TCP/IP OVER WIRELESS LINKS: PERFORMANCE EVALUATION

Anna Calveras Augé, Josep Paradells Aspas
Departament de Matemàtica Aplicada i Telemàtica
Universitat Politècnica de Catalunya
Barcelona, Spain
e-mail: {acalveras, teljpa}@mat.upc.es

ABSTRACT

Transmission Control Protocol (TCP) is a large extended transport protocol, mainly used in fixed networks. It works well in those networks in which the main cause of packet loss is congestion. Some mechanisms included in TCP optimise their use in this kind of networks. Today, Internet users need access it over wireless links in order to get this service anywhere they are. However, TCP does not work well in wireless environments where losses are due to high Bit Error Rates (BER). In this paper, we evaluate the end-to-end throughput of a TCP connection with a wireless link. First, we demonstrate the impact of high BER in TCP performance. Then, we evatuate the impacts of header compression.

Keywords: Internet, Mobile Communications, TCP/IP.

I. INTRODUCTION

Data transmission services over mobile networks are one of the telecommunication sectors that are due to experiment an important expansion in the near future. The potential demand on data mobile services is growing so much that is forcing the telecommunication operators to supply this kind of communications.

The use of applications such as e-mail, and in general access services to Internet, is driving portable users to demand them through the mobile link. Internet, the biggest computer network, with millions of terminals, is based on the TCP/IP protocol architecture. The most used applications are e-mail, ftp, telnet and access to WEB's, based all of them on the TCP transport protocol [Ste94, RFC793], which is end-to-end, reliable and connection oriented. It is known that 80% of the whole traffic of the wide area networks is TCP traffic [CDJ91].

TCP/IP protocols were designed to be used in fixed networks and it were never thought that they would be applied in a new environment with very different characteristics: the mobile radio channel. That is why a very determining aspect when evaluating data services based on TCP/IP over mobile networks, is the analysis of the interaction between this protocol architecture and this new communications link.

Since losses of packets in nowadays fixed networks are mainly due to congestion, modern TCP implementations include congestion avoidance mechanisms to react against congestion [JaK88, Flo95, BrP95]. However, in mobile environments packet losses are frequent, due to high BER (Bit Error Rate) and handover processes. Therefore, congestion control algorithms that TCP activates when errors occur may lead to a little efficient protocol behaviour, effects of which are to be evaluated.

There are several studies on the aspects related to the TCP performance over wireless links [YaB94, BaB94, CaI95, BSK95, MyS93, DCY93, BBK96, CTG97, BBK97].

In this paper we analyse different aspects of the TCP over these environments. With this aim, we have used a real implementation of the TCP/IP protocol, encapsulated in PPP (Point to Point Protocol) [RFC1548] and a simulation application emulating the mobile radio channel transmission.

This paper is organised as follows. First we describe the experimental platform. In section 3, we present some results of TCP over wireless environments. In section 4, we discuss the Van Jacobson TCP/IP header compression mechanism. We conclude by presenting a summary of results and conclusions.

II. EXPERIMENTAL PLATFORM

This study is based on the evaluation of a real TCP/IP implementation, Linux 1.2.13, which is based on [RFC793]. It includes the Slow Start and Congestion Avoidance mechanisms [JaK88], and it does not include the Fast Retransmit and Fast Recovery Algorithms [Ste94]. Table 1 shows the features of different implementations and the comparison among them [Ste94, BrP95, FaF95].

The experimental platform is shown in Fig. 1. The TCP/IP connection is initiated using the ftp application in the PC that emulates de mobile host. Data are encapsulated in PPP [RFC1548], and the Channel Simulator introduces errors in the forward and/or the backward direction. Finally another PC receives data emulating a fixed host. The connection is traced using the tcpdump program.

