Performance Analysis of Data Transfer Protocols over Space Communications

TOMASO DE COLA

MARIO MARCHESE, Member, IEEE
Italian National Consortium for Telecommunications

The work presented here concerns the study and the investigation of data transmission mechanisms, employed above the network layer of the protocol stack, used to transmit a data file through an end-to-end path composed of low Earth orbit (LEO) and geostationary (GEO) portions. A link built in the unexplored band W connects the Earth station to a LEO satellite (called DAVID). The communication among DAVID and other sites is performed by means of an inter-satellite link in Ka band, which connects the LEO satellite to a GEO one (called ARTEMIS) and the latter to the Earth stations. The main peculiarities of the investigated scenario consist of the visibility of a LEO satellite, limited to a few minutes, and of the only partial availability over time of a return link on the GEO path. LEO visibility may seriously affect the data communication because it implies suspend/resume mechanisms to match nonvisibility periods. The partial unavailability of the return link makes (transmission control protocol) TCP-based solutions barely applicable and implies the design of effective protocol architectures, in order to guarantee a reliable data communication.

Several investigations about alternative novel architectures have been produced in order to individuate solutions that meet all the network requirements in terms of service reliability, exploitation of the network resources, and overall service time. The analysis is two-fold: on one hand the attention has been addressed to the LEO path, highlighting the features offered by a transport layer splitting approach and by an application-based solution exploiting the CCSDS (Consultative Committee for Space Data Systems) protocol stack, whose performance is compared with the results provided by the TCP/IP protocol stack, commonly used in the Internet (where IP stands for Internet Protocol). Concerning GEO path, assuming the return link only partially available, two alternatives of the CCSDS protocol stack have been investigated: the first one implies the use of a negative acknowledgement scheme, the second one does not require any availability of a return link. The two paths together compose the end-to-end communication, which is also globally analyzed on the basis of the performance offered by the solutions presented for the two portions. All the results have been obtained by using a simulator developed for the aim.

Manuscript received April 15, 2003; revised January 26, 2005; released for publication April 14, 2005.

IEEE Log No. T-AES/41/4/860793.

Refereeing of this contribution was handled by M. Ruggieri.

This work was performed within the framework of the DAVID project and funded by the Italian Space Agency under Contract I/R/238/00.

Authors' current addresses: T. de Cola, CNIT (Italian National Consortium for Telecommunications), University of Genoa Research Unit, Via Opera Pia 13, 16145 Genova, Italy; M. Marchese, Dept. of Communication, Computer and Systems Science, University of Genoa, Genova, Italy, E-mail: (mario.marchese@cnit.it).

I. INTRODUCTION

The increasing technology development and the continuous request of multimedia services (e.g. Internet, Video on Demand, e-mail, videoconference) [1] imposes a redefinition of the communication systems in the satellite environment. In particular, research activities consider the problems related to the service distribution and hence to territory cover. The study presented here develops in this context of data dissemination and refers to the DAVID project (data and video interactive distribution) [2], which has been promoted by the Italian Space Agency (ASI) in collaboration with the University of Rome "Tor Vergata," the Polytechnic of Milan, and CNIT, as scientific partners, and Alenia Spazio, Space Engineering, and Telespazio, as industrial partners.

The investigated environment is composed of an Earth station, a low Earth orbit (LEO) satellite (DAVID), a geostationary (GEO) satellite (ARTEMIS), and a destination Earth station. The aim is transmitting data files through the described path. The first hop is characterized by a link, built in the experimental band W, connecting the Earth station, located in the Antarctic region, with the LEO satellite [3]. The main peculiarity of this portion is represented by the limited visibility of the DAVID satellite, which imposes strict time constraints on the scheduling of data delivery operations and requires a full use of the available bandwidth. The second hop is composed of an inter-satellite link, in the Ka Band, connecting DAVID (where the data coming from the first hop are stored) with the Earth destination station through the ARTEMIS platform. This portion deserves particular attention, because the return link (destination-DAVID) is not always available, and a data transmission scheme based on a continuous feedback (such as in TCP-based solutions (transmission control protocol) [4]) is not applicable.

In order to meet the requirements of DAVID scientific mission, the performance analysis has been carried out considering an amount of data to be transferred equal to 306.6 Mbytes. Even though the telecommunication network investigated here explicitly refers to the DAVID project framework, the protocol analysis has a wider validity. The whole investigation may be extended without loss of generality to any multi-hop satellite environments (e.g. Galileo, GPS, Iridium systems); in this perspective, this work proposes a set of architecture alternatives for data file transfer to be applied over a more general heterogeneous satellite scenario. Concerning the LEO portion, this work compares the performance of three different approaches: a TCP-based architecture; a transport layer splitting architecture, where additional agents speed-up the performance over the LEO portion; and a full CCSDS (Consultative Committee for Space Data Systems) protocol stack, where the transfer is performed through CCSDS file delivery protocol (CFDP).

In all the considered cases, LEO link is bidirectional and allows exchanging information (data and acknowledgment) between DAVID and Earth. The file information is divided into blocks and the analysis of the performance in dependence of the blocks' dimension is one of the most operative and interesting results presented here.

As far as the GEO portion is concerned, assuming the return channel available only partially (with probability p) over time, only CCSDS architectures are evaluated. More specifically, CFDP acting in reliable mode (by using a negative acknowledgment scheme when the return channel is on) is compared with a new proposal of the authors based on CFDP acting in unreliable mode. Also in this case, the information to send is structured into blocks whose dimension heavily affects the performance of the transfer and the design choice.

The performance analysis has been accomplished by employing a proper simulation tool developed for the aim; for each simulation, the validity of the tests has been checked considering 40 runs, always sufficient to assure a confidence interval of at least 95%.

The paper is structured as follows. Section II is devoted to the state of the art of research activities within satellite environment, outlining the role played by the CCSDS [5] in the design of protocol specifications suitable for space communications. A particular emphasis on the CFDP is given in Section III in order to show the advantages offered by such approach and its applicability in the investigated scenario. Section IV and V address the performance analysis of data communication accomplished in the LEO and GEO portions respectively, while Section VI shows the impact of the proposed architectures over the end-to-end performance of the communication. Section VII contains the conclusions and possible future directions of the proposed work.

II. RELATED WORKS

Data transfer over satellite links raises many problems due to the peculiarities of the channels. Within an Internet environment (more exactly a TCP/IP-based environment), the main problem is linked to the implementation of the TCP. As indicated in the RFC 1323 [6], the TCP performance does not depend upon the transfer rate itself, but upon the product of the transfer rate with the round-trip delay (RTT), namely the "bandwidth-delay product," which measures the amount of unacknowledged data that TCP must handle in order to exploit the whole channel bandwidth. TCP performance problems arise when the bandwidth-delay product is large. It is typical in satellite links [7], where the high propagation delay makes the acknowledgement arrival slow and the transmission window needs a long time to grow. Another problem concerning TCP over satellite networks, which are heavily affected by noise, is represented by its reaction in presence of transmission errors. It is well known that TCP is not able to distinguish congestion events from link errors. The protocol reduces the transmission window size [8] at each loss, independently of the cause, degrading the overall communication performance, when the loss is not due to congestion.

The problem of improving TCP over satellite has been widely investigated in the literature [6–9]. Many schemes aimed at mitigating the impairments introduced by nonterrestrial links have been proposed and analyzed [10]. A possible classification, taking emphasis on the design issues [11], is reported in the following.

1) Pure Transport Layer. It consists of protocol specifications acting on the transport layer and properly designed for providing effective data communications in the considered environment. The specifications are implemented in software over the terminal hosts. TCP modifications and tuning belong to this class.

