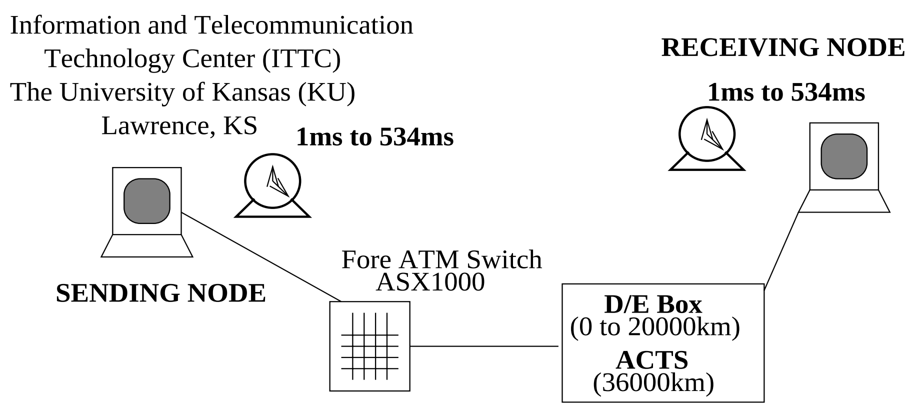

Figure 1: Scenario used to carry out the experiments. The D/E box was used to insert delays of 1 ms, 50 ms, and 100 ms with 4 different BERs. The satellite system was used for the extremely high BDP path test (534 ms).

3 Experiments and Results

Transmission errors can have a negative effect on TCP performance. TCP assumes that every loss is due to congestion [6] and hence the TCP congestion algorithms activated with a slow growth of the window size. But in wireless or satellite environments, losses are due to errors caused by the physical medium. A single error can cause the current congestion window size to be reduced by half.

Experiments were conducted on the testbed shown in Figure 1, by inserting errors in the link using the D/E device. BERs of $0.8 * 10^{-7}$, $1.3 * 10^{-8}$, $1.1 * 10^{-9}$, and $1.1 * 10^{-12}$ were used. Experiments were also conducted with RTT paths of 1 ms, 50 ms, 100 ms, and 534 ms (satellite environment). All three TCP protocols were tested in this environment. Our results are summarized in Table 1. [1]

Also an OC-12 (622 Mbps) satellite link with a BER of $8.9 * 10^{-4}$ was evaluated. The throughput obtained for this case was for TCP Reno was 1.435 Mbps, TCP New Reno 1.527 Mbps, and TCP SACK 1.959 Mbps.

[1] A network with a BER in the range of 10^{-12} has an error almost every two hours. Therefore, all TCP extensions will deliver near optimal performance results. The throughput differences among TCP Reno and TCP New Reno or TCP SACK is caused by operating system kernel issues rather than network related issues.

	RTT(ms)	Throughput(Mbps) (BER=$0.8 * 10^{-7}$)	Throughput(Mbps) (BER=$1.3 * 10^{-8}$)	Throughput(Mbps) (BER=$1.1 * 10^{-9}$)	Throughput(Mbps) (BER=$1.0 * 10^{-12}$)
R	1	88.409	89.563	92.132	119.838
E	50	10.063	48.925	91.417	114.058
N	100	6.299	24.884	76.124	113.435
O	534	1.533	5.132	16.342	109.717
New	1	113.422	127.337	129.998	126.178
RE	50	10.043	33.018	96.181	126.023
N	100	6.015	22.949	60.715	125.745
O	534	1.678	6.518	23.112	121.237
S	1	117.584	124.387	129.934	128.876
A	50	10.115	33.525	100.587	128.634
C	100	6.548	25.372	71.557	127.539
K	534	2.111	7.508	26.563	121.268

Table 1: TCP (Reno, New Reno, SACK) performance results over various BDP channels with four different level of probabilities of errors.

4 Discussion on Results

The results shown in Table 1 indicate that on noisy channels, throughput degradation is severe on high BDP connections. On the other hand, the effect of errors is less critical on throughput over low delay channels. This is due to the fact that the sender can very fast ramp up and complete slow start or congestion avoidance after continuous segment drops. If the probability of error on the channels gets better, high delay connections obtain better throughput than before, but still is lower than that obtained by low delay connections with the same channel error probability. Also, these results indicate that none of the TCP extensions can recover on links with high probabilities of bit error on high delay connections.

TCP does not differentiate on losses caused by congestion or errors due to noisy paths. Every time there is a segment loss, the congestion algorithms are activated. This reaction results to unnecessary TCP window reduction and performance degradation on wireless or satellite links, where there is high probability of channel random errors. This is shown by the results (Table 1) obtained from the scenario where the effects of noisy channels on TCP performance were investigated over various BDP connections. One can observe by looking at the results of Table 1 that as the BER increases on a high delay communication channel, throughput drops dramatically. Throughput degrades on low delay high BER connections as well, but not as much as it does on high delay paths. This throughput degradation is due to the TCP dynamics and it occurs with all three tested TCP implementations. TCP SACK though, because of its ability to recover from multiple lost segments in one RTT, obtains better performance than TCP New Reno and TCP Reno in all cases of tested channel BER and BDP connections. The same happens when comparing TCP New Reno with TCP Reno. The performance improvement of TCP SACK and TCP New Reno over TCP Reno over the satellite channel with the tested BERs is shown in Table 2. As it can be seen from the results of Table 2, as the BER on the communication channel drops, the performance improvement of TCP New Reno and TCP SACK over TCP Reno increases. This is due to the fact that on heavy noisy channels the TCP extension algorithms (SACK, New Reno) will not be able to recover from the so many and frequent segment losses as quickly as they would have done on channels with lower BER.

