A SIMULATION STUDY OF TCP PERFORMANCE OVER SATELLITE CHANNELS

G. Neglia, V. Mancuso, F. Saitta, I. Tinnirello

Department of Electrical Engineering, University of Palermo
Viale delle Scienze 9, 90128 Palermo, Italy

ABSTRACT

This document is mainly about the influence of protocols on the performance of Satellite Communication. It is widely recognized that satellite communications are affected by some peculiar problems, which penalize heavily the performance and efficiency of the Transmission Control Protocol (TCP). In fact, wireless satellite channels are usually characterized by link-asymmetry and higher Round Trip Time and Bit Error Rate in comparison to wired links. It means that TCP, which was developed for wired channels, exhibits often poor performance in a satellite scenario (unfair bandwidth repartition, low throughput and long file-transfer delay). We have studied and compared many TCP variants recently proposed. In particular we have compared TCP-Reno standard implementation with TCP-SACK, TCP-Westwood, TCP-Vegas and TCP-Tibet. We tested the different protocols on a simulated satellite scenario, with the support of NS2, a well-know network simulator platform, annotating the advantages and drawbacks of various protocols in order to improve the transmission of IP data packets over satellite channels.

Keywords: Satellite link, Performance analysis, TCP behaviors.

1. INTRODUCTION

The use of satellites for Internet traffic is a very attractive proposition since the wired network is highly congested. On the average, any IP flow needs to travel through 17 to 20 nodes, and hence may go across multiple bottlenecks. On the other hand, with satellites, just one hop is sufficient to connect two very distant sites.

Unfortunately, the use of satellite is limited by the poor performance obtained by usual TCP/IP suite protocols. In fact the long propagation delay of satellite channel has the same effect of traveling through various bottlenecks affecting the congestion control scheme of many connection oriented protocols, which erroneously underestimate the bandwidth available on the link and assume to be in a congestion state. Moreover, in the future communication system, the satellite component is supposed to be integrated with the terrestrial component [1] and the interaction of satellite and terrestrial connections brings new performance problems which are actually partially unexplored and not completely solved. Satellite communication needs to be completely integrated in the actual communication system without technical handicaps, if it wants to assume a particularly important role, and not only complementary, in the future communication world, especially if aiming at real global coverage and ensuring access to maritime, aeronautical and remote users.

The main goal of this paper is to evaluate the performance of typical mobile Internet applications, using different TCP schemes. While high efficiency on fast satellite links may eventually require the introduction of a new protocol, many studies [2] indicate that full link utilization may be achieved by using the TCP performance enhancements that have already been approved or that are currently in the standards process. Of particular interest in the satellite environment are performance enhancements for scaled windows and timestamps, fast retransmit and recovery, and selective acknowledgement. This paper describes the protocols which adopt these performance enhancements and show the effects of their implementation on increasing TCP's performance over satellite links.

We analyze some representative satellite scenarios with a classical GEO configuration satellite repeater. The paper is organized as follows. Section 2 reviews the role of satellites. In Section 3, we highlight the characteristics of satellites likely to have an effect on TCP and other Internet applications. Section 4 describes in more details TCP protocol variations used in our experiments. In Section 5, we describe the experiments and present the experimental results in Section 6. Section 7 gives the conclusions.

2. THE ROLE OF SATELLITES

In the future, satellites will play an important role in providing wireless access to mobile users, for enhancing coverage of existing or developing land mobile systems and to ensure long-range mobility and large bandwidth. The presence of satellite links provides extra capacity and an alternate and less congested route to existing wired and wireless systems, thus offering unique opportunities for improving efficiency and reliability. The broadcast nature of the satellite channel also makes satellites suitable for broadcasting/multicasting. Besides the satellite is the only way for extending the range of communication out of our planet connecting space stations and space shuttles in a future interplanetary communication network. The growing interest in exploring new worlds needs an affordable and performing system of communication for granting the success of space missions, in order to guarantee a complete control of mission evolution.

The satellite may also be integrated with short-range wireless networks (Bluetooth) in order to provide Internet services to mobile vehicles. In this scenario, the satellite will connect one vehicular terminal to the Internet while the Bluetooth system will be able to interconnect equipment inside the vehicle (car, bus, train, plane or ship).

The main satellite communication systems, providing large bandwidth directly to users, are based on a geosynchronous orbital configuration. A single GEO satellite is able to cover a very wide area with multiple narrow spot beams using multibeam antennas. In this way, very reliable links can be provided even with small terminals. Such GEO systems provide (or are designed to provide) high availability high data rate links (up to 2 Mbit/s and more) and huge total capacity (of the order of 10 Gbit/s). Many of them are expected to provide regional limited coverage in a few years and global coverage (excluding poles) finally.

