?Proposal of a Modified Version of the Slow Start Algorithm to Improve TCP Performance over Large Delay Satellite Channels

Mario Marchese
CNIT - Italian National Consortium for Telecommunications
DIST - Univ. of Genoa Research Unit, University of Genoa, Italy
Via Opera Pia 13, 16145 Genova (Italy)
Phone: +39-010-3532985
Fax: +39-010-3532154
e-mail: mario.marchese@cnit.it

Abstract - **The paper presents a study and a performance analysis of a modified version of the TCP (Transmission Control Protocol) over a GEO (Geostationary Orbit) satellite link. The Round Trip Time (RTT) is above 500 ms. TCP has not been designed for these network characteristics: the high delay to receive acknowledgements decreases the performance and makes the quality perceived by the users really poor. The performance of TCP may be improved by properly tuning some parameters and modifying algorithms. The behaviour of the protocol with a modified slow start algorithm is investigated in the paper on the base of previous work concerning variations of the buffer length and of the initial congestion window. Configurations that drastically improve the performance (measured by the throughput in bytes/s and by the overall transmission time) are proposed. A real test-bed, composed of two remote hosts connected through a satellite channel is adopted to obtain the results. The analysis has included both the single application case and the multiple application case, where several connections share the satellite link at the same time.**

I. INTRODUCTION

The interest in delivering TCP/IP services over satellite is strongly increasing. Unfortunately, the network used heavily affects the behaviour of the protocols [1]. An issue of particular importance in this environment is the performance of the transport protocol (TCP, in the treated scenario). The problem of improving TCP [2] over satellite has been investigated in the literature for some years: see [3] among the first works about long fast networks, [4] for a first overview on the topic and [5] for a more specific study in TCP/IP networks with high delay per bandwidth product and random loss. More recently, reference [6] provides a summary about improved TCP versions as well as issues and challenges in satellite TCP; reference [7] highlights the ways in which latency and asymmetry impair TCP performance; reference [8] lists the main limitations of the TCP over satellite and proposes many possible methods to act. Reference [9] represents, at the best of the author's knowledge, the more recent tutorial paper on the topic. References [10] and [11] contain an extensive analysis of the TCP behaviour by varying parameters as buffer length and initial congestion window and they are considered the starting point for this paper. Many works, as [12] and [4], propose architectures to provide Internet services via satellite and introduce relay entities to split the TCP connection and to isolate specific parts of the network that deserve a particular attention concerning the transport level (e.g. satellite links). Many national and international programs and projects (listed extensively in [10]) in Europe, Japan and USA concern satellite networks and applications. In particular, some of them, or part of them, are aimed at improving the performance at the transport level. NASA ACTS (summarised in [13] and [14]), ESA ARTES-3 [15] and CNIT-ASI [16], which supports the present work, deserve a particular attention, among many others.

The paper focuses on a GEO system with a large delay per bandwidth product and symmetric channel. There are some ways to improve TCP over satellite; a possibility is acting on the protocol parameters and algorithms to tune and modify them so to mitigate the negative effect of the satellite channel and to improve the performance. That is the choice performed in this paper. The paper focuses the effect of modifying the slow start algorithm after tuning the initial congestion window and the buffer length and analyses the protocol behaviour and the performance. Even if the tests are performed on a particular network and the numerical results strictly depends on the type of network, the design methodology introduced is not affected.

The paper is structured as follows. Section II contains the slow start algorithm proposed. Section III describes the test network and the application used to get the results. The results are contained in section IV. Section V presents the conclusions.

II. THE SLOW START ALGORITHM PROPOSED

A. Slow start main characteristics

The short summary in the following is intended to focus on the TCP characteristics that concern the paper. The parameters are substantially set by following the standard in [17] and [18]. The notation used herein has been introduced in [17]. The slow start phase begins setting:

? the congestion window (cwnd) to 1 segment (1·smss, where smss, measured in bytes, stands for Sender Maximum Segment Size);

? the slow start threshold (ssthresh) to a very high value (infinite).

A parameterisation of the initial congestion window (IW) for satellite links is followed in the paper by setting cwnd=IW?smss. At each received acknowledgement

?This work has been supported by the Italian Space Agency (ASI) under the CNIT contract ASI-ARS-99-72 "Integration of Multimedia Services on Heterogeneous Satellite Networks".