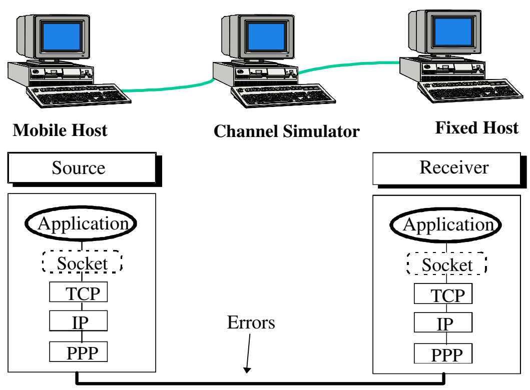

Fig. 1.- Experimental Platform and protocol Architecture.

	Linux v.1.2.13	*Tahoe*	*Reno*	*Vegas*
Clock Resolution	10 ms	500 ms		system dependent
RTT frequency sample	every segment	once per window		every segment
time-out resolution	every 10 ms	every 500 ms		
time-out activation	last segment sent	one segment per window		
Complementary error detection	ACK's are used to compare the retransmission timer expiration	Fast Retransmit	Fast Retransmit and Fast Recovery	modified Fast Fetransmit and Fast Recovery
cwnd (Congestion Control Window) after congestion in segments	1		1 (time-out) or cwnd/2 (fast retransmit and fast recovery)	1 (time-out) or cwnd/2 or cwnd (fast retransmit and fast recovery)
Congestion Control	Slow Start and Congestion Avoidance			Preventive mechanism

Table 1.- TCP implementations comparison.

The Channel Simulator introduces different types of distribution of errors (uniform, deterministic and Poisson), and the Fritchman [Fri67, WaM95] and temporal model. The link rate ranges from 110 bps to 19200bps, which covers the maximum transmission rate in different wireless networks.

Next section discusses the reaction of TCP over this platform, emulating a wireless link, having special importance the different mechanisms that uses TCP to guarantee an error free end-to-end connection.

III. TCP IN MOBILE WIRELESS ENVIRONMENT

The TCP/IP protocol architecture was designed to be used in fixed networks with high Bit Error Rates (BER). Afterwards, the evolution of fixed networks, in which the main problem is congestion and not the errors, made the evolution of the protocol stack in order to work well in congestion situations [Ste94, BrP95, Flo95, JaK88]. In these cases, the protocol was optimised to activate algorithms against congestion in case a lost is detected.

In wireless environment, the main cause of lost data are high BER. And a very important aspect is to analyse how TCP reacts to them.

Impairments in a wireless link result on bursts of errors that have a bad effect in the performance of the transport protocol, and is the main responsible for the difficult adaptation of the TCP over wireless links.

Burst of errors have been modelled by the Fritchman model. After a transfer of 100 Kbytes with a MTU of 296 bytes and without TCP/IP header compression [RFC1144], the obtained throughput for the connection is shown in the Fig. 2. The maximum achievable throughput for this connection is 960 bytes/s.

Fig. 3 shows the effects in the TCP segment number, and in RTO (Retransmission Time-Out) of the TCP connection. In this case the BER is 1/50000. For deterministic errors, lost packets are rapidly detected by means the positive acknowledgement of TCP. Nevertheless, in the other case, errors affect one TCP transmission window, and it can be seen the large period of inactivity which produces the performance degradation. Specially, the RTO evolution in case of burst of errors shows the exponential back-off activation [Ste94], which slows down the TCP transmission rate.

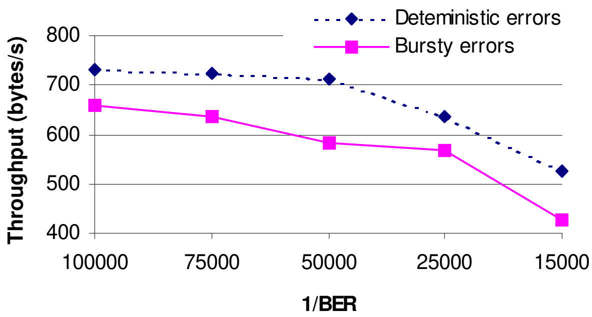

Fig. 2 .- Throughput evolution.