Previous works [8-12] has addressed the performance of TCP/IP over geostationary satellites. This has mainly focused on evaluating performance of different TCP schemes (Tahoe, Reno, New Reno, SACK), where the connection between two fixed stations goes across a GEO satellite. In this paper, we also address other TCP schemes (Westwood [12] and Tibet[7]) and evaluate TCP performance when terrestrial and satellite TCP connection share the same bottleneck in order to establish which protocol has the better behaviour when used for both long and short Round Trip Time (RTT) connection; besides we exposed both type of connections to the aggressive behaviour of a UDP flow to evaluate the ability to conserve their throughput. Such an environment creates a very realistic scenario for the evaluation of performance in a satellite environment.

3. SATELLITE FEATURES AFFECTING PERFORMANCE OF IP APPLICATIONS

First, when considering the performance of real time (e.g. voice and video) applications the large propagation delays represent a critical issue. The problem is particularly acute in TCP applications over links with large bandwidth-delay products, where a large TCP window is required for efficient use of the satellite link. A lossy satellite link can cause frequent “slow start” events, with significant impact on throughput. The key satellite network features that need to be considered in order to evaluate the impact of satellite on internet application performance are: propagation delay, Delay-Bandwidth Product (DBP), signal-to-noise ratio (SNR), satellite diversity, routing strategy.

Satellite systems are changing their role from the “bent-pipe” (transparent) channel paradigm to the “on-board routing” (regenerative) paradigm associated with packet transmission and switching. In this process each satellite is an element of a network, and new design problems are involved both at the network and the transport layers.

While considering the performance of standard Internet protocols over paths including satellite links, the large propagation delays have to be taken into account. The problem is particularly relevant to TCP due to its delay-sensitive nature. For voice and other real time applications too, the delay may be critical.

Some of the main aspects that need to be considered in order to evaluate the impact of system architecture on the performance of network protocols are:

Propagation delay:

Due to the large distance that packets need to travel in satellite networks, they can experience a significant transmission delay (depending on the type of constellation and the routing strategy adopted).

The transmission delay in a GEO architecture depends on the user-gateway distance (or user-user if direct connection is allowed) when a single satellite configuration with no ISLs (inter-satellite links) is considered, or on the connection strategy if ISLs are used. This delay can be considered constant in the case of fixed users while it is variable in the case of mobile users.

However due to queue phenomenon in terrestrial router even fixed users can experience a large variability in propagation delay. This phenomenon is even more critical if ISLs are implemented, since the routing strategy may also play a very important role.

The delay variability is particularly relevant to TCP, since it causes the delivery packet time to be variable and may affect the estimate of Round Trip Time (RTT) by TCP. Whenever a change in RTT occurs, it takes some time for TCP to adapt to the change. If the RTT keeps changing, TCP may not be able to update its estimate of RTT quickly enough. This may cause premature timeouts/retransmissions, reducing the overall bandwidth efficiency.

Delay-Bandwidth Product (DBP):

The performance of TCP over satellite links is also influenced by the Delay-Bandwidth-Product (DBP). This has been evaluated through simulations and experiments [9, 10, and 11]. This performance can be enhanced by adopting techniques such as selective acknowledge and window dimensioning [9, 11]. Also, techniques such as TCP spoofing, splitting and cascading may be used [11].

Satellite diversity:

Satellite diversity is a very efficient technique for improving link availability or SNR by utilizing more than one user-satellite link to establish communication. In a connectionless system, the satellite diversity feature has the advantage of increasing the probability of packet delivery to/from the satellite network from/to Earth. However, this increases the number of busy channels, which causes a reduction in capacity.

Signal-to-noise ratio (SNR):

In GEO constellations, the SNR (or equivalently Bit Error Rate, BER) is characterized by a great variability due to free space losses and troposphere propagation (including rain, over 10 GHz) for fixed communications and also due to shadowing in case of mobile communications. Both shadowing and deep rain fading can cause packet loss.

Poor SNR is an extensively studied issue for GEO satellites. Since TCP has been designed mainly for wired links, it assumes that the BER is very low (of the order of 10^{-10}, or less) so that when a packet is lost, TCP attributes it to network congestion. This causes TCP to employ its congestion control algorithms, which reduces the overall throughput. This is detrimental in satellite links, and causes an unnecessary reduction in throughput.

4. TCP PROTOCOLS DETAILS

The Transmission Control Protocol (TCP) was originally designed for wired link so it is based on some assumption related to specific characteristic of wired link. Moreover TCP is based on the assumption that the network does not provide any explicit feedback to the sources. Therefore each source must form its own estimates of the network properties, such as round-trip time (RTT) or usable bandwidth in order to perform efficient end-to-end congestion control.

In order to improve the performance of TCP connections for preventing congestion events and achieve a fair share of bandwidth among different connections, the original TCP protocol was extended with some additive function, the mains of which were implemented in TCP Reno. In the following we describe these functions:

Slow Start and Congestion Avoidance

The slow start and congestion avoidance mechanisms were introduced in 1988 in [13] and added as a requirement for TCP implementations in RFC 1122[14].

