Effect of Channel Errors on TCP Throughput over Satellite Links

Daniel Brodbekier, Felipe Mathias and João Célio Brandão

Abstract: **Satellite transmission is impaired by large delays and relatively high occurrence of channel errors compared to terrestrial transmission over optical fiber. These problems may impact seriously the throughput achieved in a data transmission using Transport Control Protocol (TCP) since channel errors cause packet losses which imply packet retransmission and lowering data transmission rate. In this paper, we compare the throughput degradation of different versions of TCP due to channel errors in a typical satellite link. Among the TCP versions, particular attention is given to Satellite Transport Protocol (STP) which has been specifically developed for satellite transmission. In addition to random statistically independent errors, we also consider error bursts generated in Viterbi decoders of convolutional codes. The evaluation is carried through simulation using the software packages Matlab and ns-2 [15].**

Index Terms: **TCP/IP, Satellite Communications, Internet, Burst Errors**

I. INTRODUCTION

Following the impressive development of Internet, TCP/IP standard has progressively becoming the general platform for all kind of telecommunications services in different networks. Concerning TCP, several versions of the protocol have been developed in order to improve the network performance in different transmission media.

The enormous distance between the earth and the satellite makes it difficult for satellite transmission to achieve the same performance of the one observed in terrestrial optical fiber transmission. If there is not much to do concerning the long delay, on the other hand, several techniques have been discussed and proposed to improve TCP throughput [1-4]. Among a large number of possible mechanisms, IETF RFC 2488 [2] recommends some basic techniques to be used in satellite links. In addition to standard TCP mechanisms Slow Start and Congestion Avoidance, Fast Retransmit, Fast Recovery, Initial Large Window and Selective Acknowledgement are recommended. Also, error correcting codes are considered a fundamental and important enhancing technique in the physical layer for lowering the bit error rate.

Packet losses, TCP retransmission, Congestion Avoidance and Slow Start mechanisms interact to produce throughput degradation in long distance links as follows. When a packet is lost, it is retransmitted but, at the same time the sender reduces the amount of transmitted data in order to avoid congestion. Due to these mechanisms (Slow Start and Congestion Avoidance), the recovering of previous data rate takes long time and degrades the throughput.

Slow Start and Congestion Avoidance mechanisms are very important congestion control techniques if packet losses are caused just by buffer overflow. However, as it is discussed in [3, 4] there is no reason to use these mechanisms when a packet is discarded due to channel errors. Unfortunately an efficient method to distinguish between these two types of loss is not known [5].

In this paper we evaluate the throughput degradation due to channel errors in a TCP transmission over a satellite link. The performance of the main versions of the TCP protocol and a recent one called STP [6] are compared in presence of statistically independent random errors and burst errors. The burst errors are generated in Virtebi decoders of convolutional codes which are supposed to be part of the transmission system. The results allow one to assess the impact of channel errors in TCP throughput over satellite links and consequently the importance of searching a more efficient congestion control mechanism which is able to identify the loss type and avoid the unnecessary reduction of transmission rates.

TCP throughput degradation in different scenarios has been evaluated in several papers [7-10]. Our work is closely related to [7] and [9]. In [7] the error statistics of some convolutional codes are obtained and TCP throughput evaluated for a typical implementation of just one version of the protocol. In [9] the authors compare different versions of TCP in wireless links subject to burst errors caused by propagation phenomena. We have addressed a satellite link considering the same TCP versions and including STP.

The rest of the paper is organized as follows. Section II presents a statistical analysis of the error pattern observed in some convolutional codes. Section III describes the transmission system and simulation model. Section IV presents the simulation results and Section V the conclusions.

Daniel Brodbekier, Nokia e-mail brod@cetuc.puc-rio, João Célio Brandão, CETUC-PUC/Rio, e-mail: jcelio@cetuc.puc.rio.br

II. CONVOLUTIONAL CODES AND BURST ERRORS

Typical satellite transmission systems may employ convolutional coding in order to improve error performance and reduce transmission power requirements and antenna sizes in the ground segment. The Viterbi algorithm is generally used for sequence decoding and is known to produce errors in burst at the decoder output. The error burst patterns produced at the output of a Viterbi decoder depend on the coder parameters and are also influenced by details on the decoder implementation, such as the length of the decode path memory [11,12].

Generic models, relying on few statistical averages such as the mean burst length and the mean number of errors per burst may be used to model burst errors. More precise description may be obtained analytically from the coder trellis diagram or by simulation techniques.

In our study we have used the software Matlab to simulate some convolutional codes with Viterbi decoder. A pseudo random source of bits has been generated, convolutional encoded and fed into a module that introduces random errors. After decoding, traces of error sequences are stored for analysis and used as input to ns-2 network simulator.