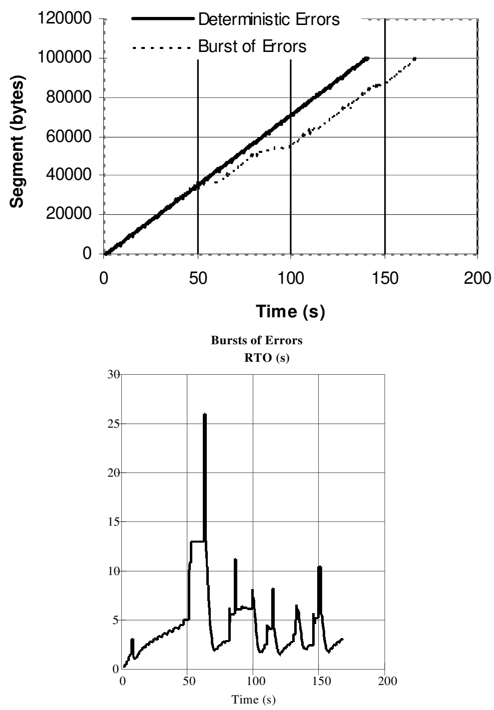

Fig. 3.- Sequence number (above) and RTO evolution (below).

In conclusion, the mechanisms that recover from congestion situations, are not the most suitable in case of bursty errors. TCP is highly affected by errors since TCP slows down traffic injection in case of errors. Burst of errors produces worst performance degradation as errors affect some TCP segments in the same transmission window, which produces the activation of the RTO exponential backoff, causing laps of time in the data transmission. In any case, the performance degradation of the throughput has to be evaluated.

IV. VAN JACOBSON COMPRESSION

As bandwidth is limited in wireless channels, a great effort has to be done in order to optimise its use. One method to decrease the used bandwidth is to reduce the headers of the network protocols that are sent to the channel. In the particular case of the study, we have evaluated the Van Jacobson algorithm [RFC1144] to reduce the TCP/IP headers.

The advantages of the header compression are:

- The reduction of the response time in interactive applications. For instance, in *rlogin* applications each character is sent independently in one segment. If we consider a rate of 960 bytes/s, and a TCP/IP header of 55 bytes, the echo of a single character is about 117 msec which is higher than the maximum acceptable in interactive applications (100 msec). If we use the Van Jacobson header compression, the TCP/IP header is reduced to 3 or 5 bytes, and then the response time is about 8 msec.

- Header compression contributes to independence the segment size and the achievable throughput for a connection, which is important when interactive and bulk data transmissions are concurrent. That is:

$$throughput = \frac{UI}{UI + NS*HL}*BW =$$

$$= \frac{UI}{UI + \left\lceil \frac{UI}{MSS} \right\rceil * HL}*BW \xrightarrow[HL \ll MSS]{} BW$$

UI: Useful Information, **HL**:TCP/IP header length (in average), **NS**: Number of sent segments, **MSS**: Maximum TCP Segment Size, **BW**: Bandwidth

- As in wireless environments the MTU is small due to the need of minimise the Packet Loss Probability (PLP), header compression seems to be appropriate to reduce the packet loss and optimise the bandwidth usage. Fig. 4 shows the PLP for different BER when the MTU for the link of 296 bytes in case we use compression or not.

- Moreover, the new version of the TCP/IP Protocol (IPv6) has a header length of 60 bytes [Tho96] due to the increase in IP addresses. Also, IP includes some options to allow host mobility and facilitate datagram routing, which increases the TCP/IP header length [DEN96].

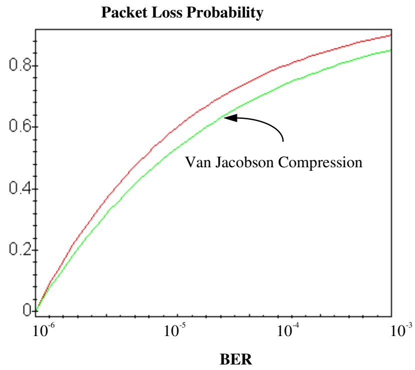

Fig. 4.- PLP with and without header compression.