Slow start is used to gradually increase the rate at which the sender injects data into the network. Slow start begins by sending one segment and waiting for an acknowledgement. For each acknowledgement the sender receives, it injects two segments into the network, thus leading to an exponential increase in the amount of data being sent. Slow start ends when the receiver’s advertised window is reached or when loss is detected. Because the amount of time required for slow start to achieve full bandwidth is a function of round trip time, satellite links are particularly sensitive to the limited throughput available during slow start.

Congestion avoidance is used to probe the network for available bandwidth by sending one additional segment for each round trip time (up to the receivers advertised window). In the original slow start/congestion avoidance scheme, when the sending TCP detects segment loss (by default assuming network congestion), it drops back into slow start until the packet sending rate is one half the rate at which the loss was detected and then begins the congestion avoidance phase.

Fast Retransmit and Fast Recovery

TCP Reno also incorporates Fast Retransmit and Recovery, RFC 2001[15]. Although these mechanisms have been found in many Unix variants of TCP for several years, they were not documented as a Standards Track RFC until 1997. Fast retransmit reduces the time it takes a TCP sender to detect a single dropped segment. Rather than waiting for the retransmit timeout (RTO), the TCP sender can retransmit a segment if it receives three duplicate ACKs for the segment sent immediately before the lost segment.

Fast recovery works in conjunction with fast retransmit. As mentioned above, when a sender retransmits a segment, it normally recovers by moving first into a slow start phase followed by a congestion avoidance phase. If the sending TCP detects the segment loss using fast retransmit, however, fast recovery is used instead. Fast recovery halves the segment sending rate and begins congestion avoidance immediately, without falling back to slow start.

Large Windows

The original TCP standard limits the TCP receive window to 65535 bytes. TCP’s receiving window size is particularly important in a satellite environment because the maximum throughput of a TCP connection is bounded by the round trip time [16], as seen in the formula:

$$troughput_{\max} = \frac{receive\_buffer\_size}{round\_trip\_time}$$

Without large windows, then, a TCP connection over a typical geosynchronous satellite is limited to throughput:

$$troughput_{\max} = \frac{64Kbytes}{580ms} \approx 900Kbps$$

Note that this upper bound on TCP throughput is independent of the bandwidth of the channel. A TCP connection running over a full T1 channel (1.536.000 bits/second) could still only achieve a maximum throughput of approximately 900 Kbits/second with a 64Kbytes receive window. As specified in RFC 1323[13], large windows (window scaling) can allow TCP to fully utilize higher bandwidth links over long-delay channels such as those found in satellite links.

In addition to these functions, in order to improve the original TCP, many other extension of the TCP protocol were suggested and implemented in different TCP protocols. So in order to evaluate the performance of TCP over satellite links we compared, in the same simulation scenario, the original TPC together with TCP-Reno and other TCP protocols, whose main features are described in the following:

TCP SACK

This protocol introduces the ability for TCP to use Selective Acknowledgments. This aspect is considered very important in satellite scenario, because the cumulative positive acknowledgments employed by TCP are not particularly well suited to the long-delay satellite environment due to the time it takes to obtain information about segment loss. Using a selective acknowledgement (SACK) mechanism, as defined in RFC 2018[4], the SACKs generated at the receiver explicitly inform the sender about which segments have arrived and which may have been lost, giving the sender more information about which segments might need to be retransmitted and avoiding to retransmit segments which travel trough long delay link, such as satellite connections.

TCP Newreno

This protocol implements some modifications in Fast Recovery algorithm, in order to improve the response to partial acknowledgments, which were first proposed by Janey Hoe [6]. In the original implementation of the TCP Fast Recovery algorithm the TCP data sender only retransmits a packet after a retransmit timeout has occurred, or after three duplicate acknowledgements have arrived, triggering the Fast Retransmit algorithm. A single retransmit timeout might result in the retransmission of several data packets, but each invocation of the Reno Fast Retransmit algorithm leads to the retransmission of only a single data packet. TCP NewReno defines a "Fast Recovery procedure" that begins when three duplicate ACKs are received and ends when either a retransmission timeout occurs or an ACK arrives that acknowledges all of the data up to and including the data that was outstanding when the Fast Recovery procedure began. In this way it can recover a situation when multiple packets have been dropped from a single window of data in a faster way than TCP-Reno.

TCP Vegas

In order to increase throughput and decrease losses TCP Vegas introduces three techniques [18]. Firstly, a more timely retransmission of dropped segments is used. This is obtained by using a more accurate RTT estimate, i.e. by reading the system clock each time a segment is sent and again when an ACK arrives. When a non-duplicate ACK is received after a retransmission, Vegas again checks to see if the time interval since the segment was sent is larger than the timeout value. If it is, then Vegas anticipates the retransmission of the segment. Furthermore, Vegas only decreases the congestion window if the retransmitted segment was previously sent after the last decrease. In fact, a loss that happened before the last window decrease does not imply that the network is congested for the current congestion window size.

