Study and Performance Analysis of Transport Layer Mechanisms over LEO Satellite Environment

Tomaso de Cola, Mario Marchese
CNIT - Italian National Consortium for Telecommunications, University of Genoa Research Unit
Via Opera Pia 13, 16145 Genova (Italy)
Phone: +39-010-3532806, Fax: +39-010-3532154
e-mail: tomaso.decola@cnit.it, mario.marchese@cnit.it

Abstract. **The paper concerns the study of transmission mechanisms employed in telecommunication environments, where LEO (Low Earth Orbit) orbits are present. Several investigations about novel network architectures have been produced in order to individuate the solution that meets all the network requirements and characteristics in terms of delay, reliability and speed. Two types of solutions are proposed: the first one, where the terminals are modified and no additional tool is inserted in the network; the second one, based on a protocol – splitting philosophy. This latter supposes to add special tools called gateways to improve the performance. The effectiveness of the proposed solution is then evaluated by using a software simulator that has been adapted for the communication in the different protocol architectures studied.**

I. INTRODUCTION AND OVERVIEW

The increasing technology development and the continuous request for multimedia services (e.g. Internet, Video on Demand, e-mail, videoconference) imposes a redefinition of the communication systems in the satellite environment. The study presented in this paper develops in this context and takes as reference point the DAVID project (Data and Video Interactive Distribution), which has been promoted by the Italian Spatial Agency (ASI) in collaboration with the University of Rome "Tor Vergata", the Polytechnic of Milan and CNIT, as scientific partners; Alenia Spazio, Space Engineering and Telespazio, as industrial partners. The main object of our investigation is to design transmission mechanisms aimed at guaranteeing information transfer of very big dimensions with high performances. In order to match these requirements, the TCP /IP protocol – suite [1] is taken as reference, because it is currently widely applied and allows a reliable stream delivery. The problem of more suitable recovery strategies is that, in satellite systems [2] [3], the loss of information is mainly due to channel errors rather than to congestion events, as supposed by the TCP. In this perspective, an important indication is presented in the CCSDS File Delivery Protocol (CFDP) [4], which we consider as a solution suitable for our communication system, at the application layer.

The general configuration [5] is composed as follows. A link in the W band connects the DAVID earth station, which will be located within the Antarctic region, to the DAVID satellite (LEO orbit). An inter – satellite link in the Ka band connects DAVID with the ARTEMIS satellite (geostationary). ARTEMIS is linked to the destination earth station with a communication channel in Ka band. The main aspect of this configuration is due to the visibility window of the LEO satellite, which is not continuous. For this reason, when the satellite is no longer visible to the DAVID terrestrial station or to the ARTEMIS satellite, the transfer is interrupted and it can be restarted when the DAVID satellite is visible again. In particular, in this paper, we present the performance analysis concerning the data communication achieved in the LEO channel, from the DAVID terrestrial station to the DAVID satellite. The paper applies concepts already known in the literature as transport layer splitting, introduces possible protocol architectures to be used and investigates the behaviour of protocols specifically dedicated to this environment. The performance metric is the throughput of the file transfers measured in bytes/s (defined as $\frac{\text{dimension of file}}{\text{overall transfer time}}$).

The paper is structured as follows. Section II contains a more detailed description of the communication over the LEO link and different solutions, aimed at guaranteeing high performances, are presented. Details about the approaches and results that we have obtained during our simulation are presented, in order to individuate the most suitable protocol stack architecture, in Section III and in Section IV for the TCP – based and Protocol – Splitting approaches, respectively. Section V contains a comparison among the best results obtained. Finally, in section VI, the conclusions about the results obtained during the tests are reported and some considerations on the whole performance analysis are indicated.

II. LEO LINKS PERFORMANCE ANALYSIS

The LEO satellite is connected to the DAVID earth station with a link built in the W band (the bandwidth is 100 Mbps), which experiences a propagation delay of about 5ms (Fig. 1). The main characteristic of the communication channel is the fading effect due to meteorological events like rain and by the multipath distortions. Another important aspect of this environment is the satellite visibility, limited of few minutes, which imposes a fast transmission of the whole information set or the need of suspending / resuming operations.

