ELSEVIER

Computer Communications 24 (2001) 877–888

computer communications

www.elsevier.com/locate/comcom

Performance analysis of the TCP behavior in a geo satellite environment

M. Marchese*

CNIT — Italian National Consortium for Telecommunications, DIST — University of Genoa Research Unit, Via Opera Pia 13, 16145 Genova, Italy

Received 6 November 2000; accepted 6 November 2000

Abstract

The paper shows the main problems that the Transmission Control Protocol (TCP) meets in a Geo Stationary Orbit (GEO) satellite environment characterized by high 'delay per bandwidth product'. In a GEO environment, the delay in the delivery of a message is very high, the Round Trip Time (RTT) is above 500 ms. These characteristics heavily affect the acknowledgement mechanism on which the TCP is based and the performance of the protocol is much lower than in cable networks.

The paper proposes and analyses possible solutions aimed at mitigating the negative effect and at improving the performance.

A real test-bed is used to carry out the study: two remote hosts are connected together through a satellite link in the Ka-band (20–30 GHz) by using IP routers. The system has been tested by using a ftp-like application, that allows transferring data of variable size between the two hosts. This is due to the observation that most of the Internet multimedia applications, as browsing and distance learning, use massive file transfer.

The protocol behavior is investigated both by tuning the parameters buffer size and initial congestion window and by modifying the dynamic characteristics of the slow start algorithm. The analysis itself allows suggesting solutions and taking decisions.

Keywords: TCP/IP; Internetworking; Satellite environment

1. Introduction

The opportunities offered by technologies like optical fibers, by the improved speed of devices and by the advanced network protocols allow the development of very complex multimedia applications. The new applications are designed not only for stand-alone PCs or workstations, but they need to run in an extended network environment. In this context a great interest has arisen in these last few years to connect Local Area Networks (LANs) by using satellite or terrestrial/satellite networks. The satellites have an inherent broadcast capability, they can connect remote sites when there is no terrestrial infrastructure, as in rural areas, and, at the same time, they can provide high-speed links. Due to their characteristics, they can represent an efficient way to provide efficient interconnections and multimedia services.

Many national and international projects (listed extensively in Ref. [13]) in Europe, Japan and USA concern satellite networks and applications. In particular, some of them, or part of them, are aimed at improving performance at the transport level. NASA ACTS [8,11], ESA ARTES-3 [5] and Italian National Consortium for Telecommunications (CNIT)–Italian Space Agency (ASI) [1], which supports the present work, deserve specific attention, among many others.

CNIT–ASI is a project aimed at analyzing the problems related to a satellite or terrestrial/satellite interconnection concerning both transmission and network problems. It is funded by the ASI, and carried out by the CNIT, a research center composed of several Italian universities. It is divided into two integrated lines: an experimental activity of multimedia services over a terrestrial-satellite network and a study activity for the system evolution concerning network protocols, integration of satellite and cable networks, medium access techniques, resource allocation, development of terminal equipment, and user interfaces. The project CNIT–ASI uses the ITALSAT satellite, works at 2 Mbits/s with an antenna of 1.8 m in the Ka-band (20/30 GHz).

The paper focuses on a Geo Stationary Orbit (GEO) system with a large delay per bandwidth product and symmetric channel. Transmission Control Protocol (TCP) works quite efficiently over 64 or 128 kbytes GEO systems; the problem arises if the packets are transported at high speed and the network introduces a large latency. In a geo

* Tel.: +39-010-353-2806; fax: +39-010-353-2154.
E-mail address: mario.marchese@cnit.it (M. Marchese).

0140-3664/01/$ - see front matter
PII: S0140-3664(00)00361-3

Table 1
Slow start algorithm

Slow start [cwnd ≤ ssthr]	cwnd = 1·smss ssthr = ∞ ACK → cwnd = cwnd + 1·smss

stationary system the Round Trip Time (RTT) is above 500 ms. The high delay to receive acknowledgements, on which the TCP is based, together with the large bandwidth used, makes the protocol inefficient [2] and the quality perceived by the users really poor. For instance, the delay of remote login or file retrieval in a satellite environment may be unacceptable for the user. On the other hand, in GEO systems there are also positive aspects: the RTT is approximately constant and the connectivity is guaranteed. The transmission errors measured are very low, at least in the test-bed used where error rates below 10^{-8} have been measured, and, as a consequence, the TCP does not fail to interpret each non-arrived packet as congestion presence. Ref. [4] contains a discussion about the TCP need to distinguish loss due to transmission errors from loss due to network congestion.

The object of the paper is the investigation of the TCP behavior and the proposal of some modifications to improve performance.

The topic has been investigated in the literature for some years: Ref. [15] contains a first overview. A more specific study in TCP/IP networks with high delay per bandwidth product and random loss may be found in Ref. [12]. More recently, Ref. [9] lists the issues and the challenges in satellite TCP and Ref. [10] highlights the ways in which latency and asymmetry impair TCP performance. Ref. [4] lists the main limitations of the TCP over satellite and proposes many possible methods to apply. A recent tutorial, which reports various improvements both at the transport level and at the application and network level, is Ref. [6]. This last paper focuses on the large delay per bandwidth product networks and suggests possible modifications to TCP, as the variation of the buffer size. The buffer size is also the object of the study [13], along with the initial congestion window. The solutions proposed in Ref. [13] are used also in this paper, which reports a new study about the effect of congestion and a new proposal of a modified version of the slow start algorithm. A preliminary version of the paper may be found in Ref. [14]. The modifications of the dynamic characteristics of the slow start algorithm are aimed at adapting the protocol to the characteristics of the channel. The performance analysis has been conducted experimentally by using a real test-bed composed of two hosts connected through a satellite link in the Ka-band (20–30 GHz).

The paper is structured as follows. Section 2 contains a short description of the TCP congestion control characteristics. Section 3 introduces a possible parameterization of the protocol and two proposals to modify the slow start algorithm. The experimental environment is described in Section 4. Section 5 contains the results and the observations about the performance analysis. Section 6 reports the conclusions.

2. TCP congestion control

The TCP functions mainly responsible of the TCP behavior over satellite channels are highlighted in the following.

A NewReno TCP is used under the 2.2.1 version of the Linux kernel. The parameters are substantially set following the standard in Refs. [3,7]. The notation used herein has been introduced in Ref. [3]. The behavior with no modifications has been monitored step by step.

The transmission begins with the *slow start phase*, where the congestion window (cwnd) is set to 1 segment (1·smss) where smss, measured in bytes, stands for sender maximum segment size. The slow start threshold (ssthresh) is set to a very high value (infinite). At each received acknowledgement (ack), cwnd is increased by 1·smss. If the value of cwnd is less than ssthresh, the system uses the slow start algorithm. Otherwise, the *congestion avoidance phase* is entered, where cwnd is incremented by (1·smss) at each RTT. More precisely, cwnd is increased by (1·smss) after receiving a number 'cwnd' of acknowledgements. If there is a loss, a packet is considered lost after four acknowledgements that carry the same number (duplicated acks), the system enters the *fast retransmit/fast recovery algorithm* and retransmits the missing segment without waiting for the retransmission timer to expire. In the TCP version used by the 2.2.1 Linux kernel, ssthresh was set to cwnd/2. The tests have been performed by setting ssthresh to the maximum between FlightSize/2 and 2·smss, as indicated in Ref. [3], where FlightSize is the measure (in bytes) of the amount of data sent but not yet acknowledged, i.e. the packets still in flight. Then the quantity cwnd is set to (ssthresh + 3·smss). When the error is recovered, i.e. when the lost packets have been successfully retransmitted, the value of cwnd is set to ssthresh. The real transmission window value is, in any case, the minimum between cwnd and the receiver's advertised window (rwnd), if the buffer space of the transmitter does not represent a bottleneck. In this latter case, the transmission buffer governs the transmission speed. The receiver window rwnd has been measured to be 32 kbytes at the beginning of the transmission being the receiver buffer space automatically set to 64 kbytes. The Selective Acknowledgment (SACK) mechanism is utilized [7].