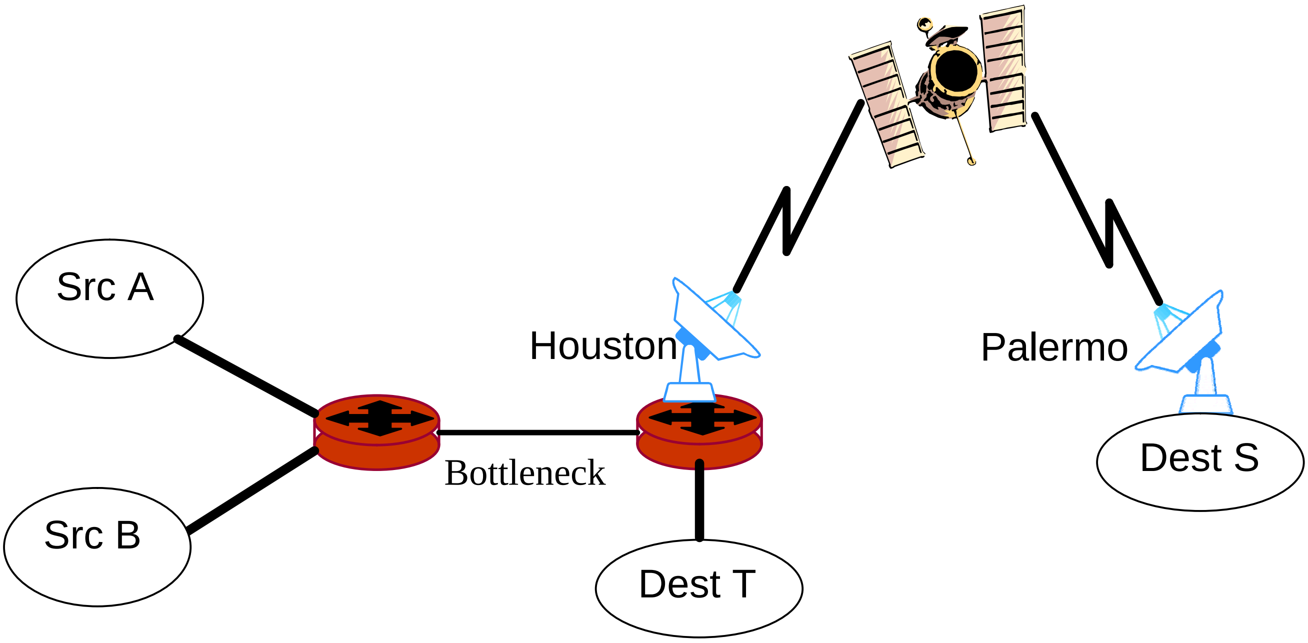

Fig. 1 *Description of simulated scenario, the bottleneck link connects the two routers with a 2 Mb/s link*

Secondly, congestion is anticipated by TCP, and transmission rate is tuned accordingly. Using Vegas, in congestion avoidance state, the window size is increased every RTT by one segment, and the throughput achieved is compared to the one obtained when the window was one segment smaller. If the difference is less than one-half the throughput achieved when only one segment was in transit they decrease the window by one segment.

Thirdly, TCP's slow-start mechanism is modified so that packet loss is avoided while trying to find the available bandwidth. This is obtained by allowing exponential growth only every other RTT, while in the middle, the congestion window doesn't change.

TCP Westwood

This protocol [12] has the advantage that can be implemented only at the sources of TCP connections so at difference of SACK not need a worldwide diffusion for working correctly. It performs an estimate of the available bandwidth by measuring the returning rate of acknowledgements, and uses this estimate for setting the parameter of the Fast recovery mechanism in order to avoid the blind halving of the sending rate as in TCP Reno after packet losses. Resetting the window to match available bandwidth makes TCP Westwood more robust to sporadic losses due to wireless channel problems. These often cause conventional TCP to overreact, leading to unnecessary window reduction. This explicit bandwidth estimation scheme has a deep impact over the performance of TCP, especially in presence of random, sporadic losses or with paths with high bandwidth/delay product typical of satellite links.

TCP Tibet

This protocol implements some modifications in TCP Westwood mechanism for estimating the available bandwidth. In particular, it introduces a double filtering technique to obtain correct estimates of the bandwidth used by TCP sources without error introduced by sporadic losses. The estimate is realized by measuring and low-pass filtering the length of acked packets and the intervals between ACKs' arrivals. In addition it is also possible, in order to estimate the used bandwidth, to apply a low-pass filter directly to the packets' length and the intervals between sending times. Thanks to the Double filtering technique TCP Tibet has a bandwidth estimation mechanism which is not biased by the phenomenon of ACK compression, which usually occurs when packets travel trough long-delay links, and obtains bandwidth estimates which oscillate around the fair-share value when all TCP sources experience almost the same path conditions.