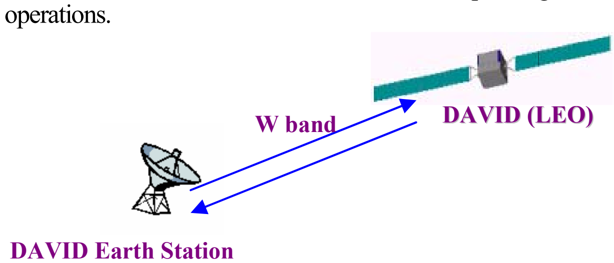

Fig. 1. LEO link

One of the project requirements is to keep a PC using a TCP/ IP protocol suite on the DAVID earth station to send the data towards the destination. The problem may be considered in two different ways, from a network point of view: TCP – based and protocol – splitting approaches. The main difference between the TCP – based and the protocol – splitting approach is represented by the possibility, offered by the latter, to modify (or add) network tools to improve the overall performance. Within this framework a special gateway is used to isolate the satellite portion. Different solutions have been developed and compared in this paper by taking these two approaches as a reference point. Concerning the TCP – based one, a complete protocol stack based on the TCP – IP suite has been considered. The transmission of data is performed by the File Transfer Protocol (FTP) [6][7] at the application layer. The transport layers are constituted by two possible TCP versions: TCP SACK New Reno [8] commonly used and a version of the same protocol taken from the literature and designed to improve the network performance in specific satellite environments. Alternatively a connection – splitting scheme [9] is performed by a gateway that uses the CFDP protocol at the application layer. Concerning the simulation phase, a transfer of information for an amount of 315 Mbytes in order to test the system performance has been considered. Furthermore, the information unit is split into different blocks, in order to evaluate the performance of the system by varying the number of blocks that compose the file. The variation of the block dimension is used also to compare the system performance employing CFDP and FTP protocol solutions and evaluating what happens in the two cases. When FTP is used, a TCP connection is opened for the transmission of each block. The connections are opened in sequence and not at the same time (i.e. the connection I+1 – th is opened only when the I-th is closed). The case of transmission when only one block is used is also investigated in the FTP implementation for the sake of completeness, but, in case of critical link status, it produces throughput values lower than the ones reported when several blocks are employed. This behaviour is due to the error impact, which causes several retransmissions, performed at TCP layer determining performance deterioration. Therefore, in the following, only the case when the information is split in blocks of data is referred. CFDP protocol has been designed to allow the transmission of data divided into blocks. When it is adopted at the application layer, the transmission mechanism depends on the working mode employed, namely reliable and unreliable modes. When CFDP operates with the second option, the need for a connection - oriented transport layer is straightforward. In this case, as in the FTP solution, the transmission of each block of data is performed by opening a connection at the transport layer. Furthermore, on the basis of the MTU path discovery [10], a length of 1500 bytes for each IP datagram has been imposed.

As far as the channel model employed during the simulation study is concerned, a slow fading channel has to be considered, because of the relative motion of the LEO satellite with respect to the earth station. In this perspective, an ideal interleaving system operating together with a FEC mechanism is needed at the datalink layer. In this way, it is possible to mitigate the impairments due to the multipath and rain fading, which strongly affect the quality of the communication. In this way it is realistic to consider BER values (evaluated after the code correction process) ranging from 10^{-7} to 10^{-9}. The percentage of packet corruption is then dependent of the Bit Error Ratio (BER), which has been fixed in each simulation. In detail, the packet error rate (PER) is evaluated as $1-(1-BER)^{l}$ [11], where l is the packet length. If BER<<1, as in our approach, the packet error rate can be approximated with $PER \cong l \cdot BER$. The value of PER is aimed at describing the effect of low layers (physical and data link), seen by the network layer. That's the motivation for using term "packet".

III. TCP - BASED APPROACH

As introduced in the previous section, the protocol implementation in the DAVID earth station follows the TCP – IP suite. As a consequence, if a TCP - based approach is adopted, a full protocol stack, based on the TCP – IP suite, is needed on the DAVID satellite (as indicated in Fig.2).