2) Hard State Transport Layer. This approach encompasses all the schemes based on connection-splitting and spoofing mechanisms, provided by a specialized gateway operating as an intermediate agent and interfacing the terrestrial network with the satellite portion.

3) Soft State Cross-Layer Signaling. It includes specific techniques implemented at the layers acting below transport layer (namely network and datalink) aimed at notifying the transport layer about the satellite channel state, in terms of traffic load or percentage of lost packets. Signaling may be used to distinguish loss due to link errors and to congestion events.

4) Application Layer. This approach refers to enhancements directly operated at the application layer in order to improve the performance of data communication. In practice, the requirement of data transfer reliability is shifted from the transport to the application layer. This solution is often used keeping unchanged the transport and the underlying layer protocols, even if, in particular environments, it is worthwhile introducing a dedicated protocol stack, as done by the CCSDS approach considered in the work presented here.

Concerning pure transport, significant proposals about TCP congestion control scheme have been produced. RFC 2488 [12] and RFC 2760 [13] list the main limitations of the TCP over satellite and proposes possible methods to act. Regarding the protocol proposal, TCP Peach+ [14] exploits available network resources by employing a probing-based scheme by means of nihil segments. An alternative congestion control is implemented in the TCP Westwood [15], able to evaluate the available resources on the path by means of bandwidth estimation techniques. Explicit control protocol (XCP) [16] generalizes the explicit congestion notification (ECN) [17] scheme and introduces the new concept of decoupling utilization control from fairness control.

Hard state transport layer schemes may be associated with performance enhancing proxies (PEP) architectures. RFC 3135 [18] deals with this concept by analyzing the mechanisms implemented at different layers of the protocol stack. In more detail, to improve the performance, the satellite portion of a network may be isolated and receive a different treatment with respect to the other components of the network. Methodologies as TCP splitting [19, 20] and TCP spoofing [21] bypass the concept of end-to-end service by dividing the TCP connection into segments managed by intermediate specialized gateways. This approach allows designing a proper ad-hoc protocol stack on the satellite side and hence optimizing the performance.

The soft cross-layer signaling approach is based on the knowledge of the transmission channel state, which is communicated to the transport protocol. In this approach transport protocol has the possibility of distinguishing whether a packet loss is due to congestion events or link errors. The satellite-link aware communication protocol (S-LACP) [22] is able to enhance the performance of TCP over satellite links. It provides a unified interface for controlling the capabilities of the satellite service, able to perform QoS and resource management packet loss notifications, transparently to the upper layers. The explicit transport error notification (ETEN) and its variants [23] refer to other notification forms such as ECN and explicit loss notification (ELN). They allow TCP to discriminate the loss nature and to react accordingly, avoiding unnecessary bandwidth wasting if possible.

Application-based solutions include schemes introduced directly at the application layer and also modifications operated at the socket interface in order to enhance the overall performance. In this sense, it is worth mentioning the extendend file transfer protocol (XFTP) [24] that presents a more aggressive transmission behavior, compared with file transfer protocol (FTP) [25], achieved by opening multiple simultaneous TCP connections. This solution, however, as pointed out in [26], has not general portability and its application in large networks may lead to congestion collapses. An alternative solution is based on coding techniques, such as erasure-coding schemes [27] performed at the application layer, which allow making the communication more robust. An example is given by the repeated transmission of the same block of data.

Due to its importance, particular attention needs to be addressed to the CCSDS, which did a great standardization effort, in terms of design and implementation of protocol solutions suitable for space communications. The following is worth mentioning.

1) The SCPS-TP (also known as TCP-Tranquility) specification [28], enhances the performance of TCP via satellite, by exploiting the benefits of TCP Vegas and by implementing signaling schemes, managed from the underlying layers, which help distinguish congestion events from transmission errors.

2) The CFDP [29] and proximity-1 space link protocol [30], are designed for the employment at the application layer and at the datalink layer, respectively. The former is responsible of managing the reliability of data communication by means of retransmission mechanisms together with suspend/resume operations. The latter defines coding and modulations techniques suited to environments experiencing high bit error ratio (BER). The overall approach may be classified within "application-based" solution, even if, in this case, a full protocol stack is used in alternative to TCP/IP suite.

The CFDP is topical for the study performed in this work, whose novelty is mostly related to a detailed performance investigation of this protocol over a heterogeneous satellite network. More details are reported in next section.

III. CCSDS FILE DELIVERY PROTOCOL

Usually, space networking has to deal with hazardous communication conditions, such as long propagation delay, intermittent link connectivity, and high BERs. In this environment, the employment of conventional automatic repeat request (ARQ) schemes is discouraged and specific protocols are preferred in order to assure the reliability of the data communication [31]. In this view, a possible choice is represented by the CFDP, which implements an enhanced ARQ scheme based on negative acknowledgments (NAK). It manages file transfers as the FTP, but extends its capabilities in terms of suspending and resuming operations.

From this point of view, the recommendation specifies two possible operative procedures, namely core and extended. The former corresponds to a point-to-point exchange of data, where no further agents are required in the middle. The latter allows multi-hop data communications, accomplished through intermediate agents able to store the data in local mass-storage units and then to forward it to the next hop. It is straightforward that the "extended" approach is really suitable for the investigated scenario, composed of a "postman" LEO satellite that receives the data from the Earth at each passage, stores them up to the visibility of a GEO satellite and downloads them to the destination through the GEO portion when possible.

Concerning the reliability management, two possible modes are provided [29].

1) CFDP unreliable. CFDP itself is not responsible for a reliable data delivery, which needs to be guaranteed by the underlying layers, i.e., TCP. A possible use of CFDP unreliable is introduced by the authors for the GEO portion and it is called CFDP-repeat. Details are provided in the remainder of this work.

2) CFDP reliable. CFDP itself is responsible for the reliability of the data communication, which is assured by a NAK scheme, implemented through the transmission of NAK PDUs that carry information about lost PDUs (actually blocks of data whose dimension heavily affects the performance). Four possible schemes of retransmission are possible: immediate, asynchronous, prompted, and deferred. In the following, a short characterization is given for possible retransmission schemes. The first two have been extensively considered throughout the remainder of this work.

In NAK immediate mode, the NAK PDUs are issued by the CFDP entity as soon as out of order data PDUs are received, in order to solicit the retransmission of missing PDUs; a retransmission timeout is associated with each NAK notification; if the number of retransmissions for the same missing DATA PDU exceeds a maximum value (assumed equal to 15 here), the data communication is aborted.

In NAK asynchronous mode, the schemes necessary to assure the reliability of the communication are the same as defined above, including NAK issuances and a retransmission timeout associated with each NAK notification, but NAK notifications are not immediately released when out of order PDUs are received. They are triggered by asynchronous events such as manual interventions or channel availability fluctuations (typical for intermittent link connectivity). This scheme is really promising in the investigated scenario, due to the partial availability of the return link on the path between DAVID satellite and Earth terminals through the GEO satellite ARTEMIS.

In NAK prompted mode, NAK PDUs are transmitted only when solicited by the data source.

In NAK deferred mode, NAK PDUs are emitted only when "end-of-file" arrives at the destination; the correct reception of all the blocks is checked and, in case of some failure, NAKs are produced for errored blocks.

The choice of investigating only "NAK immediate mode" and "NAK asynchronous" is due to the characteristics of the network. Concerning the LEO portion, the return link is always available and "NAK immediate mode" seems to be simple and efficient. Neither "prompted" nor "deferred mode" seem to have the possibility of improving the performance. Concerning GEO link, due to the partial availability of the return path, the "asynchronous mode" is the only possible choice, concerning reliable modes.

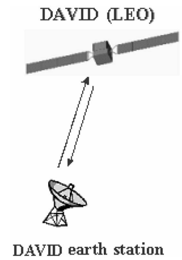

Fig. 1. LEO path configuration.