5. SIMULATION SCENARIO

We used Network Simulator NS-2[19] to create a significant scenario in which it was possible to test the throughput performance of the different protocols previously described. In order to differentiate our work from the other similar comparisons of different TCP protocols in particular scenarios [2, 3, 8-12], we choose to focus our work on the effects of interactions between terrestrial short-delay connections and satellite long-delay connections when contending a single bottleneck link. We used the network topology drawn in fig. 1.

The satellite link is used for transferring data between two terminals sited at Palermo and Houston, and it is obtained by means of a GEO satellite placed on the Atlantic ocean at longitude 34.5° West. This link has two 20 Mb/s channels for download and upload from satellite.

Each source node, “Src A” and “Src B”, establishes 15 FTP connections toward the destination “Dest S” for satellite connections and “Dest T” for terrestrial connections. The bottleneck link is between the routers and is characterized by a 2 Mb/s link capacity and 10 ms link propagation delay; all the other terrestrial links have 20 Mb/s link capacity and 10 ms link propagation delay.

The simulations had been running for 6000 simulated seconds and were characterized by three events which affect the way in which the bottleneck capacity is subdivided between connections. In more details, first at 0 seconds we started 15 FTP connections toward the satellite destination “Dest S”, then after 2000 seconds we started 15 FTP connections toward the terrestrial destination “Dest T”, and finally after 4000 seconds we started an UDP connection toward the terrestrial destination. In this way we were able to valuate the performance of each TCP protocol in the following wide rage of situations: first when there are only satellite connection on the bottleneck, then where there are both terrestrial and satellite connection on the same bottleneck, and finally when there is an “aggressive” UDP flow which contend bandwidth both to terrestrial and satellite connections.

Moreover, we conducted some simulations where different TCP protocols were used for terrestrial and satellite connection in order to estimate the behaviour of the system when the new satellite oriented protocols are exposed to the influence of old not wireless oriented TCP protocols largely diffused in the actual IP systems.

6. EXPERIMENTAL RESULTS

In this section we present the results of the simulation based on the previous scenario. Figures from 2 to 8 have been obtained making the assumption that all the sources use the same version of the TCP protocol, while in the successive figures 9-11, we make the confront between the best protocol individuated for satellite connection and the most diffused TCP protocol in the actual distribution of IP network devices.

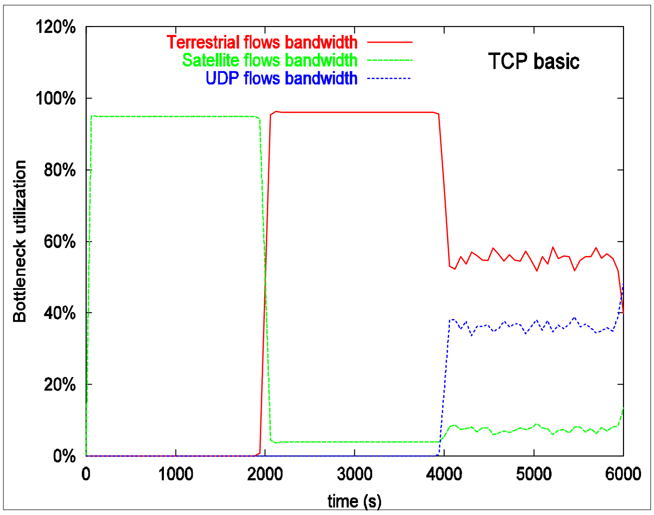

Fig. 2 *Simulation with original TCP protocol for both terrestrial and satellite connections*

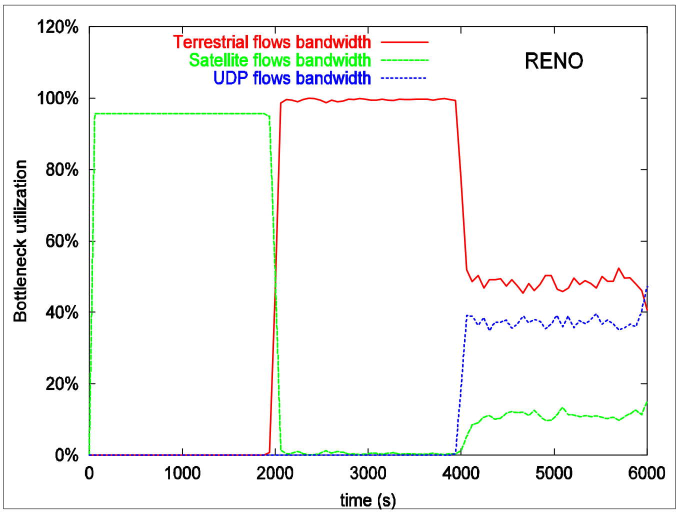

Fig. 3 *Simulation results with TCP Reno protocol for both terrestrial and satellite connections*