The slow start algorithm holds the interest for this paper. Its features are summarized in Table 1.

This mechanism, where the congestion window grows in dependence on the received acknowledgements, takes a long time to recover from errors in a high 'delay per bandwidth product' environment An acknowledgement needs a long time to arrive. If, for example, just one segment was

Table 2
Parameterized slow start algorithm

Slow start [cwnd ≤ ssthr]	cwnd = IW·smss ssthr = ∞ ACK → cwnd = cwnd + *F* (# of received acks, cwnd)·smss

sent, it takes at least one RTT to be confirmed. The increase of the window is very slow, even in the slow start phase. This heavily affects the performance of the applications based on TCP.

The parameters and algorithms to tune:

- the transmitter/receiver buffer space (buf);
- the dimension of the initial congestion window (IW), i.e. the value taken by cwnd at the beginning of the connection;
- the slow start algorithm;
- the congestion avoidance scheme and the fast retransmit/recovery algorithm.

The paper studies the protocol behavior by varying the receiver/transmitter buffer space and the dimension of the initial congestion window, under different schemes to increase the congestion window in the slow start phase. The performance metrics considered are the throughput and overall time required for the transmission.

3. Definition of the increase function in the slow start algorithm

A possible parameterization of the TCP, aimed at evidencing the key points to improve the performance is shown in Table 2. The function $F(\cdot)$ is aimed at regulating the size of the congestion window in the slow start phase. The characteristics of $F(\cdot)$ affect the increase of the window and, as a consequence, the transmission speed and the protocol performance. The definition of $F(\cdot)$ is not trivial and many considerations may influence the decision. The choice made in this paper is aimed at increasing the transmission speed in the initial phase without entering a congestion period. The increment in cwnd strictly depends on the current value of the cwnd itself and on the number of the received acknowledgements, as indicated in Table 2. The choice allows tuning the behavior of the protocol in dependence on the congestion window, and to measure, at some extent, the network status represented by the arriving acknowledgements.

Let N_ack be the number of received acknowledgements. The reference TCP sets the function

$$F(\text{N\_ack}, \text{cwnd}) = 1 \tag{1}$$

Two different alternatives have been tested in the paper:

$$(a)\quad F'(\text{N\_ack}, \text{cwnd}) = K \tag{2}$$

At each acknowledgements received, the increment is constant.

$$(b)\quad F''(\text{N\_ack}, \text{cwnd}) = \begin{cases} \text{N\_ack} & \text{if cwnd} \le \text{thr} \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

The latter is the only function that really depends on the two variables, as introduced in Table 2. 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,

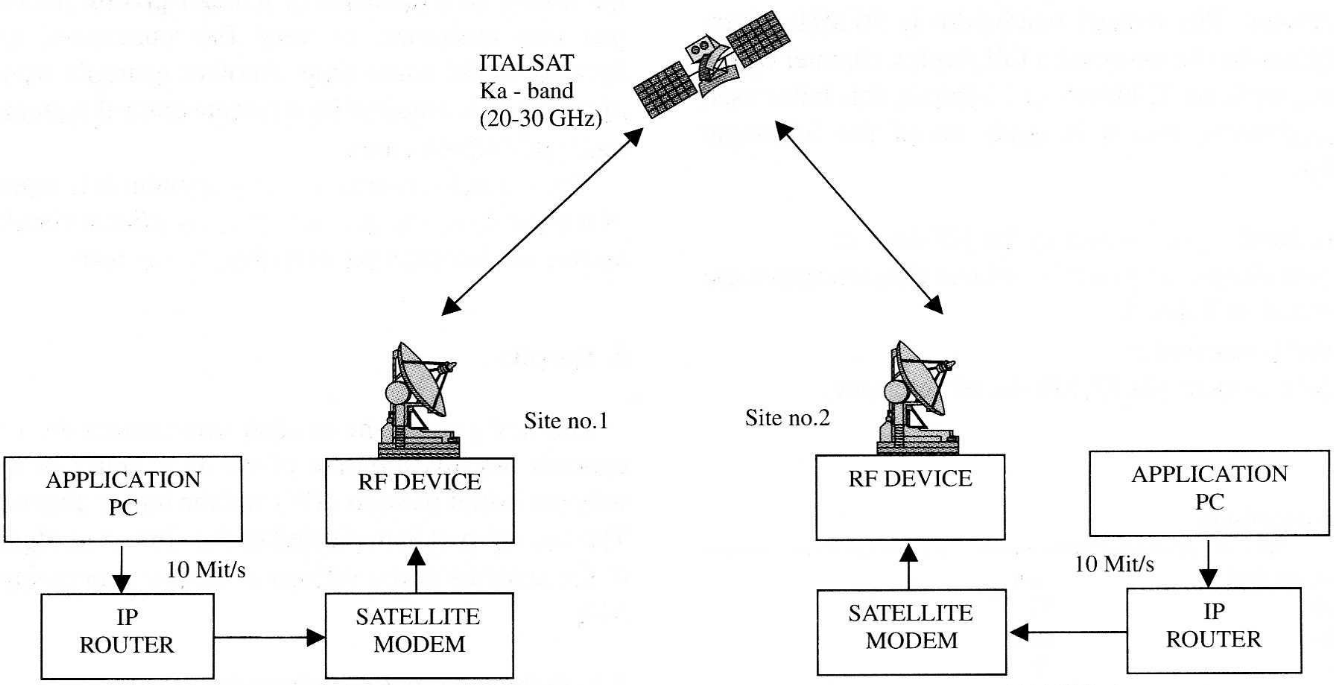

Fig. 1. Point-to-point IP satellite topology.

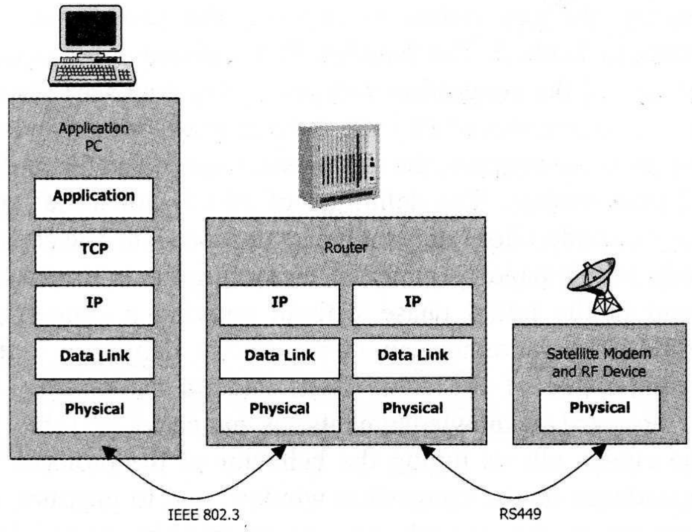

Fig. 2. Protocol stack.

the method is identified as "Linear 20"). It is important to note that, when the increment is linear, the protocol behavior is very aggressive. If no loss is experienced, the number of received acknowledgements, as shown in Eq. (4), rules the size of cwnd (cwnd is, actually, a function cwnd(N_ack) of the number of received acknowledgements)

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

4. Experimental environment

A real test-bed has been used. Two earth stations have been interconnected as shown in Fig. 1. The protocol stack is reported in Fig. 2.