Table I presents the average burst length for a set o convolutional codes after processing the data obtained from the described simulations. The Viterbi decoder data window length was set as five times the value of the code constraint length.

TABLE I
PARAMETERS OF SIMULATED CONVOLUTIONAL CODES

Generator Polynomial	Rate	constraint length	Burst mean length
$[15\ 13]_{oct}$	1/2	K = 4	2.7 bits
$[171\ 133]_{oct}$	3/4	K = 7	5.0 bits
$[171\ 133]_{oct}$	1/2	K = 7	5.4 bits
$[171\ 133\ 145]_{oct}$	1/3	K = 7	5.7 bits
$[4335\ 5723]_{oct}$	1/2	K = 12	16.1 bits

III. TCP VERSIONS

Window adaptation and loss recovery mechanisms based in slow start and congestion avoidance algorithms in TCP are well known and are used in all TCP versions since Old Tahoe TCP. Later versions called Tahoe, Reno and New Reno included the following modifications.

Tahoe

In addition to timeout triggering of retransmission, a fast retransmit procedure is implemented for loss detection. With fast retransmission, the sender enters the loss recovery phase when K duplicated ACK's are received.

Reno

In Reno, fast retransmit procedure is followed by fast recovery, a mechanism that avoids the slow start. Only the lost packet is retransmitted and, instead to be reset to 1, the congestion window is halved and increased by one segment at each received duplicated ACK.

New Reno

Fast retransmit and fast recovery have proved to be efficient when just one packet loss occurs inside a window but not in presence of multiple losses. Seeking to address this situation, New Reno uses fast retransmit and congestion window adaptation as in Reno. However, instead of sending just the lost packet, New Reno continues the transmission of subsequent packets like Tahoe.

Sack

TCP Sack refers to the use of selective acknowledgement, i.e. a multiple ACK from the receiver requiring all the lost packets until that instant, instead of just the packet of lower order. Then the sender will retransmit the required packets.

STP

In STP [6], the sender periodically requires to the receiver the ACK's of all received information so losses are known immediately by the receiver. The combination of these two strategies provides a lesser return traffic in the link when it has few losses and a fast acknowledgement in presence of a loss event. On the other hand, the protocol requires that the transmitter stores all the information transmitted until the ACK is received.

STP has two options of congestion control, with or without rate control. In the latter case, congestion control is very similar to the TCP standard mechanism. With rate congestion control, a maximum value of transmission rate is defined and the packets to be retransmitted are inserted in the buffer with high priority.

IV. TRANSMISSION SYSTEM MODEL

Figure 1 shows the block diagram of the simulated transmission system where the link distance of 250 ms characterizes the satellite link. The application is an unlimited file transfer using FTP. The TCP parameters are listed in Table II.

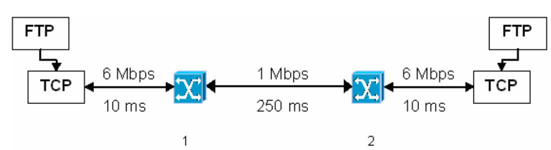

Fig. 1 - Simulation model

TABLE II – TCP PARAMETERS

Initial value of *cwnd*	1 packet
Initial value *ssthresh*	70 packets
Maximum value *cwnd*	70 packets
Maximum l value *acwnd*	70 packets
Packet Size	1000 Bytes

Two error channel models are considered for the satellite link. The first one assumes statistically independent random errors and the other considers burst errors according to the error pattern of a convolutional code obtained as described in Section II.

A packet is considered to be lost when at least one error is observed in this packet. This is the standard criterion used in TCP to define packet corruption and to request its retransmission [13, 14].

For statistically independent random errors, packet losses are also statistically independent and are implemented by a routine in the ns-2 library. In this case, packet loss *p* and bit error rate (BER) are related by

$$p = 1 - (1 - BER)^n \tag{1}$$

where n is the number of bits per packet.

For burst errors generated at the Viterbi decoder of a convolutional code, the bit error traces obtained in Matlab simulations are previously mapped to packet error traces and then input to ns-2.

V. PERFORMANCE COMPARISON

Several simulation experiments have been carried to assess the throughput achieved with the transmission system and TCP versions described previously. Curves of throughput versus BER are shown in Figs 2 to 6.

Fig. 2 shows the simulation results for a channel with statistically independent random errors. It can be noticed that the error occurrences affect drastically the performance for BER values above 10^{-8}, with the throughput reducing to half when BER is around 10^{-7}. It can also be observed that the performance of the different TCP implementations are practically the same except for TCP Tahoe which performs lightly worse.