Consequently, Van Jacobson header compression can be appropriate in TCP/IP over wireless links, and it is the objective of this section to evaluate its effects over TCP performance.

A. Ftp Transfer Without Errors

First of all, we have evaluated a ftp transmission without errors. Fig. 5 shows that the Van Jacobson header compression makes a gain of about 20% in the communication efficiency so that there is a throughput of 902 bytes/s with compression and 746 bytes/s without it. This justifies the explained above. In this case the MTU is 296 bytes and the effective link rate is 960 bytes/s.

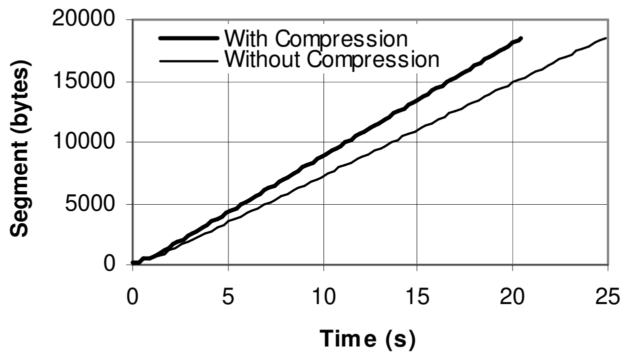

Fig. 5.- Ftp connection evolution without errors.

B. Ftp Transfer With Errors

In case of errors, the Van Jacobson compression has to be evaluated in detail.

While this method reduces the PLP, the method on itself makes a dependence between each segment and the previous one. This dependence supposes a subjective increase on the PLP, because a segment loss produces the loss of the whole TCP transfer window. Fig. 6 shows these effects.

Moreover, since compression header makes the retransmission of the whole transmission window in case of errors, its size at the moment errors occur is a determinant aspect in TCP Throughput. This parameter has two effects:

- Firstly, lost packets are detected by the sender with the RTO expiration, and its value increases as it does the transmission window when the transmission time (which is link rate dependent) is appreciable. For instance, if the link MTU is 296 bytes, and the effective link rate is 960 bytes/s, the Round Trip Time (RTT) is about 360 msec, and adding the propagation time, the RTT is about 430 msec.

With these figures in mind, the appropriate maximum transmission time for this connection is about 2 segments. Nevertheless, most of TCP implementations use bigger maximum transmission windows (that is the case of the used implementation), what means that packets are buffered at the output port. Therefore, RTT (and in consequence RTO) increases. Fig. 7 represents the RTT and RTO evolution for a TCP/IP/PPP connection.

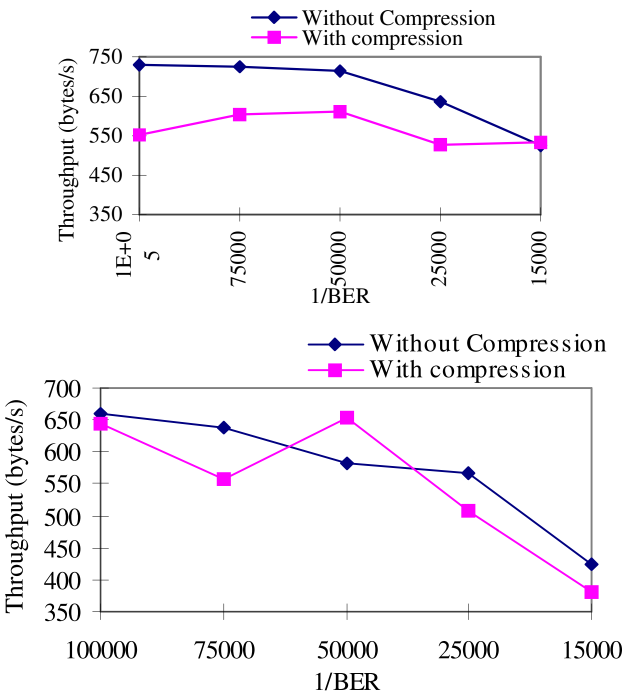

Fig. 6.- TCP Throughput evolution. Deterministic errors (above), burst of errors (below).