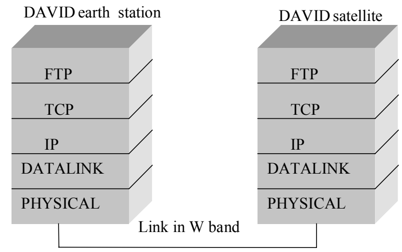

Fig. 2. TCP – based layered architecture

A file transfer procedure is performed at the application layer. Concerning the transport layer protocol specification, two different versions of TCP (Transmission Control Protocol) are considered: TCP New Reno with Selective Acknowledgement (SACK) mechanism [8], where the initial transmission window (IW) is set to 1, the increment of congestion window (cwnd) is unitary and the TCP buffer both at the receiving and at the transmitting side has a dimension of 64 Kbytes; an alternative TCP option where TCP New Reno SACK is characterized by an initial window (IW) increased to 6 segments, congestion window incremented more rapidly at each received acknowledgement and TCP buffer both at the receiving and at the transmitting side of 320 Kbytes. This redefinition of TCP parameters has been studied by using the results of tests operated in satellite environment [12].

In the first case (identified as "FTP-TCP 64K"), if the channel behaves ideally (i.e. BER=0, no losses experienced), a throughput of about 3.2 Mbytes /s is obtained as well as a transfer duration of about 100 s, when the transfer is performed with 35 blocks, which is the minimum number of blocks used. Obviously, if the channel is ideal, the throughput increases when the information fragmentation decreases. The maximum throughput value points out that the channel is not fully used (channel bandwidth amounts to about 12.5 Mbytes /s). It is a consequence of the TCP transmission mechanism. Due to the TCP buffer dimension, the whole transfer requires a long time and affects the overall performance of the system. When the conditions of the channel are more critical (e.g. BER=10^{-7}, as in the first row of Table I, where the Throughput and Transfer time values separated by a '-' are indicated for different values of BER), the information transfer is accomplished in about 103 s, for a maximum throughput value of 3.0 Mbytes /s, when 70 blocks are employed. This poor result is due to the characteristics of the TCP flow control that recognizes only congestion events; as a consequence the recovery is not efficient when the losses are caused by channel errors. When the channel state is less critical (i.e. BER < 10^{-7}), only a slight enhancement is experienced, because of the TCP implementation that does not guarantee a good utilization of the transmission link.

TABLE I

FTP – TCP 64K, 315 MBYTES FILE TRANSFER (THROUGHPUT [MBYTES/S] – TRANSFER TIME [S])

	Number of blocks				
BER	420	140	105	70	35
1.0E-07	2.525 – 124.7	2.969 – 106.0	3.018 – 104.3	3.054 – 103.1	2.500 – 126.0
7.0E-08	2.527 – 124.6	2.970 – 106.0	3.020 – 104.3	3.072 – 102.5	2.901 – 108.5
4.0E-08	2.527 – 124.6	2.970 – 106.0	3.020 – 104.3	3.073 – 102.5	2.963 – 106.3
1.0E-08	2.528 – 124.6	2.972 – 105.9	3.020 – 104.3	3.094 – 101.8	3.138 – 100.3
1.0E-09	2.529 – 124.5	2.972 – 105.9	3.021 – 104.2	3.094 – 101.8	3.139 – 100.3

 0-7803-7974-8/03/$17.00

In the second option (FTP with modified version of TCP, identified as "FTP-TCP opt", in the following), when the channel behaves ideally, a maximum throughput value of 12 Mbytes /s is obtained as well as a transfer duration below 27s (with 35 blocks). As in the previous case, the best performance is provided by the configuration with a minimum number of blocks. The enhancement, with respect to the previous case, is due to the different transmission mechanism during the slow- start phase and, mainly, to the varied buffer dimension. When the link state is critical it is necessary to spend some more words about the behaviour of the global system (described in Table II, where the Throughput and Transfer time values are indicated). In particular, when the number of losses experienced by the channel is consistent (BER=10^{-7}, BER=7.0 $\cdot 10^{-8}$ on the first and second row of Table II), a maximum throughput of about 10 Mbytes /s with transfer duration of 31.7s (when 105 blocks are employed) has been measured. Again, the intermediate configurations (70-140 blocks) provide the best performance, whose peak is measured for 105 blocks, in this case. When the link state case is definitely less critical, i.e. BER = 10^{-8}, 10^{-9} the results indicate a better utilization of the transmission channel, a global throughput of about 11.6 ÷ 11.7 Mbytes/s is obtained, with 35 blocks (the minimum number of block used).