0-7803-7097-1/01/$10.00 ©2001 IEEE

(ack), cwnd is increased by 1·smss (i.e. "ACK ? cwnd=cwnd+1·smss"). Other important TCP algorithms as congestion avoidance, fast retransmit and fast recovery are untouched. The Selective Acknowledgement (SACK) mechanism is utilised [18].

B. The modified version

The paper proposes a parameterisation of the slow start increase function "ACK ? cwnd=cwnd+1·smss" by introducing a function F(·), which depends on the number of received acknowledgements (# of received acks) and on the current dimension of the congestion window (cwnd). In short: ACK ? cwnd=cwnd+F (# of received acks, cwnd)·smss. The function F(·) is aimed at regulating the size of the congestion window in the slow start phase. The characteristics of F(·) affect the increase of the window and, as a consequence, the transmission speed and the protocol performance. The definition of F(·) is not trivial and many considerations may affect the decision: the choice performed in this paper is aimed at increasing the transmission speed in the initial phase without entering a congestion period. The behaviour has been tested with different types of functions. The increment of cwnd strictly depends on the current value of the cwnd itself and on number of received acknowledgements. The choice allows to tune the behaviour of the protocol depending on the congestion window, and to measure, at some extent, the network status represented by the arriving acks. Let the variable N_ack the number of received acknowledgements in a single TCP connection.

The reference TCP sets the function

$$F(N\_ack)=1 \qquad (1)$$

The alternatives chosen set the function F(N_ack) as follows:

a)
$$F^{'}(N\_ack) ? K \qquad (2)$$

At each acknowledgements received, the increment is constant.

b)
$$F^{''}(N\_ack) ? \begin{array}{lll} ? & ?N\_ack & \text{if cwnd ? thr} \\ ? & ?1 & \text{otherwise} \end{array} \qquad (3)$$

The increment is linear up to the value 'thr' of a fixed threshold; it is constant after this value. This method is referenced as "Linear thr" in the results presented to simplify the notation (i.e., if thr=20, the method is identified as "Linear 20"). It is important to note that, when the increment is linear, the protocol behaviour is very aggressive: if no loss is experienced, the number of received acknowledgements as shown in (4) rules the size of cwnd.

$$cwnd(N\_ack)=cwnd(N\_ack-1)+ N\_ack \cdot smss \qquad (4)$$

c)

$$F^{'''}(N\_ack) ? \begin{array}{lll} ?K_{thr_1} & ?N\_ack & \text{if cwnd ? thr}_1 \\ ?K_{thr_2} & ?N\_ack & \text{if cwnd ? thr}_2 \\ ?K_{thr_3} & ?N\_ack & \text{if cwnd ? thr}_3 \\ ? & ? & ? \\ ? & ? & ? \\ ?K_{thr_n} & ?N\_ack & \text{if cwnd ? thr}_n \\ ? & 1 & \text{otherwise} \end{array} \qquad (5)$$

In this case a variable number of thresholds (i.e. thr_n, where? **N**) may be used. Function (5) is aimed at adapting the protocol behaviour through the constants (K_{thr_n} , if n thresholds are used). Three thresholds have been heuristically estimated to be a proper number to increase the rate in the first phase of the transmission and to smooth it on time. The function chosen appears as in (6).

$$F^{'''}(N\_ack) ? \begin{array}{lll} ?K_{thr_1} & ?N\_ack & \text{if cwnd ? thr}_1 \\ ?K_{thr_2} & ?N\_ack & \text{if cwnd ? thr}_2 \\ ?K_{thr_3} & ?N\_ack & \text{if cwnd ? thr}_3 \\ ? & 1 & \text{otherwise} \end{array} \qquad (6)$$

The notation used is $\text{Linear}(thr_1 ? thr_2 ? thr_3)$; the value of the constant K_{thr_n}, with $n ? ?1,2,3?$, represents the angular coefficient of the increase line, its value governs the speed of the increase and rules the TCP behaviour. The values of the constants will be identified in section IV, dedicated to the results.

III. Test network

The real test-bed is shown in Fig. 1: two remote hosts are connected through a satellite link by using IP routers. The data link level of the router uses HDLC encapsulation on the satellite side, where a serial interface is utilised, and Ethernet on the LAN side. The main characteristics of the Radio Frequency (RF) devices used for the tests are contained in [16]. A raw Bit Error Rate - BER (i.e., BER with no channel coding) approximately of 10^{2} has been measured; the utilisation of a sequential channel coding with a code rate of 1/2, to correct transmission errors, has allowed to reach a BER of about 10^{-8}. As a consequence, the higher layer protocol (the data link protocol) 'sees' a reliable channel.