- Secondly, one or more errors per transmission window imply the retransmission of all unacknowledged packets.

Additionally, the last important aspect in the Van Jacobson header compression effects is that the transmission window size when errors occur depends on the BER of the channel. Therefore, there is a compromise between BER and window size: if BER increases throughput decreases but the achievable window size is small and the negative effects of the Van Jacobson header compression are less relevant.

Finally, in case of burst of errors the use of compression does not degrade performance, since when using compression a simple error produces the retransmission of the whole transmission window.

Fig. 8 shows the retransmitted bytes that correspond to the transfer of Fig. 6 for the case of deterministic errors. Without compression, obviously, the retransmitted bytes grow as BER does, but with compression, this does not occur. The main cause of this evolution is that when errors occur the whole window is retransmitted so its size is determinant in each case.

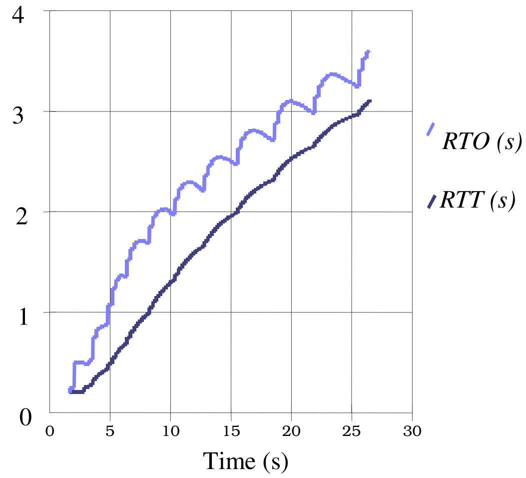

Fig. 7.- RTT and RTO evolution. MTU=296 bytes, Link rate=960 bytes/s, without errors.

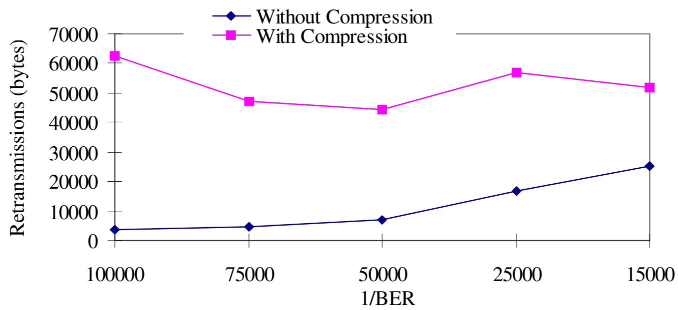

Fig. 8.- Retransmitted information. Deterministic errors.

V. CONCLUSIONS

TCP/IP in mobile networks suffers significant performance degradation due to high BER. The main causes are the protocol design issues, that are oriented to solve network congestion situations.

When errors occur, TCP reacts against them slowing down traffic injection. This reaction is appropriate in fixed networks, but it is not in wireless networks, since the transmission is not active during laps of time (waiting for congestion to disappear) avoiding the recuperation of the transmission. Moreover, RTO (Retransmission Time-Out) values of TCP are critical, because they determinate the length of the communication pauses due to errors. This is especially evident in wide area mobile networks where RTO reaches high values due to interleaving and low transmission rates.

The use of a compression method of the TCP/IP encapsulated information may have positive effects when the used communications link is the mobile radio channel. Packet size reduction decreases the response time in interactive applications, reduces packet loss probability and increases the bandwidth efficiency.

With this in mind, we incorporate the header compression algorithm with the objective of determining its use or not in this type of networks.

The obtained results show that the present implementation of the algorithm is not recommended to be used in channels with high BER, since each error leads to the retransmission of all the transmission window segments.