IV. LEO LINKS PERFORMANCE ANALYSIS

A. Overview

DAVID Earth station is connected to the DAVID satellite platform by a link built in the unexplored band W [32], experiencing an RTT equal to 10 ms and offering an available bandwidth, seen by the network layer, equal to 102.4 Mbit/s. The main peculiarity of the system is the DAVID visibility window whose visibility is limited to several minutes [33] a day. The visibility period ranges from 187 s to 309 s, for each passage, and the number of passages a day varies from 2 to 7. In these conditions, it is very important to use all the available bandwidth to exploit the temporary availability of the satellite. The reference scenario is sketched in Fig. 1.

Concerning the transmission channel characterization, due to the orbit and to the high frequency transmission, it is affected by two main factors:

1) attenuation caused by meteorological events such as rain and storms,

2) mobility of the satellite with respect to the Earth station, which causes phenomena of multi-path fading and shadowing that attenuate the strength of the transmitted signal.

In order to mitigate these effects, a forward error correction (FEC) scheme, based on convolutional punctured codes together with the employment of interleaving techniques, is applied. In facts, it is possible to reduce the rough BER (about $10^{-1} \div 10^{-2}$) to values ranging from 10^{-6} to 10^{-12}. In the remainder of the paper we refer to the BER evaluated after the coding/decoding operations.

The tests have been performed by adopting an IID (independent identically distributed) channel model, in which the corrupted bits are uniformly distributed within each packet. In particular, BER values ranging from 10^{-6} to 10^{-9} have been considered. The performance analysis about data communication has been accomplished considering a file transfer of 306.6 Mbytes; the metrics are: the normalized throughput N_thr and the file transfer time. The normalized throughput is evaluated as the ratio between the final throughput $F\_thr =$ (file dimension/file transfer time), experienced by the data communication and the available bandwidth B seen at the network layer, and expressed as $N\_thr = F\_thr/B$.

In order to have a detailed analysis of data communication over the LEO link, the following protocol configurations have been considered:

1) TCP-based architecture, which includes an FTP-like application running on both the Earth station and the DAVID satellite platform;

2) Transport layer splitted architecture, consisting in the addition of a specialized gateway responsible for isolating the terrestrial portion from the wireless side and managing an ad-hoc transport protocol on the satellite side;

3) CCSDS-based architecture, consisting in a CFDP application running on both the Earth DAVID station and satellite platform.

B. TCP-Based Architecture

T1It is characterized (Fig. 2) by a full TCP/IP protocol stack with an FTP-like [25] application on its top. Particular attention has to be addressed to the transport layer protocol. At a first instance, TCP-NewReno [34] supporting SACK [35] option and TCP buffers, on both the sender and receiver sides, with capacity of 64 Kbytes is considered. The configuration is identified in the following as TCP-IW2-64K. In order to cope with the limitations introduced by the satellite channel in terms of high bandwidth-delay product, also an enhanced TCP configuration, evaluated in previous works (e.g. [36]) of the same research group is considered. It is based on TCP-NewReno specification with SACK option, but implements the following changes: initial window is extended to 6 segments, the congestion window state variable is increased by 6 segments for each received nonduplicated acknowledgment and the TCP buffers (both at the sender and receiver sides) have a capacity of 320 Kbytes. This protocol solution is identified as TCP-IW6-320K in the remainder of the discussion.

A particular note is represented by the communication of data PDUs between application and transport layer. The FTP-like application encapsulates the information into blocks of fixed size. In correspondence of each block a new TCP connection is opened. This approach has been adopted in order to show how the effectiveness of the protocol specification changes in correspondence to the different block sizes when the channel is affected

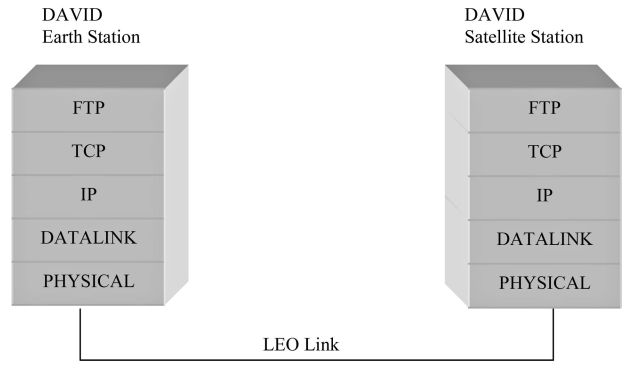

Fig. 2. TCP-based protocol stack architecture.

by errors. In more detail, for each performed test, a number of blocks ranging from 1 to 420, carrying from 306.6 Mbytes to 730 Kbytes respectively, has been considered in the tests. The bandwidth available "seen" at the network layer is set to 102.4 Mbit/s (12.8 Mbytes/s). RTT is set to 10 ms.

Considering the protocol stack reported in Fig. 2, it is possible to state a maximum performance threshold due to the overheads imposed by each protocol of the stack.

The effective bandwidth "seen" by the application layer (C_{appl}) is a portion of the overall bandwidth available at the network layer, as shown in (1). The reduction is due to the overheads introduced by the layers acting below the application

$$C_{\text{appl}} = \left(1 - \frac{L_{\text{TL}} + L_{\text{NL}}}{L_{\text{TL}} + L_{\text{NL}} + \text{payload}}\right) \cdot B \qquad (1)$$

where L_{TL} and L_{NL}, set to 20 bytes, are the overheads introduced at the transport and network layers, respectively. The payload is the information length at the application layer and it is set to 1460 bytes. The bandwidth of the network layer is B . Substituting the numerical values: $C_{\text{appl}} = 0.97\bar{3} \cdot 12.8$ Mbytes = 12.46 Mbytes. The value $(1 - (L_{\text{TL}} + L_{\text{NL}})/(L_{\text{TL}} + L_{\text{NL}} + \text{payload}) = 0.97\bar{3})$ is the maximum normalized throughput. If no further limitations are introduced at the transport layer due to the algorithms, C_{appl} allows a minimum file transfer time of 306.6 Mbytes/12.46 Mbytes/s = 24.61 s.

The maximum bandwidth pipe available at the transport layer may be computed similarly:

$$C_{\text{tr}} = \left(1 - \frac{L_{\text{NL}}}{L_{\text{NL}} + \text{payload}_{\text{tr}}}\right) \cdot B \qquad (2)$$

where $\text{payload}_{\text{tr}}$ includes the application payload and the transport header overhead. Numerically, $\text{payload}_{\text{tr}} = 1480$ bytes, $1 - (L_{\text{NL}}/(L_{\text{NL}} + \text{payload}_{\text{tr}})) = 0.98\bar{6}$ and $C_{\text{tr}} = 12.63$ Mbytes/s. This value is further limited by the TCP implementation, as should be clear in the results reported below. It is worth remembering that 20 out of 1480 bytes (1.35%) are dedicated to the TCP header.