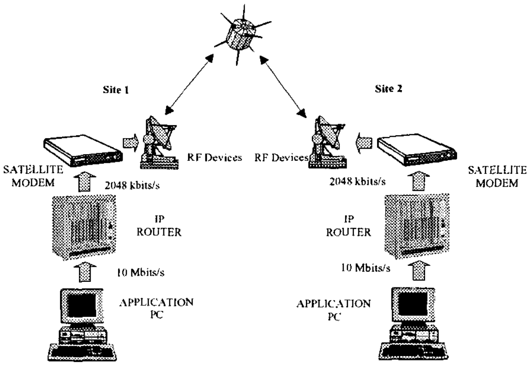

Fig. 1. Test-bed.

The system employs the ITALSAT II (13? EST) satellite, providing a countrywide coverage in the Ka-band (20-30 GHz), which is currently explored for the provision of new services. Each satellite station can be assigned a full-duplex dedicated traffic channel with a bit-rate ranging from 32 kbits/s to 2 Mbits/s (this latter adopted for the tests) and it is made up of the following components: Satellite Modem, Radio Frequency (RF) Device, IP Router, Application PC. The software in the router along with the queues' dimension and managing has not been modified. Even if optimisation about it may be performed, the router has been considered as a sort of

0-7803-7097-1/01/$10.00 ©2001 IEEE

black box in this work. The Application PC is the source of the TCP/IP traffic under test and contains the modified version of the TCP. The application used to get the results is a simple ftp-like one, i.e. a file transfer application located just above the TCP. It allows transferring data of variable dimension (H [bytes] in the following) between the two remote sites; the work has taken this application as a reference because it is thought as fundamental for most of the applications of interest. The study has considered two different application scenarios:

? Single connection

? Multiple connections

The first implies that only one connection a time is in the network; from the practical viewpoint it may represent a file access in a database of a small private network where just one customer, or very few customers, access the network at the same time. Another example may be a file of commands required by a remote control system. Speed is essential in both cases.

The second scenario is very common: it is representative of a typical web access where many clients a time access the information available on the web.

IV. Results

The work presented takes as an operative base the configurations resulted as more efficient in [10] and [11], where the value of the buffer length (buf) and of the initial congestion window (IW) have been tuned to adapt the protocol to the satellite channel. The configuration (buf=320 kbytes - IW=6), which implies K=1 because the slow start algorithm is not varied, has shown the best performance and allowed to avoid congestion even in case of multiple connections. The gain (concerning the overall transmission time, with respect to the TCP version commonly used, taken as a reference) is above 71% for a single TCP connection in the network if a file of about 2.8 Mbytes is transmitted.

The gain of a generic value T with respect to a reference value T_{REF} is computed in percentage as follows: $\text{Gain} ? \begin{cases} \dfrac{T_{REF} ? T}{T_{REF}} ?100, \text{if } T ? T_{REF} \\ 0, \qquad \text{otherwise} \end{cases}$.

The analysis presented in the paper concerns the modification of the slow start increase function F() introduced in section II.

Fig. 2 shows the throughput by varying the value K in $F^{'}(?)$. The throughput at the beginning of the connection is high and the overall transmission time drastically decreases: a transmission time of 49.21 s of the reference case can be compared with a value of 12.65 s of the (IW=6 - buf=320 kbytes - K=4) case. The gain of the overall transmission time is 74.3 %, for K=4 and 73.1% for K=2. The best result of previous studies (IW=6 - buf=320 kbytes, without tuning the slow start algorithm) guaranteed a gain of 71.6%. Fig. 2 shows also the risk: values over K=4 make the intermediate router unable to manage the entering traffic and the network congested.

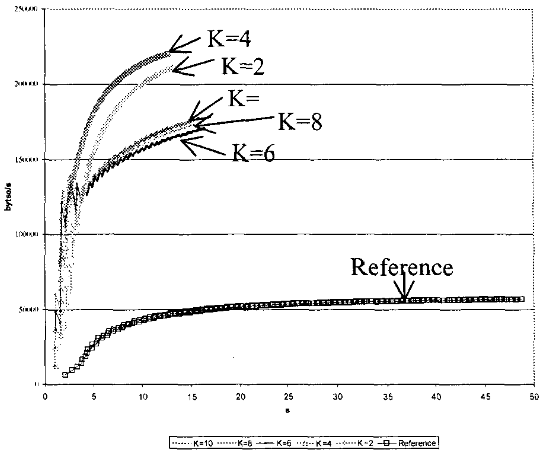

Fig. 2. Throughput vs time, different values of K, H=2.8 Mbytes.