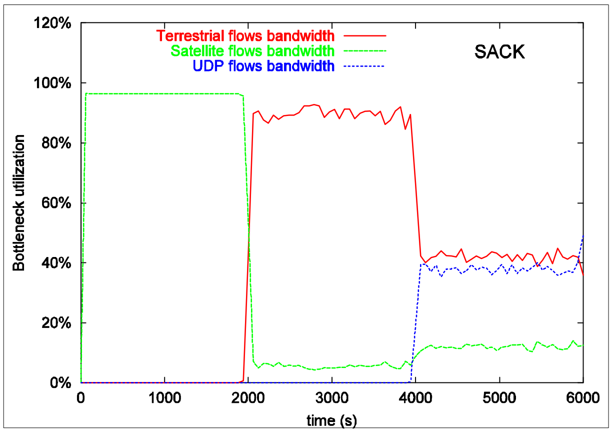

Fig. 4 *Simulation results with TCP Sack for both terrestrial and satellite connections*

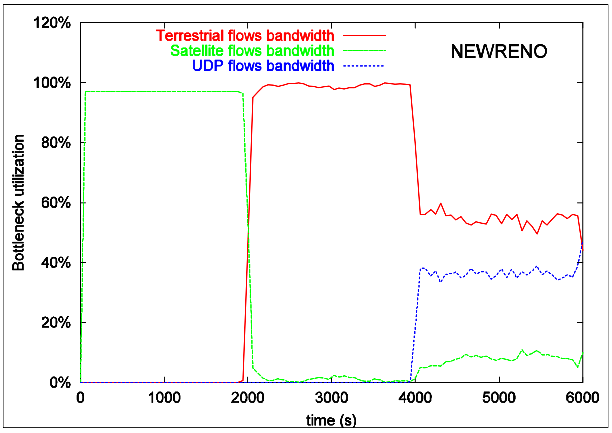

Fig. 5 *Simulation results with TCP Newreno for both terrestrial and satellite connections*

By observing the results obtained during the first part of our simulation, we can evidence that, during the period in which there are only satellite connections, all TCP protocols obtain very good link utilization very close to the 100% of bottleneck capacity, reaching the max limit in Vegas, Westwood and Tibet cases. This is quite important for Westwood and Tibet, which were studied for being implemented in satellite connections, because it is a proof that TCP connections are able to utilize efficiently the network when the right protocols are used.

However, during the second period of simulation (between 2000 and 4000 seconds), the satellite connections suffer the presence of terrestrial connections, which have a short RTT. In this case only the protocols: SACK, Westwood and Tibet are able to service some part of bottleneck capacity to satellite connection; while the other protocols cause a starvation of satellite connections. In this scenario a particular award goes to TCP Tibet, because it is able to adapt the estimated bandwidth to the previous value after the introduction of terrestrial link after a small period of time maintaining the link utilization of the satellite connections stuck to the max. A good performance is also obtained by TCP Westwood, which tries to distribute in a fairly manner the bandwidth between terrestrial and satellite connections, even though satellite connections get a more small part.

At the end, during the third period of simulation (between 4000 and 6000 seconds), the introduction of an aggressive UDP flow bring in the most cases some advantages to the satellite connections. In fact, in all the cases in which the satellite connections were defeated by the terrestrial connections the presence of the UDP stream breaks the channel utilization of TCP terrestrial links and helps the satellite connections to gain some part of bottleneck capacity. Even in this case the best performance for satellite connections are obtained by TCP Westwood and Tibet protocols, which arrive to

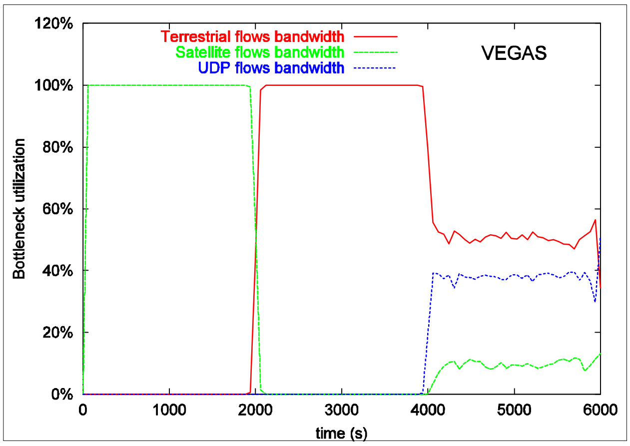

Fig. 6 *Simulation results with TCP Vegas protocol for both terrestrial and satellite connections*