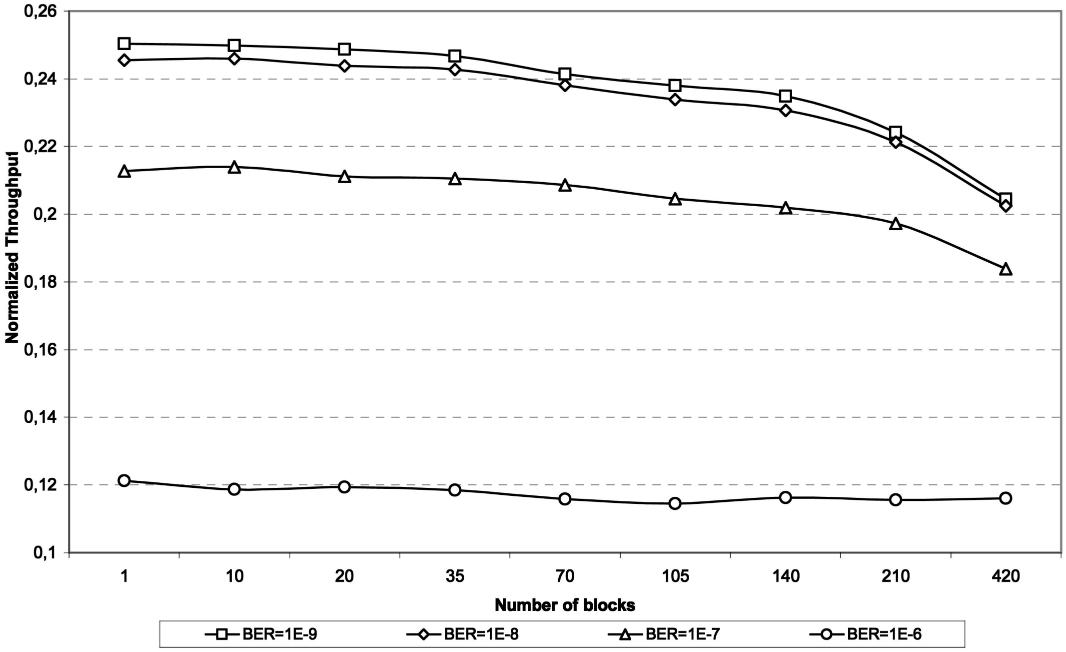

Fig. 3. TCP-IW2-64K normalized throughput.

TABLE I
TCP-IW2-64K Transfer Times(s)

Blocks	BER = 1E-9	BER = 1E-8	BER = 1E-7	BER = 1E-6
420	120.34	121.56	133.86	212.11
210	109.81	111.23	124.73	212.90
140	104.79	106.67	121.86	211.73
105	103.39	105.22	120.28	214.88
70	101.93	103.35	117.92	212.40
35	99.77	101.37	116.88	207.70
20	98.95	100.94	116.52	206.12
10	98.51	100.05	115.03	207.28
1	98.30	100.22	115.67	203.05

TCP-IW2-64K: As highlighted in the previous sections, TCP NewReno is not efficient in satellite networks, mainly because of the high bandwidth-delay product and the error ratio. This expectation is confirmed also in the investigated environment, where the transport protocol is not able to fill the available bandwidth. As shown in Table I and Fig. 3, in correspondence of low values of BER (such as 10^{-9} and 10^{-8}), the performance results experienced are similar to each other and strictly dependent of the number of blocks. As expected, as the number of blocks increases the transfer time grows up because more slow-start phases are invoked. On the contrary, when a small number of blocks is considered (from 1 to 35 blocks), registered values are close to the minimum file transfer reachable by this TCP specification. In fact, there is a performance limitation not only due to the overheads, but also to the algorithms implemented within the TCP: it allows injecting a maximum amount of 32 Kbytes (half of the TCP buffer length, which contains only the application payload) without receiving acknowledgments. It corresponds to 32 Kbytes of data each RTT (10 ms) and to 3.2 Mbytes/s of bandwidth usable by the application. Accordingly, the minimum time required to transfer a file of 306.6 Mbytes allowed by this protocol configuration is equal to 95.81 s, while the normalized throughput (N_thr) is 0.25 at most. It is worth remembering that the normalized throughput is computed by considering the available bandwidth seen by network layer (namely 12.8 Mbytes/s) and the total number of bytes that must be transmitted (namely 306.6 Mbytes). The results in Table I range from 98.3 s (1 block and BER of 10^{-9}) to 101.37 s (35 blocks and BER of 10^{-8}). Consequently, the maximum normalized throughput ranges from 0.24 to 0.23, values close to the maximum of 0.25, but very far from the maximum architectural value of 0.973.

The performance decreases when the channel conditions are more critical, i.e., in correspondence of BER values of 10^{-7} and 10^{-6}. Also in the best cases, the transfer time is about 115 s for BER = 10^{-7} and 203 s for BER = 10^{-6}. This behavior is mainly due to the ineffective recovery mechanism implemented within TCP specification, able to deal successfully with congestion events but causing performance deterioration in presence of link errors. The values of normalized throughput (Fig. 3) range from 0.18 to 0.21 concerning BER = 10^{-7} and from 0.11 to 0.12 for BER = 10^{-6}.

TCP-IW6-320k: When an enhanced specification of TCP is applied, the performance results are more satisfying. In particular, the employment of TCP buffers of capacity grown up to 320 Kbytes

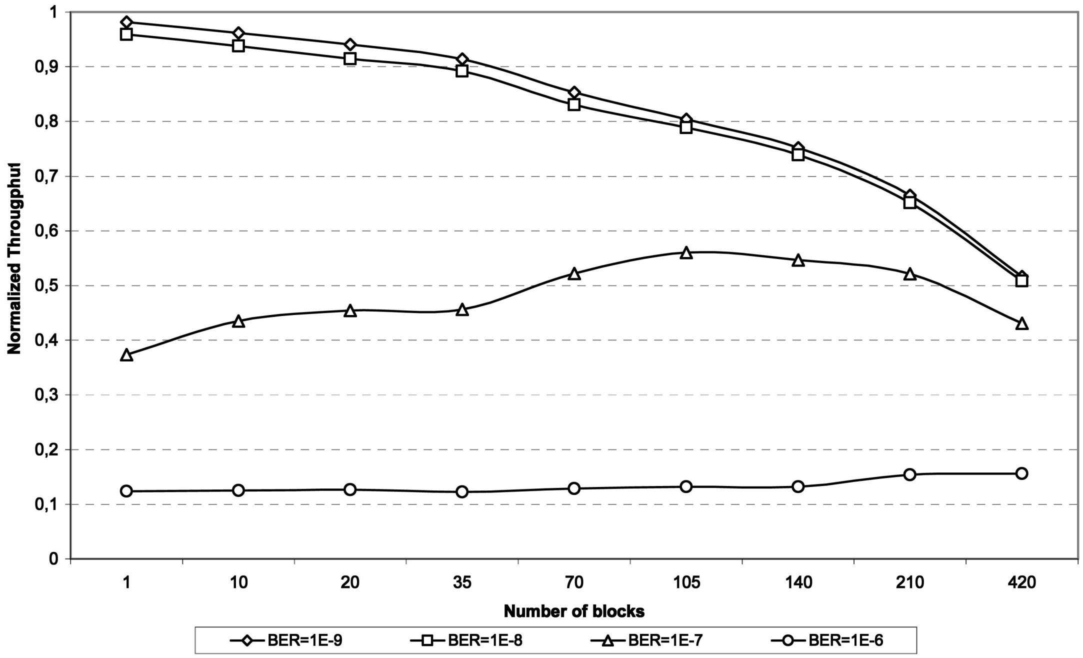

Fig. 4. TCP-IW6-320K normalized throughput.

TABLE II
TCP-IW6-320K Transfer Times(s)

Blocks	BER = 1E-9	BER = 1E-8	BER = 1E-7	BER = 1E-6
420	47.63	48.41	57.10	157.61
210	37.00	37.81	47.21	159.57
140	32.73	33.31	45.06	186.36
105	30.61	31.21	43.92	186.49
70	28.86	29.64	47.15	191.16
35	26.94	27.60	53.96	200.93
20	26.17	26.92	54.18	194.07
10	25.58	26.25	56.54	197.07
1	25.07	25.67	65.94	199.07

relaxes the bandwidth constraints imposed by the transport layer and allows almost the full utilization of the channel bandwidth. In practice, TCP can inject 160 Kbytes of data each RTT (10 ms). The corresponding bandwidth limit would be 16 Mbytes/s, which is above the effective available bandwidth at the network layer (which is equal to 12.46 Mbytes/s). It means that, at least concerning the buffer length, TCP does not represent a bottleneck. It is shown in Fig. 4 that the maximum values of normalized throughput, registered in the case of 1 block transmission, are equal to 0.96 and 0.93 in the presence of BER values of 10^{-9} and 10^{-8}. The throughput values correspond to transfer times, shown in Table II, of 25.07 s and 25.67 s, both very close to the maximum achievable on the channel, equal to 24.61 s.