TABLE II
FTP – TCP OPT, 315 MBYTES FILE TRANSFER (THROUGHPUT [MBYTES/S] – TRANSFER TIME [S])

		Number of blocks				
		420	140	105	70	35
BER	1.0E-07	6.635 – 47.4	9.208 – 34.2	9.924 – 31.7	9.370 – 33.6	3.540 – 88.9
	7.0E-08	6.732 – 46.7	9.318 – 33.8	9.928 – 31.7	9.390 – 33.5	3.558 – 88.5
	4.0E-08	6.836 – 46.0	9.408 – 33.4	9.933 – 31.7	9.446 – 33.3	4.191 – 75.1
	1.0E-08	6.931 – 45.4	9.630 – 32.7	10.297 – 30.5	10.927 – 28.8	11.634 – 27.0
	1.0E-09	6.931 – 45.4	9.633 – 32.7	10.301 – 30.5	10.931 – 28.8	11.720 – 26.8

IV. PROTOCOL - SPLITTING APPROACH

A connection – splitting system is designed together with an adapted solution in this case. There is no constraint about inserting tools and components to improve the performance; the basic idea is to separate the satellite portion from the rest of the network in order to optimise the data communication on the satellite link by inserting a gateway in the middle, which also allows preserving, in the DAVID earth station, the full protocol stack based on TCP – IP suite. The general architecture is depicted in Fig. 3, where the three dashed components are the object of the following study.

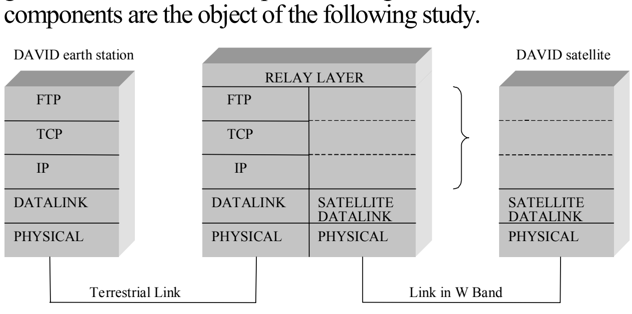

Fig. 3. Protocol – splitting general architecture

The system structure is composed of a terrestrial station (divided into two platforms) and a satellite one. For what concerns the station on earth, the first platform is the transmitting station (indicated as DAVID earth station in Fig. 3); the second one is the gateway that effectively manages the connection – splitting. From the point of view of the protocol stack architecture, a file transfer, which closes at the gateway, is performed via FTP in the first portion of the network. CFDP can operate either in reliable or in unreliable mode, the underlying layers will be defined consequently. In more detail, when CFDP is configured in reliable mode, the transport layer is no longer necessary because the loss recovery is performed at the higher layer; for this reason CFDP communicates directly with the data link layer. On the other hand, when CFDP operates in unreliable mode, the transport layer is necessary, because a loss recovery mechanism is needed. It is important to note that, when the meteorological conditions are critical, the CFDP implementation allows to interrupt the transfer session and to resume it, without restarting it from the beginning.

A. *CFDP unreliable mode*

The scheme is shown in Fig. 4.

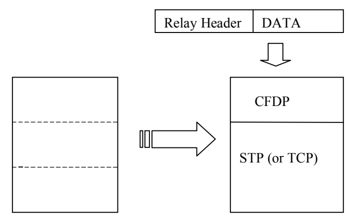

Fig. 4. CFDP protocol operating in unreliable mode

The reliability of the service has to be guaranteed by a proper transport layer. In order to evaluate the performance of the system more deeply, two different implementations of the Transport Layer are considered: TCP New Reno SACK modified version presented in the Section III (IW=6, modified slow start and TCP buffer = 320 Kbytes) and Satellite Transport Protocol (STP), which has been defined just to meet the link characteristics and to allow the full utilisation of the transmission channel. The first configuration is called "CFDP – TCP opt" and the second one "CFDP – STP".