The system employs the satellite ITALSAT II (13°EST). It provides coverage in the single spot-beam on Ka-band (20–30 GHz), which is currently explored for the provision of new services. The overall bandwidth is 36 MHz. Each satellite station can be assigned a full-duplex channel with a bit-rate ranging from 32 kbits/s to 2 Mbits/s, this latter used in the experiments, and it is made up of the following components:

- *Satellite Modem*, connected to the RF device;
- *RF (Radio Frequency) Device*, whose characteristics are summarized in Table 3;
- *IP Router* connected to:
  Satellite modem via RS449 Serial Interface,
  Application PC via Ethernet IEEE 802.3 10BASE-T link; and
- *Application PC:* PC Pentium III 500 MHz. They are the source of the service under test.

Table 3
RF device characteristics

Antenna diameter (m)	1.8
TX gain (dB)	52
RX gain (dB)	48.5
Polarization	V
TX center frequency (GHz)	29.75
RX center frequency (GHz)	19.95

Concerning the tests performed in this paper and reported in the following, a raw Bit Error Rate (BER) (i.e. the BER with no channel coding) approximately of 10^{-2} has been measured. The utilization of a sequential channel coding with a code rate of 1/2, to correct transmission errors, has allowed reaching a BER of about 10^{-8}. As a consequence, the higher layer protocol, i.e. the data link protocol, 'sees' a reliable channel, as should be evident in the results presented in Section 5. The data link level of the router uses HDLC encapsulation on the satellite side and Ethernet on the LAN side.

The software in the routers along with the queues' size and management has not been modified. Even if optimization about it may be performed, the router has been considered as a sort of 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, which 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 former 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 simultaneously access the information available on the web.

5. Results

The first part of the session summarizes the results that concern the investigation of the TCP behavior by varying only the initial window (IW) and the buffer dimension (buf). The second part is dedicated to the slow start algorithm and to the analysis of the various proposals concerning function $F(\cdot)$.

5.1. Buffer size and initial window

The buffer dimension for the source and destination is

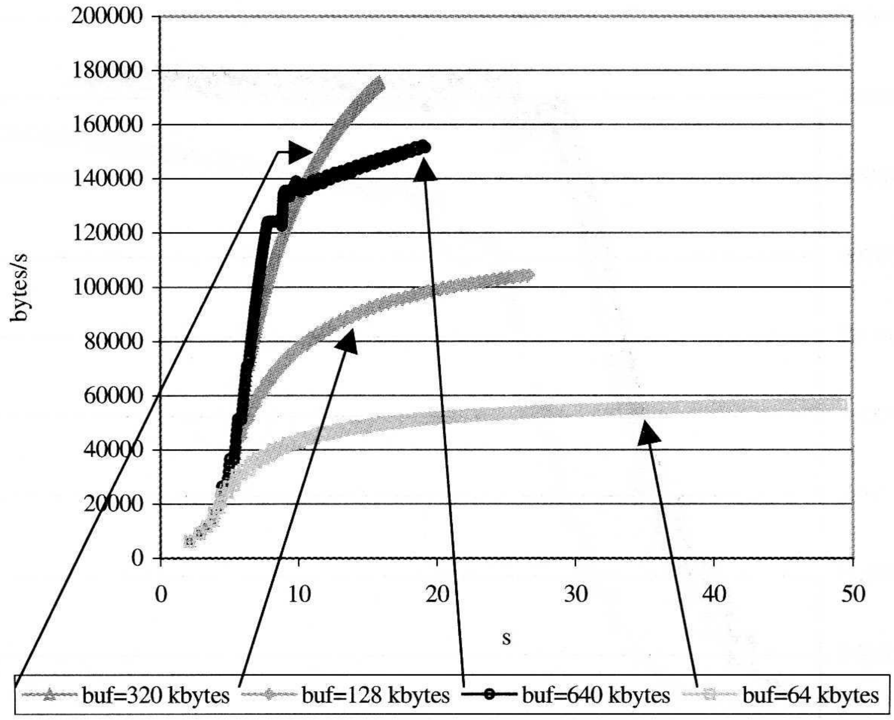

Fig. 3. Throughput (bytes/s) versus time for different values of the buffer length, IW = 1, $H = 2.8$ Mbytes.

kept equal; i.e. the buffer has the same length both for the source and the destination. This parameter is intended as the memory availability in bits and it is identified with the variable 'buf'. The effective transmission window is the minimum between cwnd and the receiver's advertised window (rwnd), which is strictly dependent from the receiver buffer length. A large buffer guarantees that the bottleneck of the system (concerning the packets in flight) is not so severe.

Fig. 3 shows the throughput (bytes/s) versus time for different values of the buffer length, for a transfer of 2.8 Mbytes ($H = 2.8$ Mbytes).

The reference TCP, currently used in cabled networks, is labeled as 64 kbytes because the buffer size is set to 64 kbytes. It is clear that, in this situation, the increase in speed is very slow, the TCP is drastically blocked by the satellite delay, the transmission window cannot augment its length because the buffer dimension represents a bottleneck for the system. The time required to transfer the data is about 50 s. The number of packets in flight is increased,

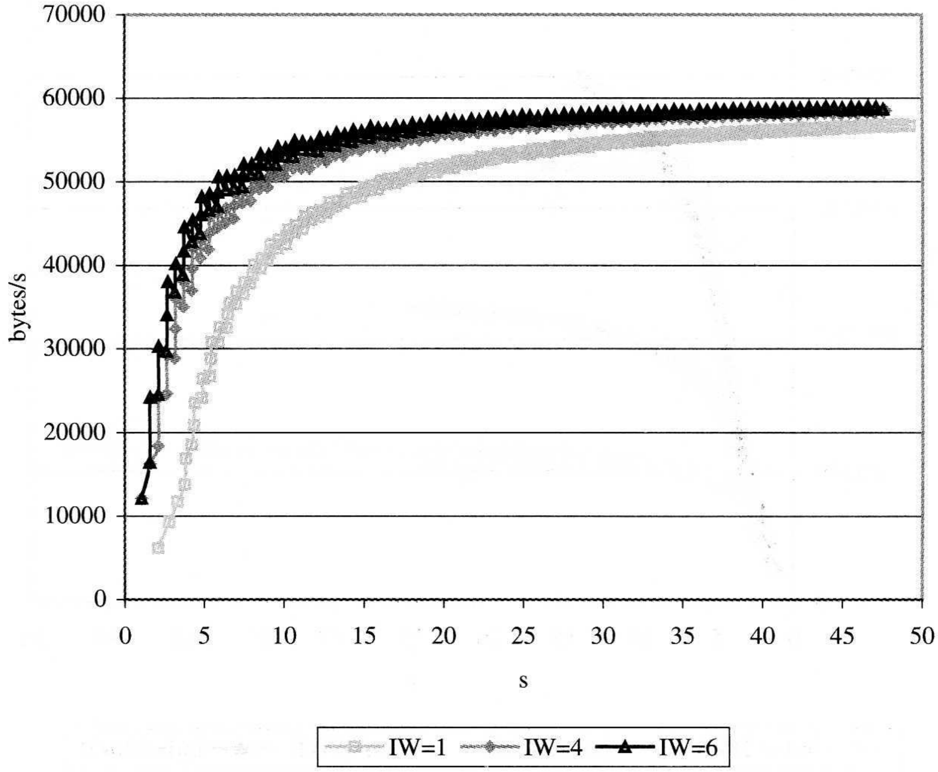

Fig. 4. Throughput (bytes/s) versus time for different values of the initial congestion window (IW), buf = 64 kbytes, $H = 2.8$ Mbytes.