A particular case is represented by the test performed with BER equal to 10^{-7}: in this scenario the best performance values are provided by transmitting 105 blocks. This behavior happens because TCP-IW6-320K implements a more aggressive transmission algorithm than TCP-IW2-64K; as a consequence, when a small number of blocks is employed, several packets are likely to be lost within each open connection, triggering repeated fast retransmit/ fast recovery phases that imply the reduction of the congestion window. On the other hand, when a higher number of blocks is used, the impact of error ratio is lower but the increased number of entered slow-start phases causes waste of the channel bandwidth; the adoption of an intermediate number of blocks, equal to 105, allows better performance: a transfer time of 43.92 s and a normalized throughput of about 0.55.

When the channel conditions are more critical (BER equal to 10^{-6}), following what has been stated before, it is expected to find better performance when many blocks are employed, because a lower number of lost packets is registered within each connection. As depicted in Table II and Fig. 4, the best performance with BER = 10^{-6} is obtained in correspondence of 420 blocks: 157.61 s and 0.15 concerning the transfer time and normalized throughput, respectively. Is worth noting that, when the channel is strongly affected by errors (BER of 10^{-6}), the benefit of an aggressive slow-start phase and increased capacity of TCP buffer is no longer exploitable: TCP-IW2-64K and TCP-IW6-320K offer similar performance. Actually an increased buffer length helps fill the bandwidth pipe but does not allow distinguishing channel and congestion errors, which is the main cause of the performance degradation in this case.

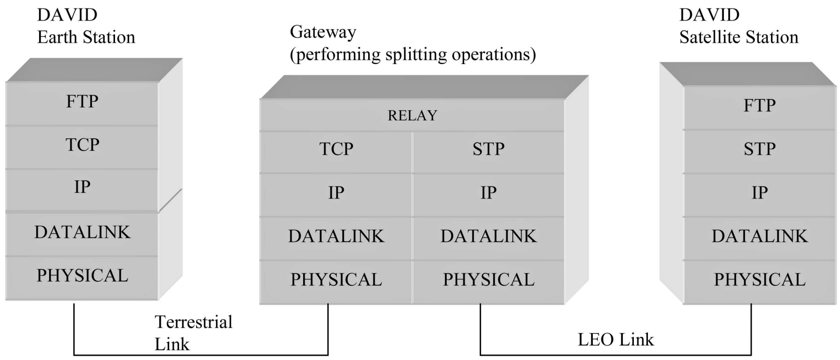

Fig. 5. Splitting architecture.

C. Splitting Architecture

A possible approach aimed at overcoming the impairments introduced by satellite links and improving the data communication performance with respect to TCP employment, is represented by a connection-splitting approach [10], shown in Fig. 5. The basic idea of this architecture is to isolate the satellite portion from the terrestrial access link (by using an intermediate agent called gateway), in order to manage the connection from the DAVID Earth station to the intermediate agent and the connection from the intermediate agent to the DAVID satellite, separately at the transport layer. In more detail, the aforementioned intermediate agent is a specialized gateway, responsible of splitting the transport layer connection into two parts and managing the communication between the two portions. A specific protocol stack tuned on the satellite characteristics is implemented on the air portion. The solution proposed at the transport layer for the satellite is called STP (satellite transport protocol): it derives from TCP NewReno with SACK, but presents meaningful differences. The slow-start phase is excluded and the transmission is ruled by maintaining a constant congestion window value, equal to the bandwidth-delay product, in order to fill the bandwidth pipe offered by the satellite channel completely and without delay due to slow start; TCP buffers are tuned accordingly. Concerning the recovery mechanism, the retransmission phase is triggered when a duplicated acknowledgment is received; after the recovery has completed, the congestion window assumes its initial value, set, as previously indicated, to the bandwidth-delay product. In the case study presented here, the capacity of TCP buffer is fixed to 320 Kbytes and the congestion window to 86 segments. This solution is identified as STP-IW86-320K.

Also in this case, in order to evaluate the importance of the information unit length, the file to be transferred has been split into blocks whose dimension is fixed within each test. The performance analysis has been accomplished considering a number of blocks ranging from 1 to 420 and carrying from 306.6 Mbytes to 730 Kbytes each, respectively.

TABLE III
STP-IW86-320K Transfer Times(s)

Blocks	BER = 1E-9	BER = 1E-8	BER = 1E-7	BER = 1E-6
420	28.50	28.62	29.58	40.29
210	26.36	26.55	28.20	38.10
140	26.36	26.50	27.91	38.79
105	25.84	26.01	27.58	38.56
70	25.67	25.79	27.28	38.44
35	25.31	25.48	27.05	37.42
20	25.17	25.34	27.00	37.40
10	25.11	25.30	26.92	37.21
1	25.04	25.21	26.92	36.79

STP-IW86-320K: As expected, the proposed architecture offers significant and satisfying performance results in terms of transfer time (reported in Table III) and, accordingly, normalized throughput values (in Fig. 6). This behavior is due to the splitting connection approach that includes the specification of the satellite transport protocol, which allows the full exploitation of the network resources for BER values ranging from 10^{-9} to 10^{-7}. In more detail, considering the best performance results registered (with 1 block transmission), the transfer time passes from 25.04 s (BER = 10^{-9}) to 25.21 s (BER = 10^{-8}), and raises up to 26.92 s in case of BER = 10^{-7}. The fluctuations of the performance are limited and close to the minimum transfer time reachable with a TCP-like architecture (24.61 s). Accordingly, normalized throughput values range from 0.96 to 0.89, as reported in Fig. 6.

When the BER is 10^{-6}, the system performance degrades because of the high number of lost packets that cause long periods of retransmission, wasting the available bandwidth also with this approach. The minimum transfer time, measured with 1 block, is 36.79 s. It corresponds to a normalized throughput of 0.65. It is straightforward that the experienced behavior is much more satisfying compared with TCP-IW2-64K and TCP-IW6-320K. The improvement is due to the modified algorithms in the transport protocol, which is adapted to the satellite channel. It