Concerning "CFDP-TCP opt", the conclusions are the same as in the "FTP-TCP opt" solution in Section III. In detail, when the data transfer is managed in clear sky conditions (BER=0), the maximum throughput is about 12 Mbytes /s, if 35 blocks are employed. When the channel is in a very critical state (i.e. BER=10^{-7}, BER=7.0 10^{-8}, BER= $4.0 \cdot 10^{-8}$) the maximum throughput is about 10 Mbytes /s, when a number of 105 block is used. When the channel condition gets better (BER=10^{-8}, BER=10^{-9}), the overall performances report a maximum throughput value close to 12 Mbytes/s. Concerning "CFDP-STP", the definition of the Satellite Transport is achieved by exploiting the knowledge about the network state. In this sense, it is possible to impose a transmission rate such to allow the full utilisation of the channel. As a consequence, the STP transmission window is dimensioned in dependence of the Delay – Bandwidth product so to guarantee the full utilisation of the channel. For what concerns the loss recovery mechanism, a selective retransmission mechanism is adopted; once the recovery is accomplished the transmission phase is re-entered without modifying the dimension of the transmission window. When there is a clear sky condition (BER=0), there is the full utilization of the channel, which corresponds to a maximum throughout of about 12.45 Mbytes /s and to a total duration of the service of 23 s, when 35 blocks are used. When the losses experienced by the transmission medium are consistent (BER=10^{-7}, BER=$7.0 \cdot 10^{-8}$, BER=$4.0 \cdot 10^{-8}$), the maximum throughput reported (with 105 and 70 blocks) is close to 12.0 Mbytes /s, as indicated in the first rows of the Table III, where the Throughput and Transfer time values separated by a '-' are indicated for CFDP – STP configuration. It is important to observe that the number of blocks has a lower influence on the performance than in the cases considered in Section III. All the options provide satisfying results because the improvement is due to the different loss recovery mechanisms that recognize correctly the kind of loss (not due to congestion) and do not limit the transmission speed, as in the TCP implementation. When the channel state is less critical (BER=10^{-8}), analogous results are reported and, in particular, the system almost fills the channel bandwidth by using 35 blocks. If the value of BER is equal to 10^{-9}, as indicated in the last row of Table III, the channel behaves almost ideally.

TABLE III

CFDP – STP, 315 MBYTES FILE TRANSFER (THROUGHPUT [MBYTES/S] – TRANSFER TIME [S])

		Number of blocks			
		420	105	70	35
BER	1.0E-07	11.060 – 28.4	11.913 – 26.4	11.907 – 26.4	11.696 – 26.9
	7.0E-08	11.064 – 28.4	11.915 – 26.4	11.907 – 26.4	11.696 – 26.9
	4.0E-08	11.064 – 28.4	11.920 – 26.4	12.074 – 26.0	11.912 – 26.4
	1.0E-08	11.064 – 28.4	12.195 – 25.8	12.275 – 25.6	12.279 – 25.6
	1.0E-09	11.065 – 28.4	12.200 – 25.8	12.280 – 25.6	12.284 – 25.6

B. CFDP reliable mode

Now the case where CFDP operates in reliable mode and performs the loss recovery ("CFDP reliable" solution) is considered. No transport layer is needed and the application layer communicates directly with the datalink layer, as shown in Fig. 5. When CFDP operates in reliable mode, the transmission is managed by an acknowledgement scheme; in particular, during the information transmission (that is effectively studied in the simulations), Negative Acknowledgements (NAK) are employed. In this implementation, when no errors are introduced by the channel (i.e. BER=0), the full saturation of the channel (with 35 blocks) is reached. When the channel status is critical (BER=10^{-7}), the weight of retransmissions is very relevant. The performance provided is low: the throughput values range from 6 Mbytes /s (420 blocks) to 2 Kbytes /s (35 blocks) and the transfer duration in the latter case is more than 150000 s.

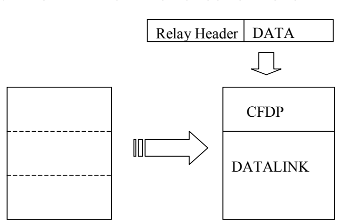

Fig. 5. CFDP protocol operating in reliable mode

The impact of block retransmission on the performance is very heavy: the transmission can be performed also many times and this procedure requires a long time because of the block dimension, which has a great importance. The best result, in fact, is measured for a high fragmentation (420 blocks), i.e. when the block dimension is limited. An overall file transfer of 51.3s is obtained in the best case. The results are indicated in Table IV, which contains the Throughput and Transfer time values for CFDP – reliable, 315 Mbytes file transfer. If BER is less severe, the losses experienced are reduced and the retransmission phase is less heavy. More precisely, if the transmission medium presents a BER of 10^{-9}, as in the last row of Table IV, a very little number of losses is experienced. The overall throughput ranges from 11.5 Mbytes/s (with 140 blocks) to 9.7 Mbytes/s (with 35 blocks).