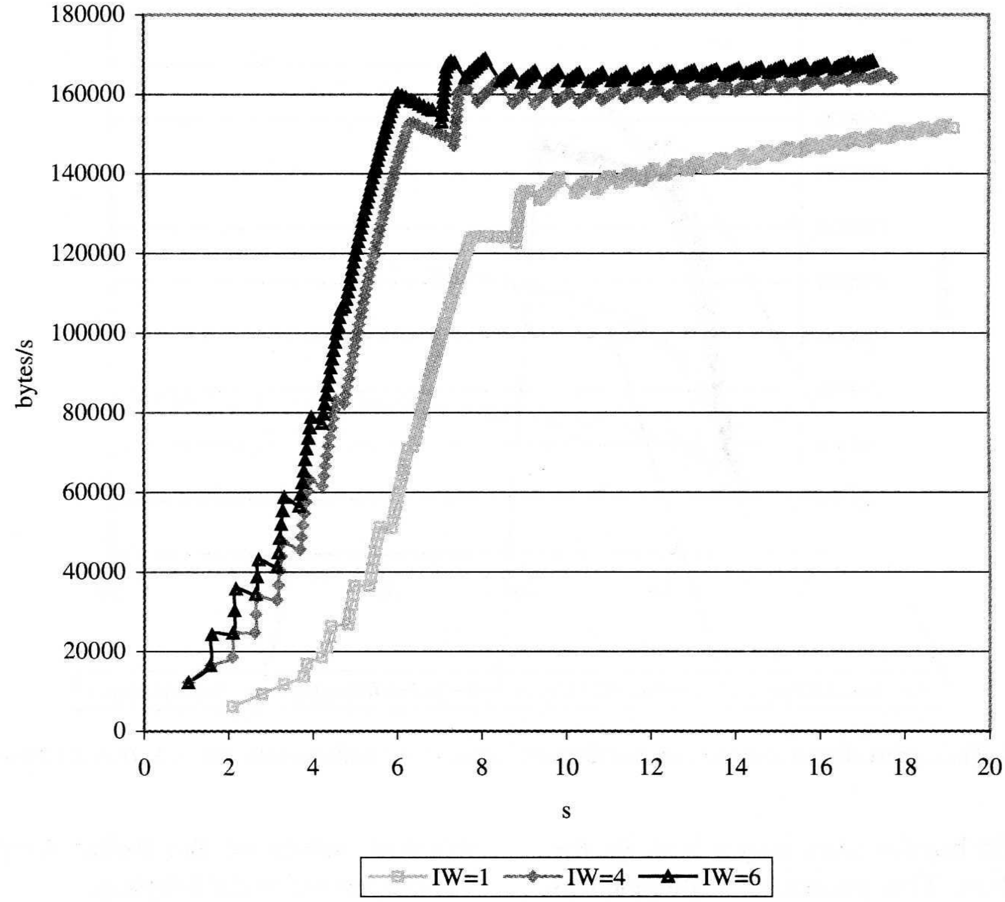

Fig. 5. Throughput (bytes/s) versus time for different values of the initial congestion window (IW), buf = 640 kbytes, $H = 2.8$ Mbytes.

as well as the system performance, if the buffer dimension is augmented. The improvement is outstanding, the transfer time is reduced to about 15 s in the best case. The behavior can be simply described as follows: the congestion window is the bottleneck until it reaches the buffer size, which happens after few seconds. Then the buffer rules the system. This is true if there is no congestion. Due to the presence of just one active connection, the congestion can be generated

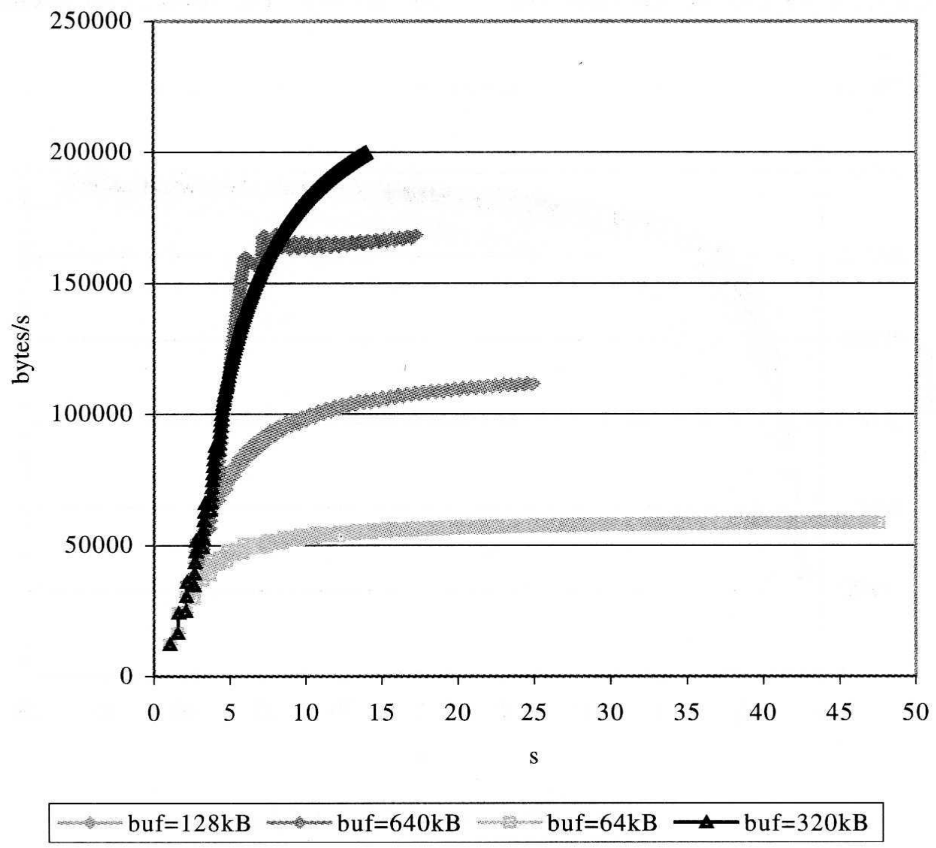

Fig. 6. Throughput (bytes/s) versus time for different values of the buffer length, IW = 6, $H = 2.8$ Mbytes.

Table 4
TCP performance by varying the initial congestion window and the buffer length, $H = 2.8$ Mbytes

IW, buf (kbytes)	Transmission time (s)	Throughput (kbytes/s)	Gain (%)
1, 64	49.21	56665	–
6, 320	13.96	199812	71.63

only by a too aggressive behavior of the TCP; in this case, the router between the host and the modem is saturated. This behavior can be evaluated by observing the line labeled as 640 kbytes. The congestion window increases exponentially for about 7 s and, before the buffer length represents the bottleneck, the system saturates, the fast retransmit strategy is applied and the congestion window is halved. At this point, the increase is linear as the congestion avoidance scheme constraints.

The buffer length is very important for the performance of the system; it rules the congestion window by imposing a bottleneck to its increment. A short buffer drastically limits the performance, but an excessively long buffer makes the system congested.