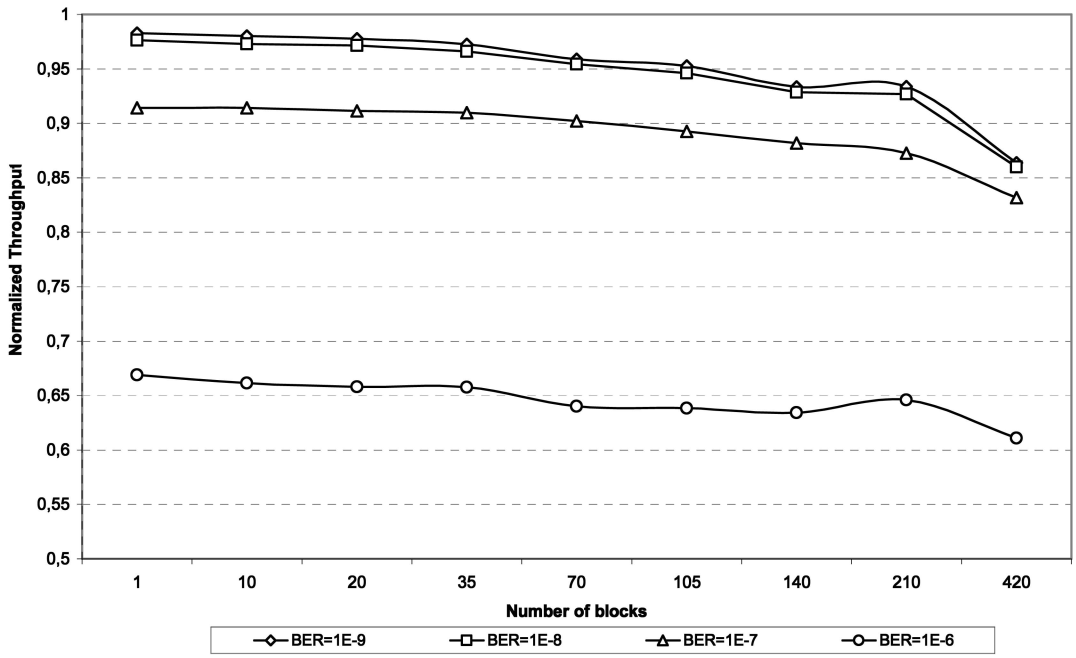

Fig. 6. STP-IW86-320K normalized throughput.

is important to highlight that, due to the new schemes, the effect of the fragmentation into blocks is less evident.

D. CFDP-Based Solution

An alternative approach consists in adopting the CCSDS-based protocol stack [37] directly on the DAVID Earth station and on the satellite platform, without inserting any intermediate agent in the middle. The basic idea is to exploit the main functionalities of protocols standardized by CCSDS, which have been intensively employed during space missions. More detail is required about the protocol stack implemented on every device. It is shown in Fig. 7. At the datalink layer, the CCSDS proximity-1 space link protocol [30] is adopted in order to use its capabilities of making reliable data communication over mobile satellite networks characterized by short propagation delays. Proximity-1 space link protocol employs convolutional punctured turbo-codes, which have a high correction capacity, and allows getting BER values lower than 10^{-5}. In particular, the case of Reed Solomon RS (223,255) codes is investigated, since it permits high recovery over very noisy channels without excessive overhead.

At the network layer, the space packet protocol is employed in order to guarantee the functionalities of addressing and routing, usually performed by the IP in the TCP/IP protocol suite.

CFDP [29] is adopted at the highest layer. As introduced in Section III, CFDP reliable NAK immediate mode is applied. It allows getting a reliable communication through a NAK scheme, which uses

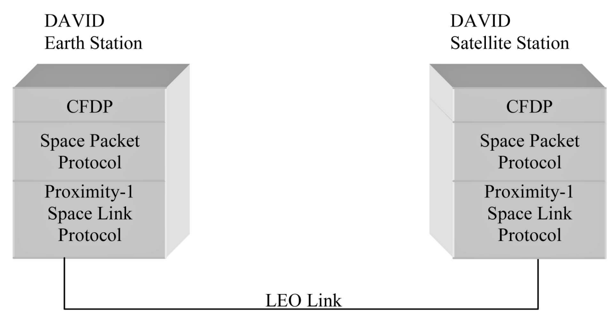

Fig. 7. CFDP-based solution architecture.

NAK PDUs. They carry information about lost blocks of data and are issued by CFDP when an out of order data block is received. The maximum number of retransmissions for each data block is set to 15.

It is worth noting that, also in this case, the upper layer segments the information into blocks, whose size, fixed at the beginning of the transaction, affects the performance. Even if blocks larger than 64 Kbytes are not allowed in the CFDP recommendation in order to avoid an excessive number of retransmitted bytes when the channel conditions are very critical, block sizes ranging from 7300 bytes to 730 Kbytes have been considered in this work for the sake of completeness. The number of blocks range from 42000 to 420. The performance degradation in presence of high BER values (i.e., BER = 10^{-7}, 10^{-6}) and large blocks will be evident.

This approach is referred as "CFDP reliable NAK immediate" in the reminder of this work. As in the TCP/IP suite, it is possible to state a maximum performance threshold imposed by the overheads. The bandwidth available (12.8 Mbytes/s) is computed

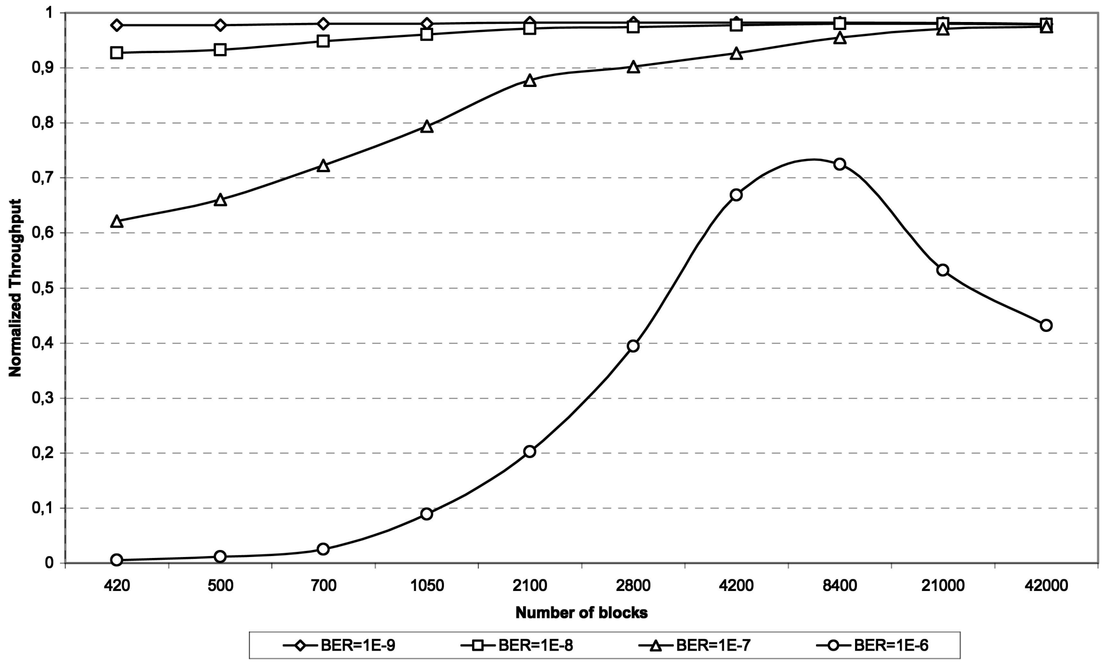

Fig. 8. CFDP reliable NAK immediate normalized throughput.

TABLE IV
CFDP Reliable NAK Immediate Transfer Times(s)

Blocks	BER = 1E-9	BER = 1E-8	BER = 1E-7	BER = 1E-6
420	25.17	26.54	39.62	4394.23
500	25.17	26.38	37.25	2192.18
700	25.11	25.95	34.05	975.57
1050	25.10	25.62	30.98	277.38
2100	25.05	25.34	28.04	121.30
2800	25.05	25.26	27.28	62.42
4200	25.05	25.17	26.55	36.78
8400	25.05	25.11	25.77	33.95
21000	25.07	25.10	25.35	46.21
42000	25.12	25.13	25.25	56.97

at the space packet protocol. The overhead of the CFDP is set to 20 bytes as well as the space packet protocol's overhead. The minimum file transfer time reachable by this architecture is 24.61 s, as well as in the TCP/IP suite.

CFDP reliable NAK immediate: The solution is really tailored for space data communication and offers interesting performance results, as indicated in Table IV, concerning the transfer time, and in Fig. 8, sketching the normalized throughput values. In particular, in the presence of BER values 10^{-9}, 10^{-8} and 10^{-7}, the registered values of transfer time are very close to the minimum value of 24.61 s.