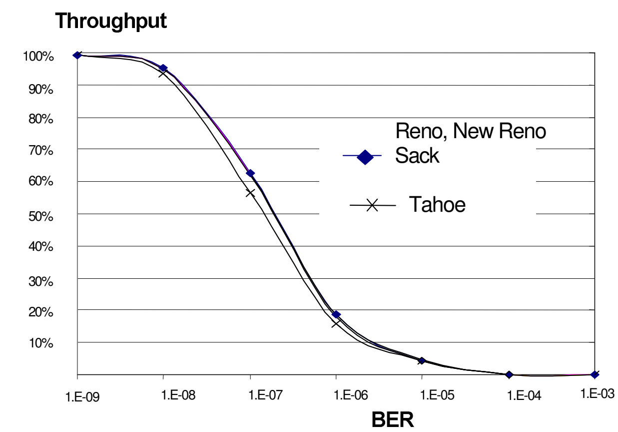

Fig. 2 - TCP throughput in presence of statistically independent random errors

The performance in the presence of burst errors generated by the [133,171] convolutional code with constraint length K = 7 is shown in Figs 3 and 4 where the code rates are ½ and ¾, respectively. These two set of curves are very similar and show that TCP has a better performance in the presence of burst errors compared to the one obtained with statistically independent errors. Again, the throughputs reached by all the TCP versions are very close.

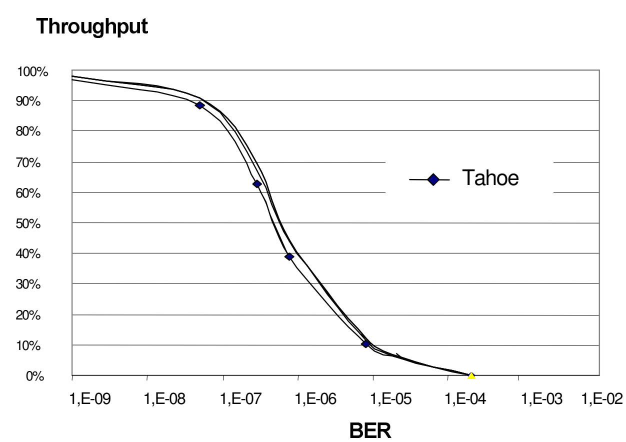

Fig. 3 TCP throughput in the presence of burst errors generated by rate ½ convolutional code

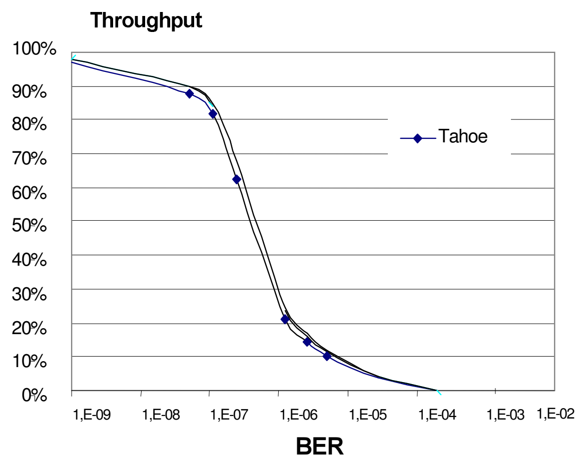

Fig. 4 - TCP throughput in the presence of burst errors generated by the rate ¾ convolutional code.

STP Performance

Fig. 5 shows simulation results for STP with rate congestion control and maximum transmission rate set to 960Kbps. Statistically independent random errors (SIE) and burst errors (BE) generated by [133,171] convolutional code with constraint length K = 7 and code rate ½ are considered. Results for the standard TCP versions Tahoe and Sack are plotted for comparison.

The results show that STP largely outperforms the other protocols, due to the rate congestion control. The use of rate congestion control makes it possible a high throughput even for large error rates. Consider for example BER = 10^{-4} which corresponds to a packet loss rate of p = 0.45 in the case of statistically independent errors (see eq.1). Under this condition, STP reaches a throughput of 14% as appearing in Fig. 5 whose results are in good agreement with the ones reported in [10]. Values larger than 80% are obtained

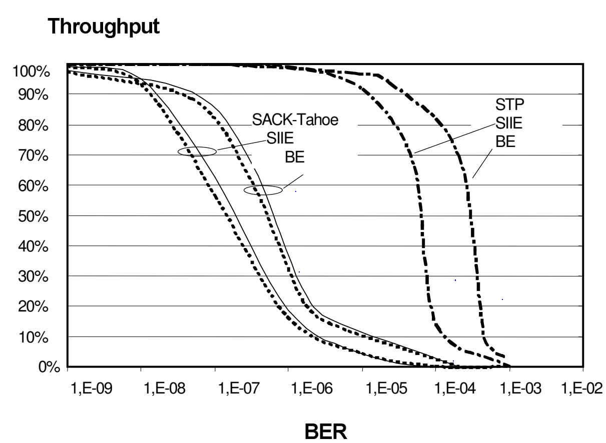

Figure 5 - Performance of STP, TCP Tahoe and TCP Sack with Statistically Independent Errors (SIE) and Burst Errors (BE)