When the system is congested, the throughput is strongly reduced. The intervention is really drastic. So, it is difficult to approximate an ideal behavior. The idea, as partially developed in the following, could be to act on the increasing strategy during the slow start phase. An important parameter of the slow start algorithm is the initial congestion window. The issue has been treated in the literature. Simulation studies, though not for the specific satellite environment [16], show the positive effect of increased IW for a single connection. Ref. [4] clarifies the strict dependence of the performance on the application environment and suggests that “larger initial windows should not dramatically increase the burstiness of TCP traffic in the Internet today”. IW is set to 1 in the TCP of reference. The performance improvement (i.e. the reduction of time required for the whole transmission and the higher throughput) provided by setting the initial value to 4 and 6 is shown in Fig. 4 for a buffer of 64 kbytes ($H = 2.8$ Mbytes). There is no congestion and TCP is more aggressive, depending on the initial window. The improvement is limited to the throughput in the first part of the connection, being the initial window responsible of this phase. A proper tuning of the two parameters (IW and buf) is very important to avoid the congestion and to improve the overall performance of the system. The effect of congestion, already envisaged in Fig. 3 is clearer in Fig. 5, where the same quantities of Fig. 4 are shown for a buffer value of 640 kbytes. The increased dimension of IW improves the performance, but the aggressive behavior, which derives from the combined action of IW and buf, saturates the network after few seconds. This behavior may be noted also from Fig. 6, where the throughput is shown versus time for different values of buf; IW is set to 6; a transfer of 2.8 Mbytes is performed. The ‘ideal’ value (among the tested ones) of IW and buf for the configuration taken into account may be taken from Fig. 6 (IW = 6, buf = 320 kbytes). Less than 14 s are required to transfer 2.8 Mbytes of the test file, in this case.

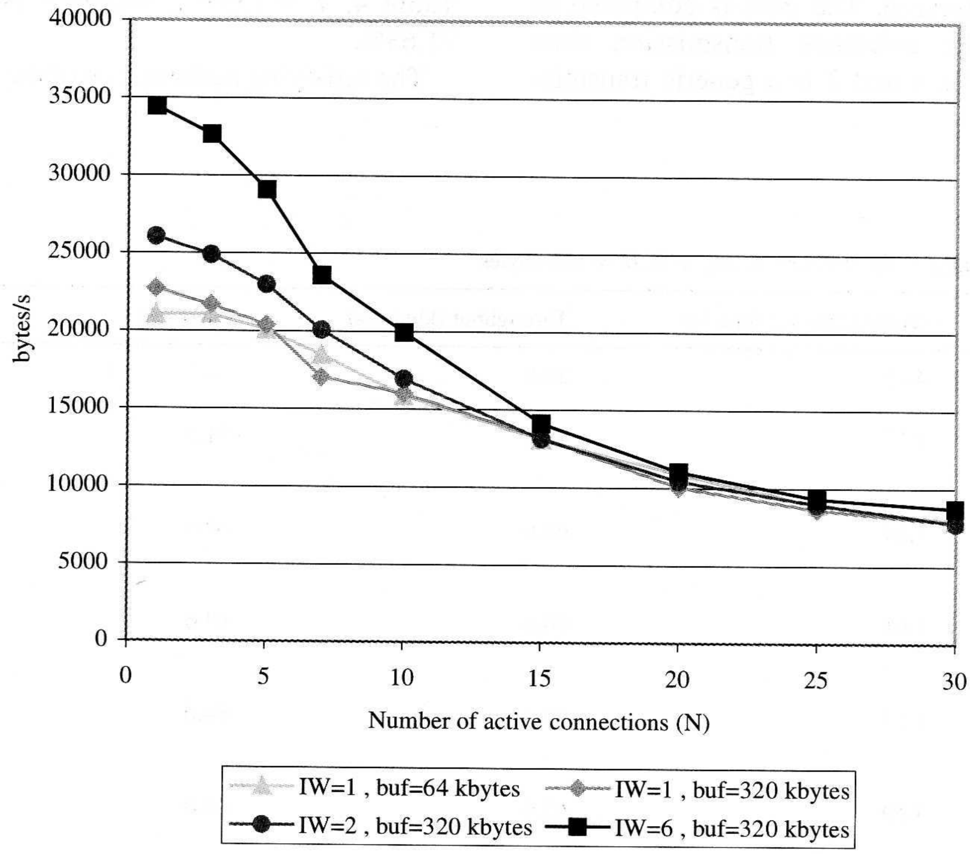

Fig. 7. Throughput (bytes/s) versus the number of active connections (N), $H = 100$ kbytes.

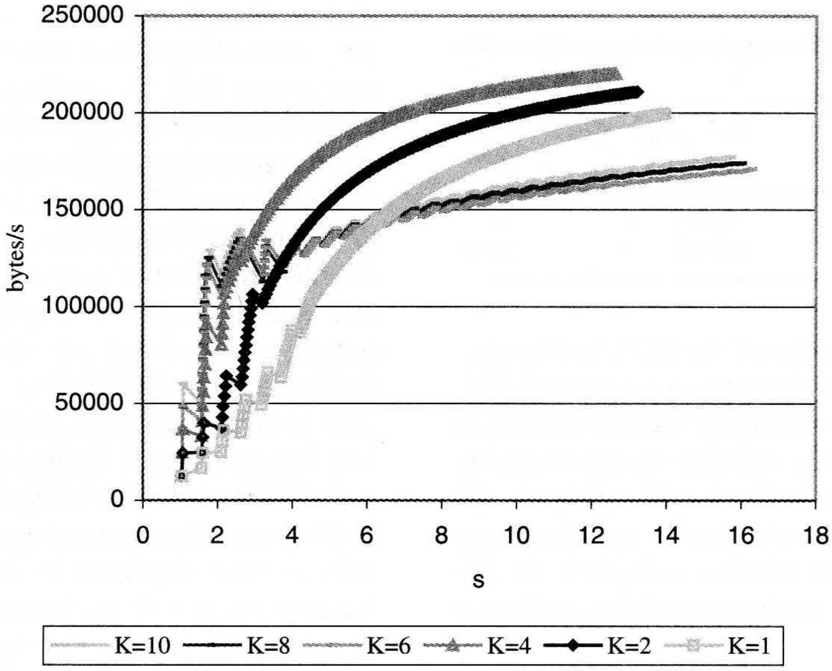

Fig. 8. Throughput (bytes/s) versus time, for different values of K, IW = 6, buf = 320 kbytes, $H = 2.8$ Mbytes.

Table 4 contains the combination of the parameters IW and buf, the time required for the overall transmission and the gain in percentage obtained with respect to the reference configuration (IW = 1, buf = 64 kbytes) with a file transfer of 2.8 Mbytes ($H = 2.8$ Mbytes). It allows summarizing the effect of the parameter tuning performed. Only the best result obtained (IW = 6, buf = 320 kbytes) is reported. It is taken as a starting point for the next steps of the research. The gain is computed as follows: if T_{REF} is the reference transmission time ($T_{\mathrm{REF}} = 49.21$ s), in Table 4 and T is a generic transmission time, the percentage gain is defined as:

$$\%\mathrm{Gain} = \begin{cases} \dfrac{T_{\mathrm{REF}} - T}{T_{\mathrm{REF}}} \cdot 100, & \text{if } T < T_{\mathrm{REF}} \\ 0, & \text{otherwise} \end{cases}.$$

If $T_{\mathrm{REF}} \leq T$, there is no gain. For example, referring to Table 4, $T = 13.96$ s, %Gain = (49.21 − 13.96)/49.21 = 71.63%.