Moreover, the transmission window size when errors occur determines the achieved Throughput. The time required to detect each error and the time that TCP spends in recovering from losses (retransmitting the affected segments) are to be studied.

VI. ACKNOWLEDGEMENTS

This work was founded by CICYT Project TIC97-0942.

REFERENCES

[BaB94] A. Bakre, B. Badrinath, "I-TCP: Indirect TCP for mobile hosts", Technical Report DCS-TR-314, Rutgers University, October 1994.

[BBK96] P. Bhagwat, P. Bhattacharya, A. Krishna, S. Tripathi, "Enhancing throughput over wireless LANs using Channel State Dependent Packet Scheduling", IEEE INFOCOM, 1996.

[BBK97] P. Bhagwat. P. Bhattacharya, A. Krishna, S. Tripathi. "Using channel state dependent packet scheduling to improve TCP throughput over wireless LANs". Wireless Networks 3 (1997) 91-102.

[BrP95] L. Brakmo, L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet", IEEE Journal on Selected Areas in Communications, vol.13 no.8 October 1995.

[BSK95] H. Balakrishnan, S. Seshan, R. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks", ACM Wireless Networks, December 1995.

[CaI95] R. Caceres, L. Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments", IEEE Journal on Selected Areas in Communications, vol.13 no.5 June 1995.

[CDJ91] R. Cáceres, P. Danzig, S. Jamin, D. Mitzel, "Characteristics of Wide-Area TCP/IP Conversations", ACM SIGCOM '91.

[CTG97] A. Chan. D. Tsang, S. Gupta. "TCP (Transmission Control Protocol) over Wireless Links".47th VTC'97 1326-1330.

[DCY93] A. DeSimone, M. Chuah, O. Yue, "Throughput Performance of Transport-Layer Protocols over Wireless LANs", Proceedings Globecom'93, December 1993.

[DEN96] M. Degermark, M. Engan, B. Nordgren, S. Pink, "Low loss TCP/IP Header Compression for Wireless Networks". Proceedings of the ACM Mobicom'96 Conference, Rye, New York, November 10-12, 1996. (cdt.luth.se/micke/low-loss-hc.ps.Z)

[FaF95] K. Fall, S. Floyd. "Simulation-based comparisons of Tahoe, Reno and SACK TCP". ftp.ee.lbl.gov/papers/sacks.ps.Z, December 1995.

[Flo95] S. Floyd, "TCP and Successive Fast Retransmits", Technical Report, Lawrence Berkeley Laboratory, May 1995.

[Fri67] B. Fritchman, "A Binary Channel Characterisation Using Partitioned Markov Chains", IEEE Transactions on Information Theory, vol.13 no.2. April 1967.

[JaK88] V. Jacobson, M. J. Karels, "Congestion Avoidance and Control", November 1988.

[MyS93] A. Myles, D. Skellern, "Comparison of Mobile Host Protocols for IP", Technical Report, Macquarie University in Sydney (Australia), April 1993.

[RFC1144] V. Jacobson, "Compressing TCP/IP Headers for Low Speed Serial Links", RFC 1144.

[RFC1548] W. Simpson, "The Point-to-Point Protocol (PPP)", RFC 1548.

[RFC793] "Transmission Control Protocol", RFC 793.

[Ste94] W. Stevens, "TCP/IP Illustrated, Vol. 1: The Protocols", New York: Addison-Wesley, 1994.

[Tho96] S. A .Thomas, "Ipng and the TCP/IP Protocols", John Wiley & Sons, 1996

[WaM95] H. Wang, N. Moayeri, "Finite State Markov Channel. A Useful Model for Radio Communication Channels", IEEE Transactions on Vehicular Technology, pag.163-171, February 1995.

[YaB94] R. Yavatkar, N. Bhagwat, "Improving end-to-end performance of TCP over mobile internetworks", Workshop on Mobile Computing Systems and Applications, December 1994.