Fig. 6 shows the performance of TCP and STP considering that each packet is protected by an error correcting code with a capability of one bit correction. The horizontal axis of Fig. 6 corresponds to the error rate before the bit correction. In this situation we can see that all the protocols (Sack, Tahoe and STP) have their performance significantly improved in the presence of statistically independent errors but only a marginal improvement is observed in the presence of burst errors. This may be expected since the effectiveness of the correcting code is reduced by the typical occurrence of more than one bit error inside each packet.

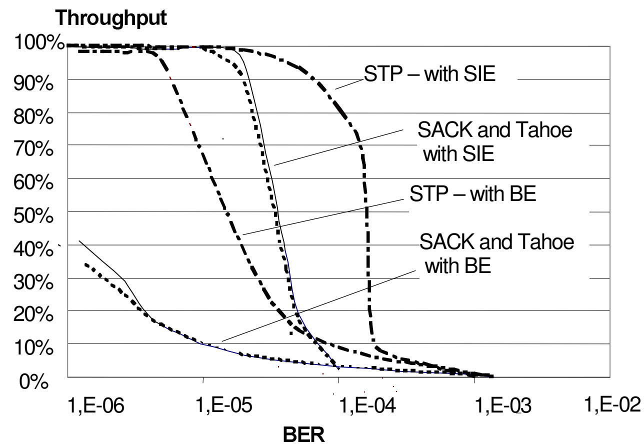

Fig. 6 - Performance of STP and other TCP versions with one bit correction in presence of Statistically Independent Errors (SIE) and Burst Errors (BE)

V. CONCLUSION

In this paper, the throughput of TCP transmission over satellite channels has been investigated. Standard TCP versions have been compared to STP – an enhanced TCP protocol to operate on satellite links – in the presence of two type of channel errors: statistically independent and burst errors produced by convolutional codes.

The results have shown that burst errors are beneficial to TCP throughput if packets are discarded in the event of one or more bit errors inside the packet. However, the opposite occurs when the packets are protected by an error correcting code.

In any case, it has been observed that the standard TCP versions have a similar performance and STP improves enormously the performance of satellite transmission in the presence of channel errors.

REFERENCES

[1] IETF RFC 2760 "On going TCP research related to satellites".

[2] IETF RFC 2488 "Enhancing TCP over satellite channels using Standard mechanisms".

[3] Nasir Ghani, Sudhir Dixit, "TCP/IP Enhancements for Satellite Networks", IEEE Communications Magazine, July 1999, pp 64-72.

[4] Ian F. Akyildiz, Giacomo Morabito and Sergio Palazzo, "Research Issues for Transport Protocols in Satellite IP Networks", IEEE Personal Communications, June 2001, pp 44-48.

[5] RFC 3155 "End-to-end performance implications of links with errors".

[6] T. Henderson and R. Katz, "Satellite Transport Protocol (STP): An SSCOP-based Transport Protocol for Datagram Satellite Networks," *Proceedings of 2nd Workshop on Satellite-Based Information Systems (WOSBIS'97)*, 1997.

[7] J. R. Heissler, Y. A. Barsoum, R. Condelho "An analysis of the viterbi decoder error statistics for ATM and TCP/IP over satellite communication". IEEE, MILCOMM 1999

[8] M. Zorzi e R. R. Rao, "Effects of correlated errors on TCP", Proc. 1997 CISS, March 1997, pp. 666-671.

[9] M. Zorzi, A. Chockalingam e R. R. Rao "Throughput Analysis of TCP on Channels with Memory" , IEEE Journal on Selected Areas in Communications, Vol 18, No 7, July 2000, pp 1289-1300.

[10] T. R. Henderson, R. H. Katz, "Transport Protocols for Internet-compatible satellite networks". IEEE Journal on selected areas in communications, vol. 17, no. 2, Feb. 1999.

[11] A. Franchi & R.A.Harris, "On the error burst properties of the viterbi decoding", ICC 1993.

[12] A. Franchi & R.A.Harris, "On the error burst properties of the "standard" K=7, rate ½ convolutional code with soft-decision viterbi decoding", Telecommunication Systems, Vol.6, No3, May-June June 1995, pp 337-351

[13] IETF RFC 1071, "Computing the Internet checksum".

[14] IETF RFC 1141, "Incremental updating of the Internet checksum".

[15] http://www.isi.edu/nsnam/ns.