The satisfying performance of the configuration (IW = 6,

Table 5
Overall transfer time and throughput, for different values of k, $H = 100$ kbytes

TCP configuration	Overall transfer time (s)	Throughput (kbytes/s)	Gain (%)
Reference: IW = 1, buf = 64 kbytes, K = 1	4.42	23.6	–
IW = 6, buf = 320 kbytes, K = 2	2.15	48.7	51.3
IW = 6, buf = 320 kbytes, K = 4	1.64	63.6	62.9
IW = 6, buf = 320 kbytes, K = 6	1.61	64.6	63.6
IW = 6, buf = 320 kbytes, K = 8	1.59	65.6	64.0
IW = 6, buf = 320 kbytes, K = 10	1.59	65.6	64.0

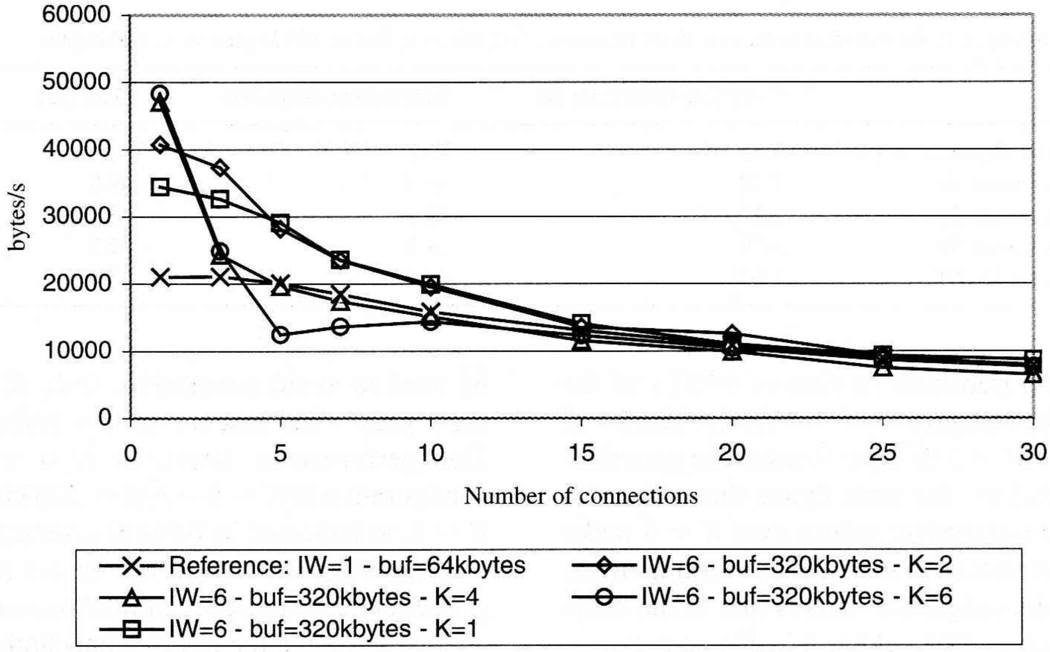

Fig. 9. Throughput (bytes/s) versus time, for different values of K in the multi-connection case.

buf = 320 kbytes) is evident also in the multi-connection case, reported in Fig. 7. It shows the average throughput per connection and compares the behavior of different TCP configurations versus the number N of active connections for a 100 kbytes transfer. Four configurations are taken into account: the reference configuration (IW = 1, buf = 64 kbytes); (IW = 1, buf = 320 kbytes), where only the buffer is varied; and two configurations where both buf and IW are increased: (IW = 2, buf = 320 kbytes) and (IW = 6, buf = 320 kbytes).

The gain is relevant up to 15 active connections; for larger values of N, the traffic load due to the number of connections in progress makes the TCP insensitive to the modifications introduced.

5.2. Slow start increase function

Concerning the modification of the slow start increase function, the functions $F(\cdot)$ listed in Section 2 have been used.

The first graph shown, Fig. 8, contains the throughput by varying the value K in the function $F(\cdot) = F'(\cdot)$, as defined in formula (2) of Section 3. The throughput at the beginning of the connection is high and the overall transmission time

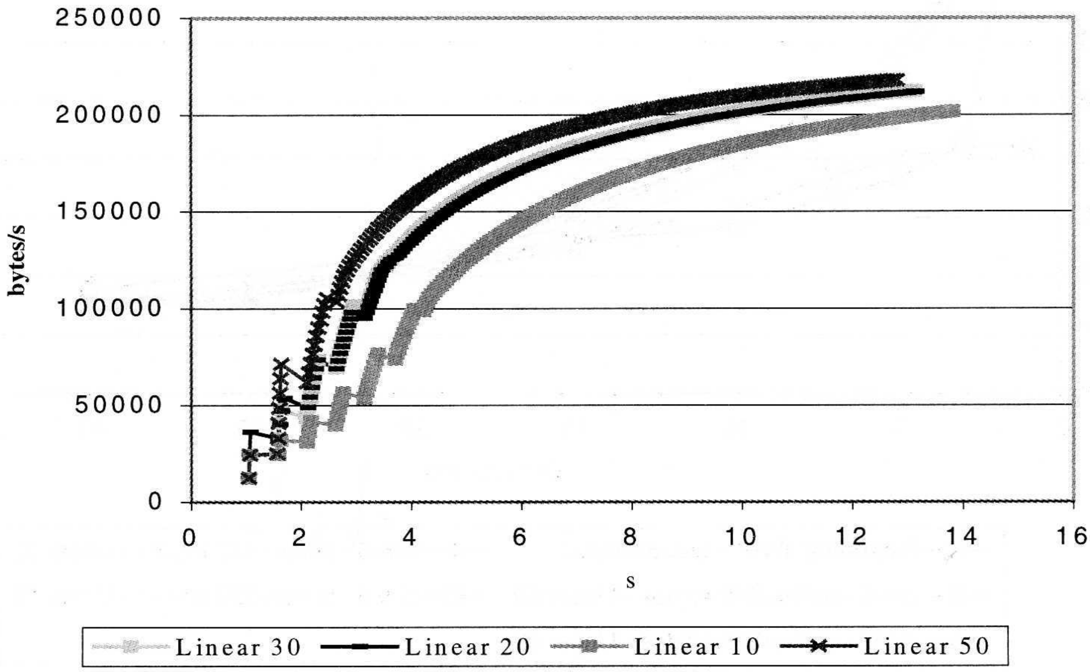

Fig. 10. Throughput (bytes/s) versus time, for different values of thr of function $F''(\cdot)$, IW = 6, buf = 320 kbytes, H = 2.8 Mbytes.

Table 6
Overall transfer time and throughput, for different values of thr of function $F''(\cdot)$, IW = 6, buf = 320 kbytes, H = 100 kbytes

TCP configuration	Overall transfer time (s)	Throughput (kbytes/s)	Gain (%)
Reference: IW = 1, buf = 64 kbytes, K = 1	4.42	23.6	–
IW = 6, buf = 320 kbytes, Linear 10	2.22	46.9	49.8
IW = 6, buf = 320 kbytes, Linear 20	2.12	49.2	48.0
IW = 6, buf = 320 kbytes, Linear 30	2.09	49.8	52.7
IW = 6, buf = 320 kbytes, Linear 50	1.62	64.4	63.3

drastically improves: a transmission time of 49.21 s of the classical case has to be compared with a value of 12.65 s of the (K = 4, IW = 6, buf = 320 kbytes) case. The gain rises up to 74.3%. Nevertheless, the same figure shows the risk for TCP of being too aggressive: values over $K = 4$ make the intermediate router unable to manage the entering traffic and the network results congested. This is due to the characteristics of the function $F(\cdot)$, which is really aggressive, being a constant for the whole duration of the slow start phase. The overall transfer time and the throughput for different values of K if a 100 kbytes file is transferred, are reported in Table 5. The gain is relevant both for time and throughput up to a value 4 of K; after that, the gain is not so evident and, even if no congestion is entered for short files, it is not convenient to set a high value of K because the risk of congestion is high.