It is interesting to show that, when the channel behavior is almost ideal, the number of blocks, in which the information is segmented, has no practical effect on the performance; the normalized throughput is always above 0.95. If BER = 10^{-8}, a too large block dimension penalizes the transfer because the time required for the retransmission is too long; the best performance is obtained for blocks of small size. This behavior is more evident for BER = 10^{-7}. When BER = 10^{-6}, the drawbacks introduced by blocks of big dimension are outstanding, but, on the other hand, the values reported in Table IV, allow observing that, when the information is very fragmented (21000 and 42000 blocks) and there is a big number of retransmissions, the overhead introduced by a large number of blocks affects the overall performance. A proper balance between the time required for retransmissions and the overhead introduced helps improve the performance. In the test performed, 8400 blocks allow obtaining the minimum file transfer (33.95 s). It is worth noting that 33.95 s is a very satisfying result corresponding to a normalized throughput of 0.71, which is excellent in the channel conditions giving origin to the shown results.

E. Comparison of the Results

This subsection contains the comparison of the protocol and architecture solutions investigated for the LEO portion.: TCP-IW2-64K and TCP-IW6-320K, and the two TCP-based solutions, STP-IW86-320K (the STP adopted in the splitting architecture), and "CFDP reliable NAK immediate," (which applies the CFDP-based solution described above).

Fig. 9 shows the normalized throughput of the different configurations by varying the BER value. For each configuration and each BER value, the best obtained result (corresponding to a specific number of blocks used) is selected and shown. If the BER values are 10^{-9} and 10^{-8}, the number of packets lost during the transfer is very limited and all the configurations,

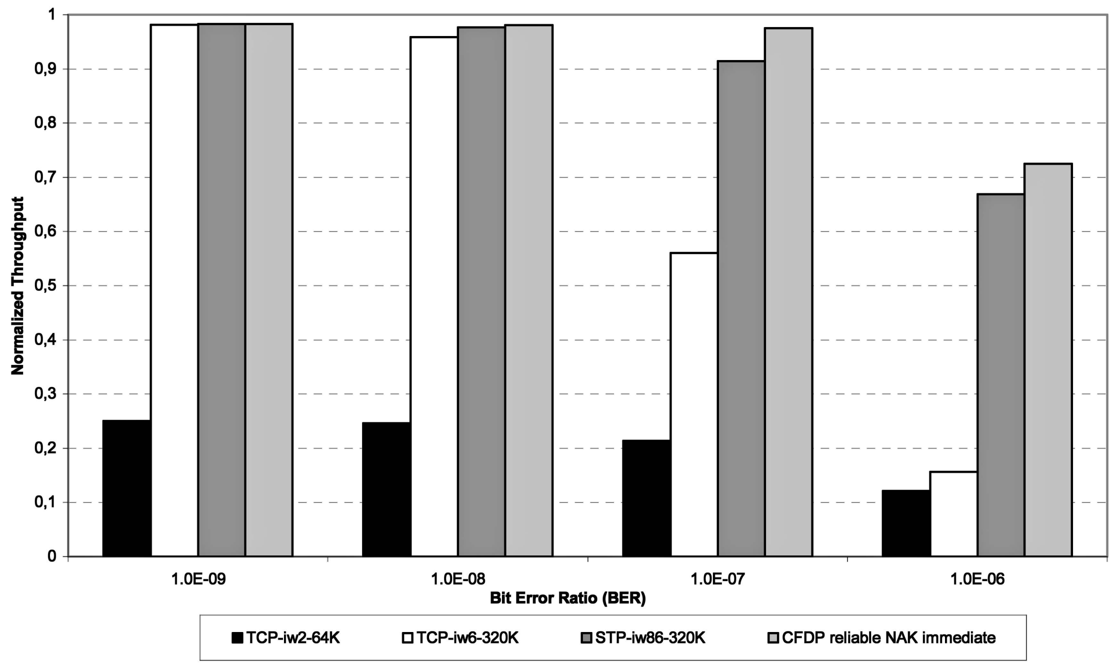

Fig. 9. Comparison of results.

except for TCP-IW2-64K, provide satisfying results. The normalized throughput ranges from 0.93 to 0.96.

TCP-IW2-64K cannot use the available bandwidth completely because of the limited TCP buffer length and offers a maximum normalized throughput close to 0.24. It is worth noting that even if TCP is not really tailored for data transmission in space missions, the specification TCP-IW6-320K can offer satisfying results, if the channel is not errored, thanks to the increased TCP buffer size that helps fill the available bandwidth.

When the impairments introduced by the satellite channel are more severe (namely BER of 10^{-7}), the TCP performance is heavily affected by the implemented recovery algorithm. The normalized throughput is 0.21 for TCP-IW2-64K and 0.55 for TCP-IW6-320K. The throughput raises up to 0.89 for STP-IW86-320K where a novel recovery algorithm, but always acknowledgement-based, is used. The best result is obtained for CFDP reliable NAK immediate (0.95). It allows concluding that a NAK mechanism is more suited for space missions than acknowledgement-based schemes; it can use the bandwidth available better by selecting exactly which information needs to be retransmitted. In particular, when the channel conditions are very critical (BER = 10^{-6}), the simple extension of the TCP buffer length is not effective because the low performance is essentially due to the error recovery scheme. In fact TCP-IW2-64K shows a throughput of 0.12 and TCP-IW6-320K of 0.15. The different recovery algorithm used by STP-IW86-320K allows reaching a throughput of 0.65. The NAK scheme used in CFDP reliable NAK immediate permits a further performance improvement that raises the throughput up to 0.71.

The discussion indicates that STP (STP-IW86-320K) and “CFDP reliable NAK immediate” are the more promising solutions in the various tests performed. From the implementation point of view, STP-IW86-320K has been designed to mitigate TCP drawbacks and its employment requires the introduction of extra complexity into the network in terms of specialized gateways responsible for managing the split connections on both the terrestrial and the satellite sides. “CFDP reliable NAK immediate” offers better performance through the adoption of a recovery mechanism based on a NAK scheme. Operatively, it implies that a full CCSDS-based protocol stack is adopted on the terminal station (in our case the DAVID Earth station and the DAVID satellite station); moreover, a crucial element determining the communication performance concerns the choice of the block size, which may seriously affect the effectiveness of the recovery phase [31].

V. GEO LINKS PERFORMANCE ANALYSIS

A. Overview

The second part of the investigation is dedicated to the data communication performed from the DAVID satellite to the end terminal, exploiting an inter-satellite link including the geostationary satellite ARTEMIS [2]. The considered environment is shown in Fig. 10. The GEO link offers an available bandwidth of 256 Kbytes/s (measured at the space

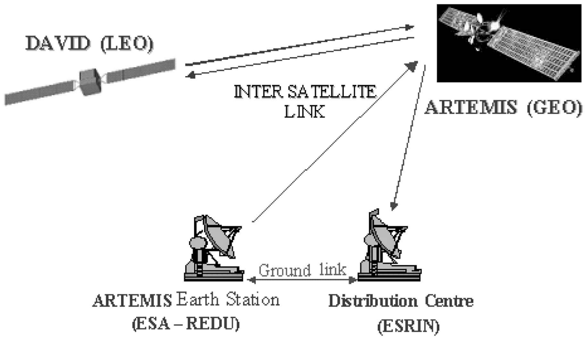

Fig. 10. LEO to end terminal path.