Fig. 3 contains the multi-connection case. Only K=2 gives a permanent gain with respect to the reference configuration and the performance is substantially equivalent to the results provided by the configuration (IW=6 – buf=320 kbytes), except for a very low traffic load. It is important to say that, even if only K=2 may have a wide application because it guarantees gain both in the single and the multiple connection case, the gain obtained for K=4 in the single connection case should not to be neglected. As already stated, the presence of just one TCP connection a time is frequent for some applications.

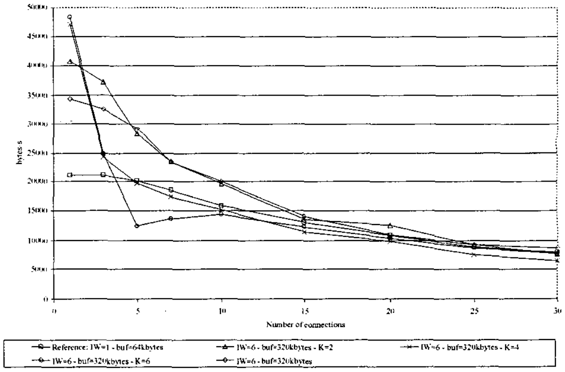

Fig. 3. Throughput vs time, different values of K, multi-connection case.

An attempt to smooth the effect of the variable K is the introduction of function $F(?) ? F^{''}(?)$. The behaviour of $F^{''}(?)$ is reported in Fig. 4. The value thr of the threshold assumes values 10, 20, 30 and 50, identified, respectively, as Linear 10, Linear 20, Linear 30 and Linear 50. The gain ranges from about 72% of Linear 10 to 74 % of Linear 50, with respect to the reference configuration. Linear 20 and Linear 30, substantially overlapped, have a gain of about 73 %. The performance of $F^{'}(?)$ and $F^{''}(?)$ (Fig. 2 and Fig. 4) is substantially equivalent but a little mistake of tuning in $F^{'}(?)$ can make the network enter a congestion status that drastically reduces the performance.

0-7803-7097-1/01/$10.00 ©2001 IEEE

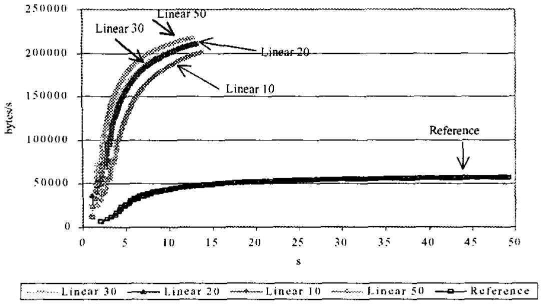

Fig. 4. Throughput vs time, different thr, $F''(?)$, H=2.8 Mbytes.

The behaviour for shorter file transfers (not reported in the paper) shows that the configuration K=2 approximately offers the same performance of the less aggressive $F''(?)$. Higher values of K offer better performance but, even if congestion is not entered for shorter transfers, $F''(?)$ should be preferred because it is safer concerning congestion and it is suited also for longer files. The performance of $F''(?)$ in the multi-connection case is reported in Fig. 5: the throughput versus the number of connections for different values of the threshold is shown along the reference version of TCP. It is important to note the efficiency of the proposed schemes (Linear 20, in particular) for a number of connections lower than 15 and the correct use of the available bandwidth, except for Linear 50, which is too aggressive.

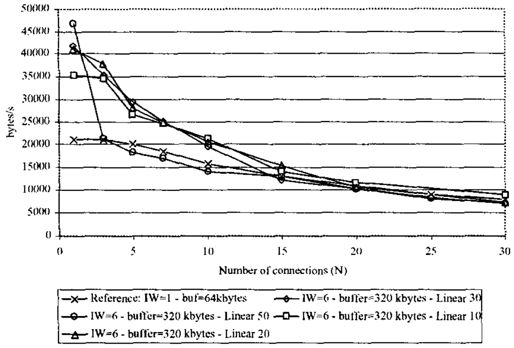

Fig. 5. Throughput vs time, $F''(?)$, multi-connection case.