Fig. 9 contains the multi-connection case for the configurations shown in Table 5 up to $K = 6$. If a larger buffer value is introduced, a smaller value of K should be used to avoid congestion. Only $K = 2$ gives a permanent gain with respect to the reference configuration. The performance improves also with respect to the configuration (IW = 6 − buf = 320 kbytes), which implies $K = 1$, as indicated in Table 5, at least for small values of N. It is important to note that the impact of the buffer length to get better performance is limited because files of 100 kbytes are transferred in the multi-connection case. The role of IW and K is more relevant.

An attempt to smooth the effect of the variable K is the introduction of function $F(\cdot) = F''(\cdot)$, formula (3) in Section 3. $F''(\cdot)$ has a flat behavior, as in classical TCP, after a quick start. The method is aimed at improving performances in the first phase of the connection (or for very short file transfers) without reaching a congestion status.

The behavior with $F''(\cdot)$ is reported in Fig. 10, for a transfer of 2.8 Mbytes. The initial window and the buffer size are set as follows: IW = 6, buf = 320 kbytes. The value thr of the threshold assumes the values, 10, 20, 30

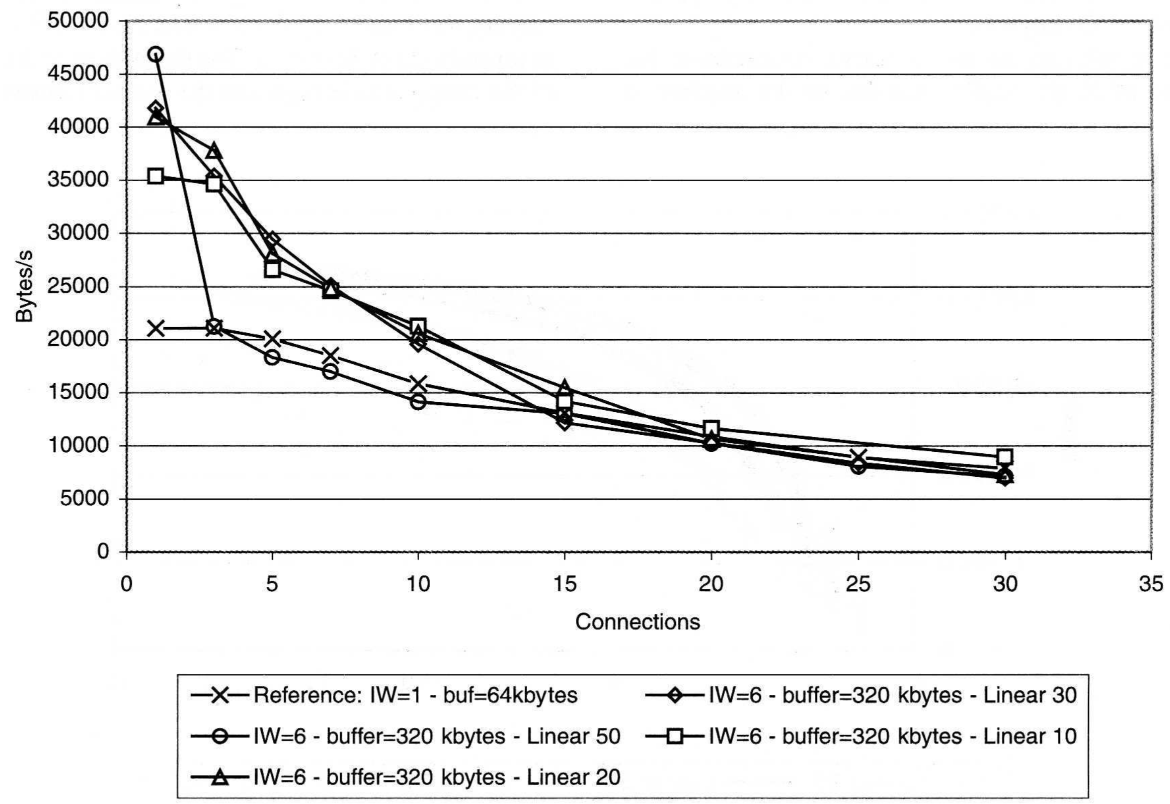

Fig. 11. Throughput (bytes/s) versus time, $F''(\cdot)$, multi-connections case.

Table 7
Overall transfer time and throughput, for a selection of configurations, $H = 2.8$ Mbytes

TCP configuration	Overall transfer time (s)	Throughput (kbytes/s)	Gain (%)
Reference: IW = 1, buf = 64 kbytes	49.21	56.7	–
IW = 6, buf = 320 kbytes	13.95	199.8	71.63
IW = 6, buf = 320 kbytes, K = 2	13.19	200.9	73.28
IW = 6, buf = 320 kbytes, Linear 20	13.15	211.9	73.28

and 50, identified, respectively, as Linear 10, Linear 20, Linear 30 and Linear 50, as stated in Section 3.

The Gain ranges from 72% of Linear 10 to 74% of Linear 50. Table 6 contains the overall transfer time and the throughput for the configurations reported in Fig. 10 along with the Gain value, with respect to the reference configuration for each of them. A transfer of 100 kbytes is performed in this case.

The modification introduced allows choosing an aggressive configuration without entering a congestion status.

The graph containing the performance concerning $F''(\cdot)$ slow start modification in the multi-connection case is reported in Fig. 11: the throughput versus the number of connections for different values of the threshold is shown along with the reference version of TCP. The configurations reported in Table 6 are used. It is important to note the efficiency of the proposed schemes: Linear 20, in particular, and the correct use of the available bandwidth, except for Linear 50, which is too aggressive.

The next two graphs allow summarizing the TCP configurations that provide the best results in the single connection case and that maintain the gain also in the multi-connection case. A configuration that guarantees gain in both cases is desirable but the advantage given by the more aggressive configurations (i.e. $K = 4$, Linear 50) in the single connection case should not be neglected. Also the single connection case has a specific application, as evidenced in Section 4, and the choice of a particular configuration for a peculiar environment or application may be necessary.

Table 7 contains the overall transfer time, the throughput and the percentage gain of the selected configurations for a 2.8 Mbytes transfer in the single connection case. The multiple connection case for the same configurations is shown in Fig. 12.