BER	New Reno Impr. over Reno (%)	SACK Impr. over Reno (%)
$8.9 * 10^{-4}$	6.4	36.5
$0.8 * 10^{-7}$	9.4	37.7
$1.3 * 10^{-8}$	27.0	46.3
$1.1 * 10^{-9}$	41.42	62.5

Table 2: TCP New Reno and TCP SACK performance improvement over TCP Reno over a satellite channel (RTT=534 ms) under four tested channel BERs.

Independently from the percentages of performance improvements given in Table 2, the throughput obtained by each TCP implementation is very low compared with the channel capacity (optimal throughput), in all cases. This can be observed in Table 3, which shows the throughput obtain over the satellite system with the four tested BERs as a percentage of the optimal throughput (which is 134.513 Mbps [12] in our network architecture).

5 Conclusions

In this paper we studied the effect of various noisy BDP connections on TCP performance and we tested three transport protocol implementations: TCP Reno, TCP New Reno, and TCP SACK.

Our results show that in cases where noisy (with very high BER) high speed satellite links are

BER	TCP Reno Mbps/% Opt.	TCP New Reno Mbps/% Opt.	TCP SACK Mbps/% Opt.
$8.9 * 10^{-4}$	1.435/1.06	1.527/1.13	1.959/1.45
$0.8 * 10^{-7}$	1.533/1.14	1.678/1.24	2.111/1.57
$1.3 * 10^{-8}$	5.132/3.81	6.518/4.84	7.508/5.58
$1.1 * 10^{-9}$	16.342/12.15	23.112/17.18	26.563/19.75

Table 3: TCP Reno, TCP New Reno, and TCP SACK throughput obtained over a satellite channel (RTT=534 ms) under four tested channel BERs as a percentage of the optimal throughput (134.513 Mbps).

established, throughput obtained by TCP Reno, TCP New Reno, and TCP SACK end systems is very low (about only 1% of the optimal throughput), showing that none of the tested TCP protocols are able to recover under these conditions. In such cases, TCP SACK delivers 36.5% performance improvement over TCP Reno, and TCP New Reno delivers 6.4% performance improvement over TCP Reno. As the channel BER drops, the performance improvement, throughput, and channel utilization of TCP SACK and TCP New Reno over TCP Reno increases. In all cases, TCP SACK is more efficient than TCP New Reno and TCP Reno, because of its ability to recover from multiple losses faster than the other two protocols under test.

References

[1] Van Jacobson, Robert Braden, David Borman. TCP Extensions for High Performance, May 1992. RFC 1323.

[2] Matthew Mathis, Jamshid Mahdavi, Sally Floyd, Allyn Romanow. TCP Selective Acknowledgment Options, October 1996. RFC 2018.

[3] W. Richard Stevens. TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, January 1997. RFC 2001.

[4] Janey C. Hoe. Improving the Start-up Behavior of a Congestion Control Scheme for TCP. Proceedings of ACM SIGCOMM '96, August 1996.

[5] Douglas E. Comer. Internetworking with TCP/IP, Volume I, Principles, Protocols, and Architecture. Prentice Hall, 3rd Edition, 1995.

[6] W. Richard Stevens. TCP/IP Illustarted, Volume I: The protocols, Addison Wesley, 1994.

[7] Jon Postel. Transmission Control Protocol, September 1981. RFC 793.

[8] Rohit Goyal, Raj Jain, Shiv Kalyanaraman, Sonia Fahmy. UBR+: Improving Performance of TCP over ATM-UBR service. Proceedings of ICC '97, IEEE ICC '97, Montreal, Canada, June 1997.

[9] L. DaSilva, J. B. Evans, D. Niehaus, V. S. Frost, R. Jonkman, B. Lee, G. Lazarou. ATM WAN Performance Tools, Experiments, and Results. IEEE Communications Magazine, August 1997.

[10] S.M. Bajaj, C. Brazdziunas, D.E. Brooks, D.F. Daly, S.M. Srinidhi, T. Robe, F. Vakil. Performance Characterization of TCP/IP-On-ATM over an ATM/SONET High Data Rate ACTS Channel. Proceedings of the 16th AIAA International Communications Satellite Systems Conference, February 1996.

[11] D. Hoder, B. Kearny. Design and Performance of the ACTS Gigabit Satellite Network High Data-Rate Ground Station. Proceedings of the 16th AIAA International Communications Satellite Systems Conference, February 1996.

[12] C.P. Charalambous, G.Y. Lazarou, V.S. Frost, J. Evans, R. Jonkman. Experimental and Simulation Performance Results of TCP/IP over High-Speed ATM over ACTS. Proceedings of 1996 IEEE International Conference on Communications, Atlanta, Georgia, June 1998.

[13] Charalambos P. Charalambous. Performance Evaluation and Analysis of TCP on ATM over High Bandwidth Delay Product Networks. M.Sc Thesis, August 1998.