TABLE IV

CFDP- RELIABLE, 315 MBYTES FILE TRANSFER (THROUGHPUT [MBYTES/S] – TRANSFER TIME [S])

		Number of blocks				
		420	140	105	70	35
BER	1.0E-07	6.140 – 51.3	2.359 – 133.5	1.435 – 219.5	0.374 – 842.2	0.002 – 157500
	7.0E-08	7.258 – 43.4	3.626 – 86.8	2.710 – 116.2	1.057 – 298.0	0.100 – 3150.0
	4.0E-08	8.784 – 35.8	5.785 – 54.4	4.584 – 68.7	3.115 – 101.1	0.835 – 377.2
	1.0E-08	10.534 – 29.9	9.716 – 32.4	8.945 – 35.2	7.597 – 41.4	5.601 – 56.2
	1.0E-09	11.122 – 28.3	11.458 – 27.4	11.302 – 27.8	10.827 – 29.0	9.706 – 32.45

V. COMPARISON OF THE RESULTS

This section contains the comparison of the solutions investigated in sections III and IV. The configurations assuring the best results are selected for each solution. Summarizing: "FTP –TCP 64K" and "FTP –TCP opt" indicate the two TCP – based solutions proposed in Section III. "CFDP – STP" and "CFDP – TCP opt" represent the two protocol – splitting solutions proposed in Section IV where CFDP protocol operates in unreliable mode. "CFDP reliable" is the protocol – splitting based solution in Section IV where the CFDP protocol operates in reliable mode and communicates directly with datalink layer. In clear sky conditions (i.e. BER=0), the "CFDP – STP" and the "CFDP reliable" solutions give the best result and substantially provide the same performance (both with 35 blocks): the full saturation of the channel (about 12.5 Mbytes /s) is reached. The solution "FTP–TCP 64K" offers low performance: the throughput is always less than 3.5 Mbytes /s. The other solutions are very satisfying and allow a maximum throughput of about 12 Mbytes /s. When the link state is very critical, that is the number of lost packets is relevant (BER=10^{-7}, BER=$7.0 \cdot 10^{-8}$ as shown in Fig. 6), the system performance strongly depends on the solution applied. The "CFDP – STP" solution (105 blocks) provides the best result. It offers a throughput of about 12 Mbytes /s. The other solutions reach a throughput of: 10 Mbytes /s ("FTP – TCP opt" and "CFDP – TCP opt", both with 105 blocks), 6 Mbytes /s ("CFDP reliable", 420 blocks) and 3 Mbytes /s ("FTP – TCP 64K", 70 blocks). The real advantage is in the use of a transport totally adapted to the link used. STP is applied together with CFDP, which is suited to transmit information in blocks and allows interrupt/resume mechanisms that could be of help for DAVID limited period of visibility. When the channel behaviour is less critical (BER=$4.0 \cdot 10^{-8}$, BER=10^{-8}) the retransmission phases required in order to recover fully the lost packets are less heavy; but a full occupation of the bandwidth is only assured by the "CFDP – STP" solution (with 70 and 35 blocks respectively) that offers a global throughput value of above 12 Mbytes/s. With the other configurations considered, the system performances are lower; in particular, the " CFDP – TCP opt " and " FTP – TCP opt " solutions can offer a throughput value of about 10 Mbytes/s (BER = $4 \cdot 10^{-8}$, 105 blocks) and 11.6 Mbytes/s (BER = 10^{-8}, 35 blocks), but they cannot guarantee the result obtained with the "CFDP – STP" option because the transport layer is not fully optimised. A good result is also guaranteed by "CFDP reliable" even if the phase of block retransmission is very burdensome and limits the throughput to values ranging from 8.8 Mbytes/s to 10.5 Mbytes/s, lower than in the previous configurations. Finally, the solution “FTP – TCP 64K” presents a minimal utilization of the channel (the simulations reported a throughput value of about 3 Mbytes/s), because of the TCP specification that does not assure an effective utilization of the transmission medium in these conditions.