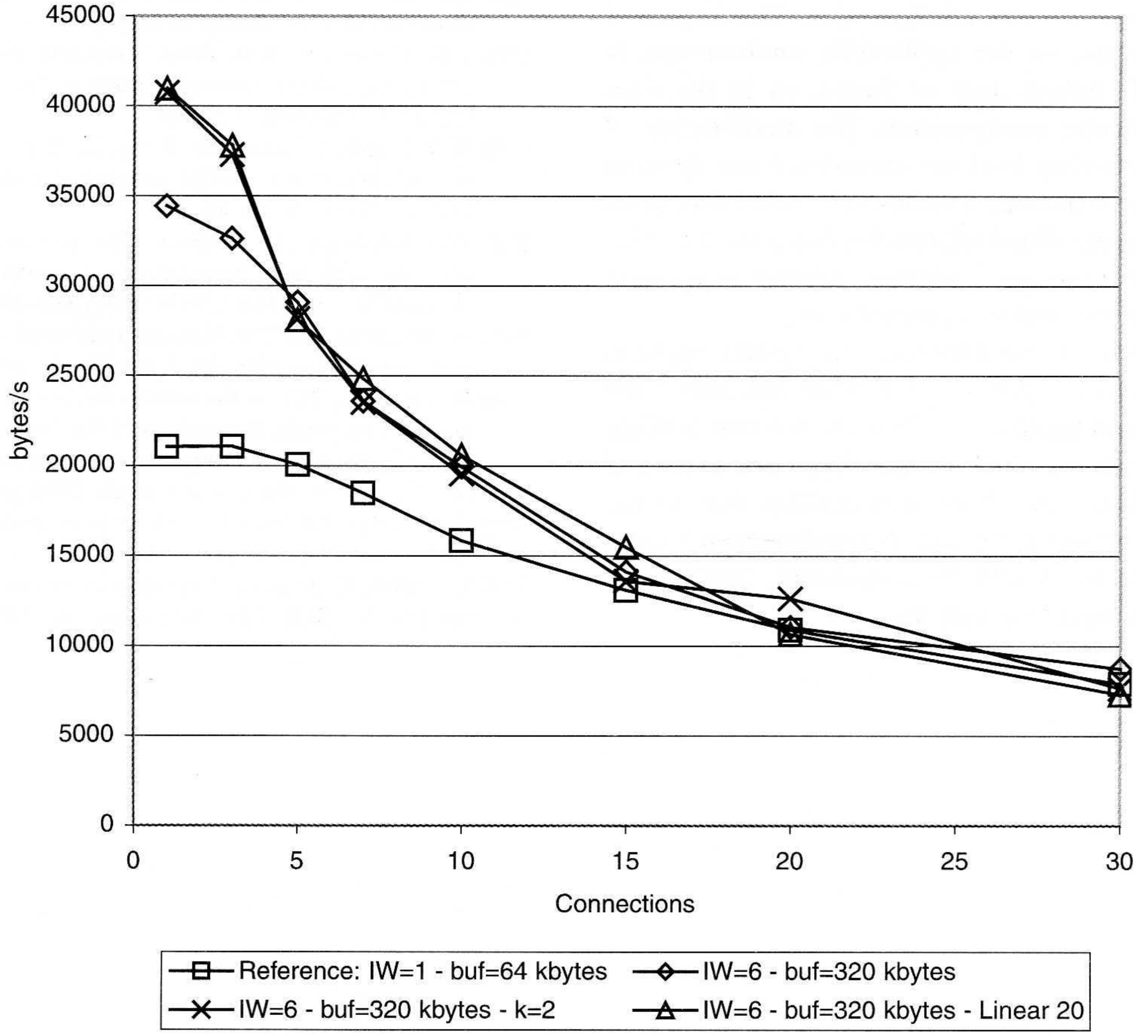

Fig. 12. Throughput (bytes/s) versus time, selection of configurations, multi-connections case.

It is important to evidence that the two configurations with modified slow start increase ($K = 2$ and Linear 20) gives an advantage with respect to (IW = 6, buf = 320 kbytes) for a small number of connections in the network. The modified TCP, as envisaged also in Fig. 7, is efficient up to 10 simultaneous connections. It is equivalent, but not worst, to the reference TCP after that value.

6. Conclusions

The behavior of the TCP in a satellite link has been investigated. Parameters as the buffer dimension and the initial congestion window have been tuned. The dynamic of the slow start algorithm has been modified. The analysis itself has allowed suggesting some solutions. TCP results highly inefficient if used with no modification, the satellite imposes a delay and drastically reduces the performances of the protocol. The improvement due to an augmented dimension of the transmitter/receiver buffer and of the initial congestion window is outstanding, but a proper tuning of the two quantities to avoid congestion is strongly necessary. A large buffer, along with an extended initial window, makes the TCP more aggressive: on one hand, this improves the network throughput but, on the other hand, if the tuning is not precise, the risk of saturation is high. It is not simple to find a solution suited for any configuration. The impact of the link capacity and of the application environment is strong. An 'ideal' tuning may be found, as in the case treated, for a particular configuration. The modification of the slow start mechanism has been considered too. Even in this case, the window increase rate has to be ruled with great attention. If the non-modified scheme implies a low transfer, it is also true that a too quick increase creates congestion and, as a consequence, reduces performance.

The system has been tested by using an ftp-like application, i.e. a file transfer application located just above the TCP, which has been taken as a reference, in a real satellite environment. An almost ideal channel has been measured and the lost packets have been substantially due to the saturation of intermediate routers. Nevertheless, a proper configuration scheme should be considered, particularly when there are multiple connections.

Acknowledgements

Contract/grant sponsor: Italian Space Agency (ASI); Contract/grant number: ASI-ARS-99-205, "Integration of Multimedia Services on Heterogeneous Satellite Networks".

References

[1] D. Adami, A. Conti, D. Dardari, M. Marchese, L.S. Ronga, CNIT–ASI project integration of multimedia services on heterogeneous satellite networks: an overview, Proceedings of the Fifth Ka Band Utilization Conference, Taormina, Italy, 1999, pp. 219–226.
[2] M. Allman, D. Glover, L. Sanchez, Enhancing TCP over satellite channels using standard mechanism, IETF, RFC 2488, January, 1999.
[3] M. Allman, V. Paxson, W.S. Stevens, TCP Congestion Control, IETF, RFC 2581, April 1999.
[4] 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.
[5] ARTES 3, The ESA Multimedia Initiative, Home Page, http://telecom.estec.esa.nl/artes/artes3/ (last access 20th October 2000).
[6] C. Barakat, E. Altman, W. Dabbous, On TCP performance in a heterogeneous network: a survey, IEEE Commun. Mag. 38 (1) (2000) 40–46.
[7] S. Floyd, T. Henderson, The NewReno modification to TCP's fast recovery algorithm, IETF, RFC 2582, April 1999.
[8] R.T. Gedney, R. Schertler, F. Gargione, The Advanced Communications Technology Satellite, SciTech Publishing, Mendham, NJ, 2000.
[9] N. Ghani, S. Dixit, TCP/IP Enhancement for satellite networks, IEEE Commun. Mag. 37 (7) (1999) 64–72.
[10] T.R. Henderson, R.H. Katz, Transport protocols for internet-compatible satellite networks, IEEE J. Selected Areas Commun. (JSAC) 17 (2) (1999) 326–344.
[11] W.D. Ivancic, D. Brooks, B. Frantz, D. Hoder, D. Shell, D. Beering, NASA's broadband satellite networking research, IEEE Commun. Mag. 37 (7) (1999) 40–47.
[12] T.V. Lakshman, U. Madhow, The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Trans. Netw. 5 (3) (1997) 336–350.
[13] M. Marchese, TCP modifications over satellite channels: study and performance evaluation, Int. J. Satellite Commun. (2001) (in press).
[14] M. Marchese, TCP in the satellite environment: analysis and some operative proposals, Proceedings of the 2000 Symposium on Performance Evaluation of Computer and Telecommunications Systems (SPECTS 2000), Vancouver, Canada, 2000, pp. 185–191.
[15] C. Partridge, T.J. Shepard, TCP/IP performance over satellite links, IEEE Netw. 11 (5) (1997) 44–49.
[16] K. Poduri, K. Nichols, Simulation studies of increased initial TCP window size, IETF, RFC 2415, September 1998.