The results obtained by using function $F'''(?)$ are shown in Fig. 6. The angular coefficients K_{thr_1}, K_{thr_1}, K_{thr_3} assume, respectively, the values 4, 2, 1 while the thresholds have been set to 10-20-30 and 20-30-40. The gain with respect to the reference configuration is above 74% for both the configurations in the single connection case. A gain above 74% means that the transfer time for a file of about 2.8 Mbytes is reduced from a value of about 50 s to a value ranging from 12 s to 13 s. The advantage in the quality of service perceived by a user who uses, for instance, a web browsing via satellite is clear.

The behaviour in the multiple connection case is reported in Fig. 7 for the configuration Linear 20-30-40 that shows a very satisfying behaviour. The advantage introduced by the new algorithm is maintained up to a relevant number of connections in the network.

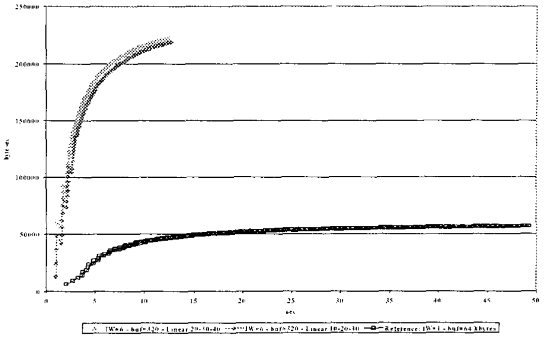

Fig. 6. Throughput vs time, $F'''(?)$, H=2.8 Mbytes.

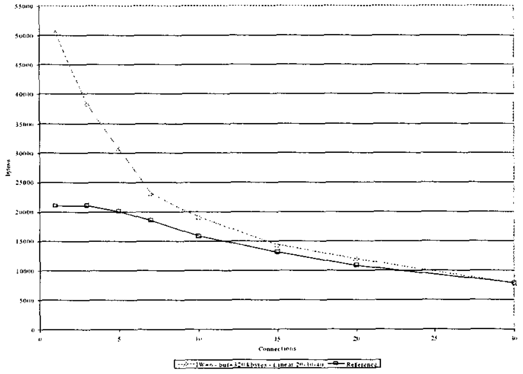

Fig. 7. Throughput vs number of connections, $F'''(?)$, multi-connection case.

Table I summarises the results concerning the overall transfer time for the configurations resulted from the analysis in the single connection case. The gain is really good for any configuration and it arises up to 74.5 % if function $F'''(?)$ is utilised with $thr_1=20$ - $K_{thr_1}=4$, $thr_2=30$ - $K_{thr_2}=2$, $thr_3=40$ - $K_{thr_3}=1$.

TABLE I

OVERALL TRANSFER TIME AND THROUGHPUT, DIFFERENT INCREMENT FUNCTIONS, H=2.8 MBYTES

TCP Configuration	Transfer Time [s]	Throughput [kbytes/s]	Gain [%]
Reference: IW=1 - buf=64	49.21	56.7	-
IW=6 - buf=320	13.96	199.8	71.6
IW=6 - buf=320 - K=2	13.22	210.9	73.1
IW=6 - buf=320 - K=4	12.65	220.3	74.3
IW=6 - buf=320 - Linear 10	13.81	201.9	71.9
IW=6 - buf=320 - Linear 20	13.13	211.9	73.3
IW=6 - buf=320 - Linear 50	12.80	217.8	74.0
IW=6 - buf=320 - Linear 20-30-40	12.57	221.9	74.5

The configurations providing the best results when more connections are routed have been selected and shown in Fig. 8. The use of configuration Linear 20-30-40 is very efficient for a limited number of connections and, at the same time, it allows avoiding congestion when the load increases. All the modified configurations are substantially equivalent if the number of active connections ranges from 7 to 15. After this latter value there is no gain in using a modified version of TCP.

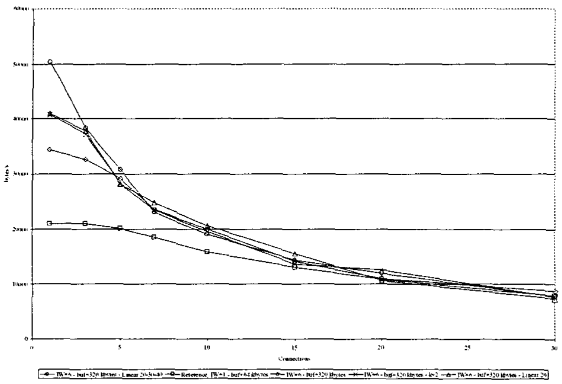

Fig. 8. Throughput vs number of connections, different slow start algorithm, multi-connection case.