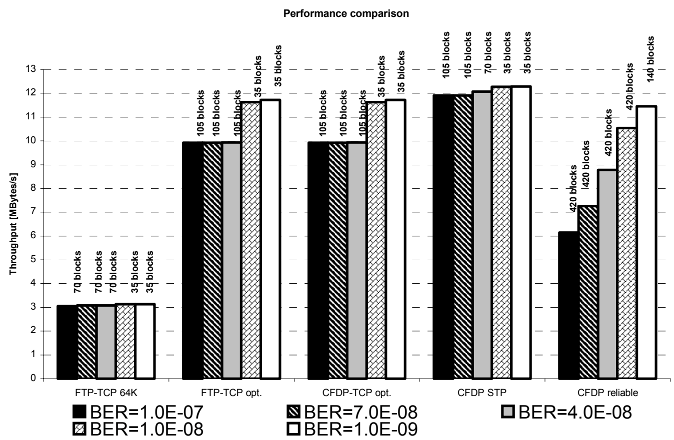

Fig. 6. Comparison of performance experienced by the different protocol architectures employed

The Satellite Transport Protocol (STP) has been designed to mitigate TCP drawbacks. It manages the link errors, allows the full utilization of the channel bandwidth and guarantees a minimum duration of the data transfer, which is, together with the use of CFDP, very important due to the limited visibility of the DAVID satellite. This result matches the protocol – splitting philosophy that allows designing solutions adapted to the link characteristics. The drawback regards implementation complexity, relevant also for future on – board tools. On the other hand, when a reliable application layer (namely "CFDP reliable″) is involved, it is clear that the retransmission of lost blocks deteriorates the overall performances and that this type of solution can provide good performance only when BER is low.

VI. CONCLUSIONS

The investigation has been addressed to the study of data communication over LEO satellite environment, in order to design protocol architectures suited to the considered network. TCP – based and protocol – splitting approaches have been employed and compared in order to evaluate the performance. The splitting architecture allows the use of tools to separate the satellite side from the rest of the network and the design of a suited protocol stack. A particular attention has been devoted to CFDP protocols implemented at the application layer, whose employment is fully justified by the limited visibility window, which determines the interruption of the data communication. CFDP adopts a suspend/resume procedure effectively suitable for this kind of scenario. The values of throughput and transfer time have revealed the convenience of implementing a transport protocol (identified as STP) fully tuned to the satellite link characteristics, together with the CFDP protocol applied in unreliable mode. This solution provides very good performance also when the channel status is critical and guarantees a full use of the bandwidth available. The drawback is the complexity of the on – board implementation, also for future components.

REFERENCES

[1] Information Sciences Institute, University of Southern California, “Transmission Control Protocol – Darpa Internet Program – Protocol Specification”, IETF, RFC 793, September 1981.

[2] 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.

[3] 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", IETF, RFC 2760, February 2000.

[4] Consultative Committee for Space Data Systems (CCSDS), CCSDS File Delivery Protocol, CCSDS 727.0-R-5, Red Book, August 2001.

[5] C. Bonifazi, M. Ruggieri, M. Pratesi, A. Solomi, G. Varacalli, A.Paraboni, E. Saggese, “The David Satellite Mission of the Italian Space Agency: High Rate Data Transmission of Internet at W and Ka bands”, 2001 (http://david.eln.uniroma2.it).

[6] J. Postel, J. Reynolds, “ File Transfer Protocol (FTP)”, IETF, RFC 959, October 1985.

[7] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", IETF, RFC 1323, May 1992.

[8] M. Mathis, J. Mahdavi, A. Romanow, "TCP Selective Acknowledgement Options", IETF, RFC 2018, October 1996.

[9] V.G. Bharadwaj, J. S. Baras, N. P. Butts, "An Architecture for Internet Service via Broadband Satellite Networks", International Journal of Satellite Communications, Special Issue on IP, Vol. 19, Issue 1, January/February 2001, pp. 29-50.

[10] J. Mogul, S. Deering, “Path MTU Discovery”, IETF, RFC 1191 (Obsoletes RFC 1063), November 1990.

[11] Marco Chiani, Enrico Milani, Roberto Verdone, "A semi-analytical approach for performance evaluation of TCP-IP based mobile radio links", GLOBECOM 2000 - IEEE Global Telecommunications Conference, no. 1, November 2000 pp. 937-942

[12] M. Marchese, “TCP Modifications over Satellite Channels: Study and Performance Evaluation”, International Journal of Satellite Communications, Special Issue on IP, Vol. 19, Issue 1, January/February 2001, pp. 93-110.