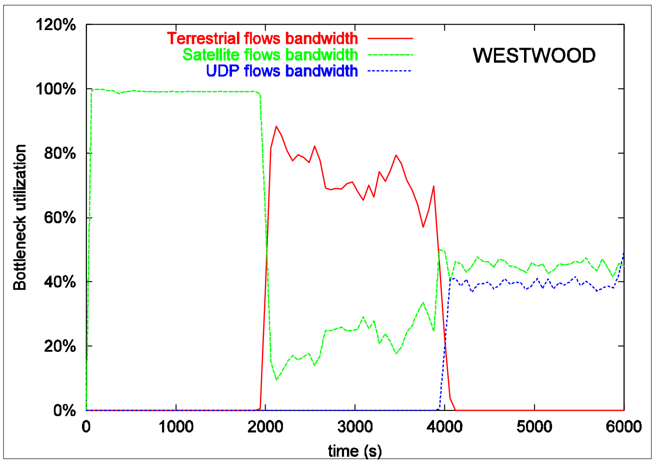

Fig. 7 *Simulation results with TCP Vegas protocol for both terrestrial and satellite connections*

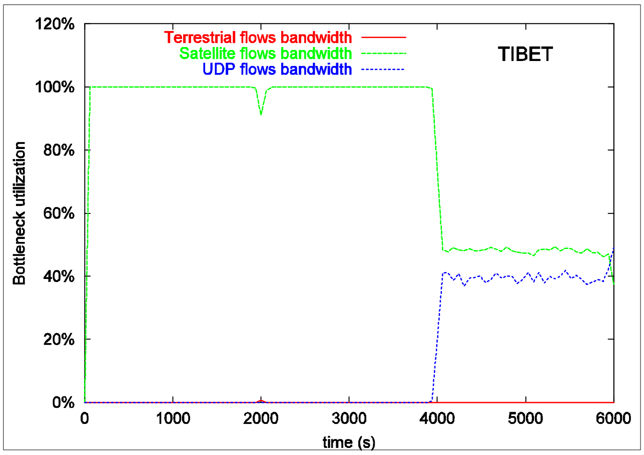

Fig. 8 *Simulation results with TCP Tibet protocol for both terrestrial and satellite connections*

block completely the terrestrial connection for serving the satellite ones.

So, after having obtained these first results, we decided to make a comparison between the Tibet protocol, which demonstrated to be the best protocol for protecting the bandwidth of satellite connections, and the most diffused TCP version actually implemented in network devices: TCP Reno. The following figures show the results obtained in a mixed TCP scenario, in which satellite connection use TCP Tibet and Terrestrial connection use TCP Reno.

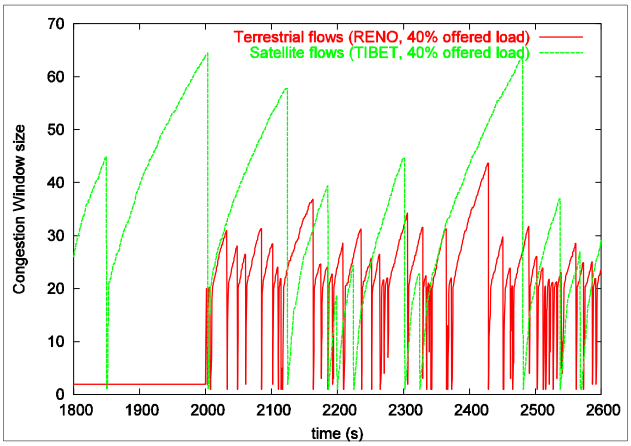

Fig. 10 *Simulation result regarding the TCP congestion windows behavior in low load working conditions*

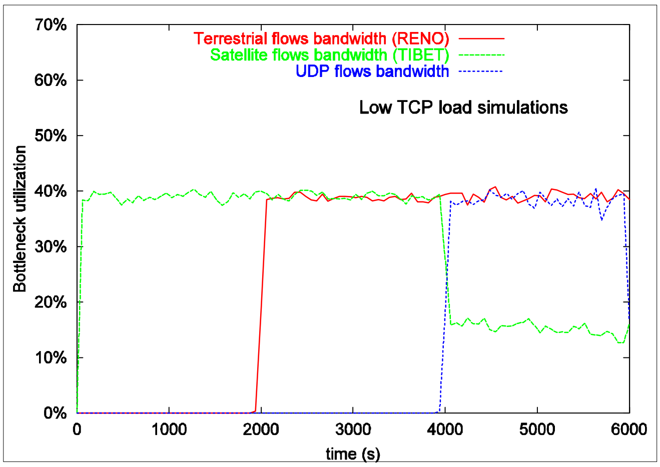

Fig. 9 *Simulation results with satellite TCP Tibet connections and terrestrial TCP Reno connections in a low load working conditions (about 40% for each type)*

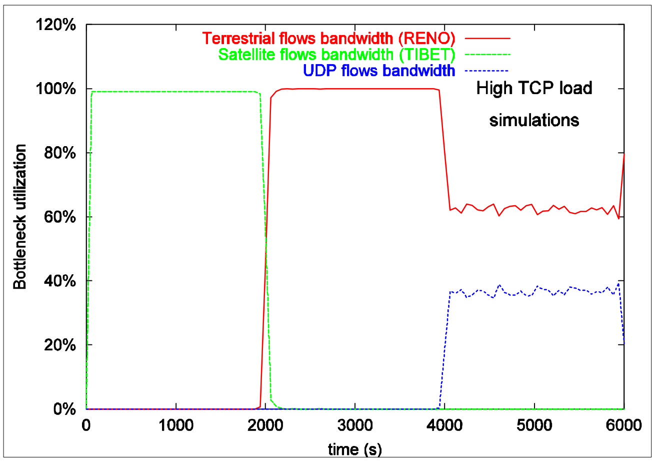

Fig. 11 *Simulation results with satellite TCP Tibet connections and terrestrial TCP Reno connections in a low load working conditions (about 40% for each type)*