V. Conclusions

The paper has presented a new version of the slow start algorithm of TCP to better adapt to large delay satellite links. The TCP version, which is commonly used in cable networks, taken as a reference in the paper, is inefficient.

The behaviour is tested and discussed for new versions of the slow start algorithm: the improvement is satisfying. A gain above 74% is reached with respect to the reference configuration, when only a connection a time is routed in the network. The analysis has shown that the best solutions proposed maintain a satisfying gain also in a multi-connection environment. The new algorithms, tested in a real satellite test-bed, have been implemented by modifying the Linux operating system and are operatively used.

References

[1] M. Allman, D. Glover, L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanism", IETF, RFC 2488, January 1999.

[2] Information Sciences Institute, University of Southern California, "Transmission Control Protocol - Darpa Internet Program - Protocol Specification", IETF, RFC 793, September 1981.

[3] S. Pink, "TCP/IP on Gigabit Networks", Measurement, Implementation and Optimization of Internet Protocols, Ph.D. dissertation, Royal Institute of Technology, Sweden, October 1993.

[4] C. Partridge, T. J. Shepard, "TCP/IP Performance over Satellite Links", IEEE Network, Vol. 11, n. 5, September/October 1997, pp. 44-49.

[5] T.V. Lakshman, U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss", IEEE/ACM Transactions On Networking, Vol. 5, No. 3, June 1997

[6] N. Ghani, S. Dixit, "TCP/IP Enhancement for Satellite Networks", IEEE Comm. Mag., July 1999, pp. 64-72.

[7] T. R. Henderson, R. H. Katz, "Transport Protocols for Internet-Compatible Satellite Networks", IEEE Journal on Selected Areas in Communications (JSAC), Vol. 17, No. 2, February 1999, pp. 326-344.

[8] M. Allman, S. Dawkinks, D. Glover, J. Griner, T. Henderson, J. Heidemann, S. Ostermann, K. Scott, J. Semke, J. Touch, D. Tran, "Ongoing TCP Research Related to Satellites", RFC2760, February 2000.

[9] C. Barakat, E. Altman, W. Dabbous, "On TCP Performance in a Heterogeneous Network: A Survey", IEEE Comm. Mag., Vol. 38, No. 1, January 2000, pp. 40-46.

[10] M. Marchese, "TCP Modifications over Satellite Channels: Study and Performance Evaluation", International Journal of Satellite Communications, Volume 19, Issue 1, 2001, pp. 93-110.

[11] M. Marchese, "Study and Performance Evaluation of TCP Modifications and Tuning over Satellite Links", Proc. IEEE International Conference on Communications (ICC2000), New Orleans, Lousiana, June 2000, pp.129-133 (also on ICC2000 CD ROM).

[12] V.G. Bharadwaj, J. S. Baras, N. P. Butts, "Internet Service via Broadband Satellite Networks", Center for Satellite and Hybrid Communication Networks Technical Report, CSHCN T.R: 99-11, also in http://www.isr.umd.edu/TechReports/CSHCN/1999 (last access February 19th, 2001).

[13] D. E. Brooks, C. Buffinton, D. R. Beering, A. W. William D. Ivancic, M. Zernic, D. J. Hoder, "ACTS 118x Final Report, High Speed TCP Interoperability Testing", NASA/TM - 1999-209272, July 1999.

[14] R. T. Gedney, R. Schertler, F. Gargione, "The Advanced Communications Technology Satellite", SciTech Publishing, Mendham, NJ, 2000.

[15] ARTES 3, The ESA Multimedia Initiative, Home Page, http://telecom.estec.esa.nl/artes/artes3/ (last access February 19th, 2001).

[16] D. Adami, A. Conti, D. Dardari, M. Marchese, L. S. Ronga, "CNIT-ASI Project 'Integration of Multimedia Services on Heterogeneous Satellite Networks': An Overview", Proc. Fifth Ka Band Utilization Conference, October 1999, Taormina, Italy, pp. 219-226.

[17] M. Allman, V. Paxson, W. S. Stevens, "TCP Congestion Control", IETF, RFC 2581, April 1999.

[18] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm", IETF, RFC 2582, April 1999.