7. CONCLUSIONS

The satellite assumes a very important role in the future telecommunication scenario thanks to the capacity of covering long distances and wide area, and in some cases it is the only way for extending telecommunication out of the planet bounds. Today, satellite connections are affected by poor performance partially due to a set of inadequate protocol used in TCP/IP suite.

In this paper, we make a comparison of some proposed TCP extensions evaluating their impact on the performance of satellite connections, which have to contend link capacity with other terrestrial connections. The simulations show that it is possible to introduce in TCP some mechanism for improving satellite throughput and performance, such as Fast recovery and Available Bandwidth estimation as described in TCP Tibet and Vegas.

Unfortunately, the simulations, conducted in a mixed scenario where different version of TCP protocols work sharing the same link, show that even the TCP Tibet protocol, which obtained the best results in homogeneous scenario for satellite connections, is not able to compete with previous implemented TPC protocol for serving link capacity to satellite connection.

In conclusion, our work has put in evidence the inadequate of the actual TCP protocols development in realistic satellite scenarios. Moreover our investigation suggest that more researches must be conducted for realizing a new version of TCP which is able to compensate the long RTT both for improving throughput of satellite link but also for obtaining a fair behaviour in the confront of the previous version of TCP protocols, before that satellite utilization become an everyday task.

REFERENCES

1. D. O'Mahony, *UMTS: the fusion of fixed and mobile networking,* IEEE Internet Computing, vol. 21, Jan.-Feb. 1998.
2. M. Allman, C. Hayes, H. Kruse, S. Ostermann, *TCP Performance over Satellite Links,* Ohio University
3. P. Loreti, M.Luglio, R. Kapoor, J. Stepahnek, M. Gerla, F. Vatalaro, and M. A. Vazquez-Castro, *Satellite Systems Performance with TCP-IP Applications*, IWDC 2001, Taormina, September 2001.
4. S. Floyd, M. Mathis, J.Mahadavi, and A.Romanov, *TCP Selective Acknoledgement Option*, RFC 2018, April 1996
5. S. Floyd, and T.R.Henderson, *The NewReno Modifications to TCP's Fast Recovery Algorithm,* IETF RFC 2582, April 1999
6. J.C: Hoe, *Improving the Start-up Behaviour of a congestion Control Scheme for TCP,* ACM SIGCOMM Computer communications Review, October 1996.
7. A. Capone, and F. Martignon, *Bandwidth Estimates in the TCP Congestion Control Scheme,* IWDC 2001, Taormina, September 2001.
8. W. D. Ivancic, D. Brooks, B. Frantz, D. Hoder, D. Shell, D. Beering, *NASA's Broadband satellite networking research*, IEEE Communications Magazine, n. 7, July 1999, pp. 40-47.
9. C. P. Charalambous, V. S. Frost, J. B. Evans, *Performance evaluation of TCP extensions on ATM over high bandwidth delay product networks,* IEEE Communications Magazine, n. 7, July 1999, pp. 57-63.
10. H. Kruse, S. Ostermann, M. Allman, *On the Performance of TCP-based Data transfers on a faded Ka-Band Satellite Link,* Proceedings of the 6th Ka-Band Utilization Conference, June 2000.
11. M. Allman, H. Kruse, S. Ostermann, *A History of the Improvement of Internet Protocols Over Satellites Using ACTS,* Invited paper for ACTS Conference 2000, May 2000.
12. C. Casetti, M. Gerla, S. S. Lee, S. Mascolo, M. Sanadidi, *TCP with Faster Recovery,* UCLA CS-Technical Report #200017, 2000
13. V. Jacobson, *Congestion Avoidance and Control*, In Proceedings ACM SIG-COMM '88, ACM, August 1988
14. R. Braden, *Requirements for Internet Hosts-Communication Layers*, October 1989, RFC 1122
15. W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, January 1997, RFC 2001
16. J. Postel, *Transmission Control Protocol*, September 1981, RFC 793
17. V. Jacobson, R. Braden, and D. Borman, *TCP Extensions for High Performance*, May 1992, RFC 1323
18. L. S. Brakmo, S. W. O'Malley, L. L. Peterson, *TCP Vegas: New Techniques for Congestion Detection and Avoidance*, 1994 ACM SIGCOMM Conference, pages 24--35, May 1994.
19. "Network Simulator (NS-2)", www.isi.edu/nsnam/ns/.