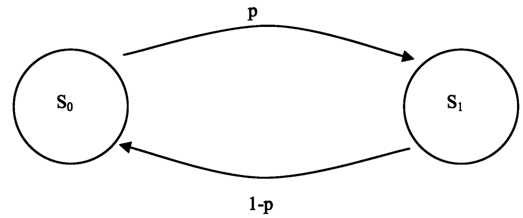

Fig. 11. Two-state channel model.

packet protocol in the CCSDS architecture) in the Ka band. The propagation delay is assumed 0.250 s. The main aspect of this scenario is that the return link from the distribution center to the GEO satellite (see Fig. 10) [3], necessary for the employment of acknowledgment-based schemes is not always available. For this motivation, TCP/IP-based architectures are not very applicable in this environment and are not considered in this work.

The following main elements characterize the considered environment.

1) As in most satellite links built in the Ka band, the communication may be strongly affected by fading due to meteorological events such as rain and storms, which determine rough BER values ranging from 10^{-2} to 10^{-4}. In this case, typical countermeasures include FEC schemes and interleaving mechanisms, which introduce a robust protection against bit corruption events and allow assuming a reduction of the BER seen by the upper layer. As done for the LEO portion, the BER considered in the remainder of this work is the value evaluated after performing such operations.

2) The communication may be affected by the partial unavailability of a return channel that is necessary to transport feedback information.

This characterization may be efficiently considered by using a two-state channel model, shown in Fig. 11, where:

1) States S_0 and S_1 describe the unavailability and the availability of the return channel, respectively.

2) The transition from a state to another is ruled by a random process; the probability to go from state S_0 to state S_1 is indicated as p, considered constant within each performed test. The whole investigation has considered values of p ranging from 10^{-1} to 10^{-3}.

3) Each state is associated to the same BER value, fixed within each simulation. Assumed BER values range from 10^{-9} to 10^{-6}.

It is straightforward that such characterization of the transmission link constitutes a very strong limitation to the application of TCP-based schemes, in which a continuously available return link for the transportation of the acknowledgments is strictly necessary. A protocol solution that does not require feedback (or continuous feedback) is more suitable for the data transmission in such scenario. In the same time, it is also important to assure reliability to the data communication. Actually, it is possible to use both heuristics based on the repeated transmission of blocks of data and solutions that exploit the temporary availability of the return channel.

The CCSDS-based protocol stack (shown in Fig. 12) seems to be proper for the environment. The CFDP protocol is adopted at the highest layer. The following CFDP configurations have been considered in the tests in order to have a complete analysis of the data communication over the GEO link.

1) CFDP reliable NAK asynchronous. CFDP is configured in reliable mode and uses an asynchronous NAK scheme [29], in order to exploit the return link when it is available. The action of sending NAK is not immediately performed when an out of order block of data is received but it is triggered by asynchronous events as well as channel availability periods. Each single block may be retransmitted for a maximum of 15 times. After that the communication is considered expired and the transfer not completed.

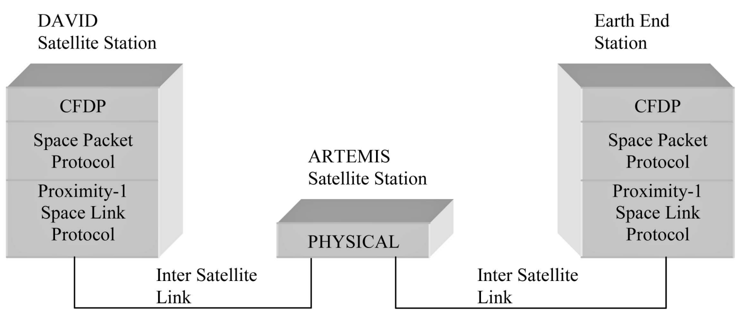

Fig. 12. CCSDS-based protocol architecture.

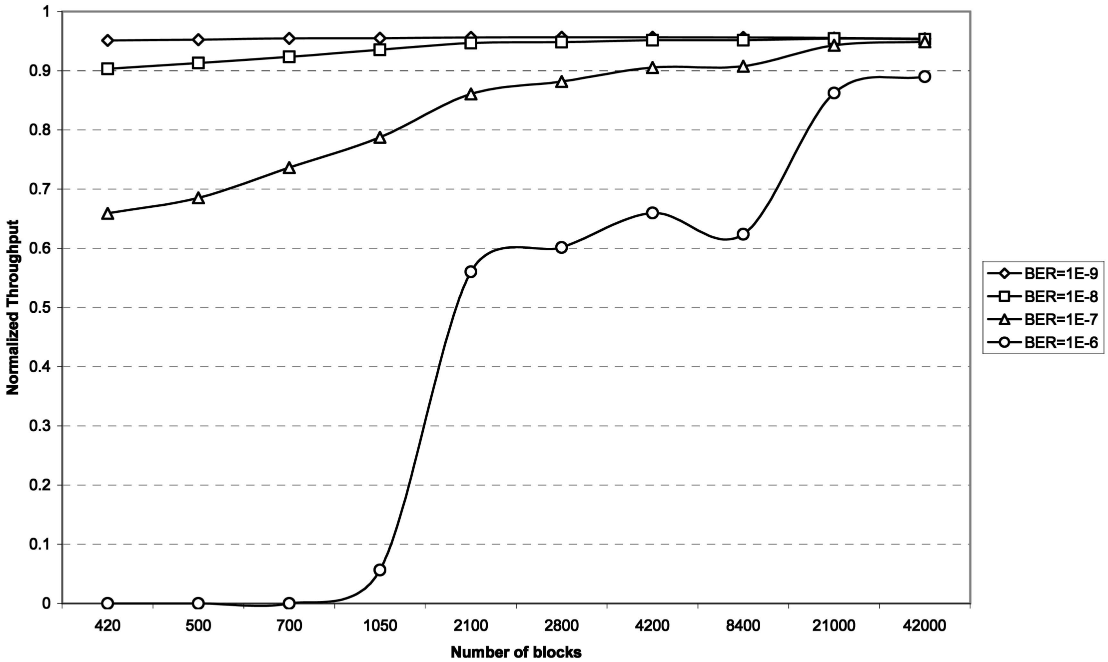

Fig. 13. Normalized throughput for $p = 0.1$.

2) CFDP-repeat. CFDP is configured in unreliable mode [29] but the reliability of the communication is assured using heuristics based on repeated transmissions introduced by the authors: each block is transmitted N times consecutively, where N has been fixed within each test. In this case the "ratio of success," indicated as S_ratio and defined as the ratio between the blocks successfully received and the total amount of blocks transmitted, has been introduced as metric.

The analysis of the performance has been accomplished by considering a transfer of a 306.6 Mbytes file (measured at the CFDP layer). The normalized throughput N_thr and the overall transfer time, as in the case of LEO link investigation, are the metrics. The normalized throughput is evaluated as the ratio between the final throughput F_thr experienced by the data communication and the available bandwidth B: $N\_thr = F\_thr/B$.

B. Evaluation of Protocol Solutions

CFDP reliable NAK asynchronous: The negative effect of noncontinuous availability of the return link is partially mitigated by configuring CFDP protocol in reliable mode operating with the asynchronous NAK scheme because, with this algorithm, it is possible to issue NAK notifications only when the return link is available.

The transmission has been ruled by sending blocks of different size, ranging from 7300 bytes to 730 Kbytes corresponding to a total number of blocks from 42000 to 420, respectively. As highlighted in the investigation about the LEO links, even in this case, the choice of the block size is important, since the impact of the BER on the data communication is strictly dependent on it. Moreover, as introduced above, an important element affecting the whole performance is the probability of return link availability p, which is assumed to vary from 10^{-1} to 10^{-3}.

Fig. 13 contains the normalized throughput when $p = 0.1$, for different BER values: the proposed protocol solution "CFDP reliable NAK asynchronous" assures the almost full occupancy of the channel when BER values range from 10^{-9} to 10^{-7}. The corresponding maximum normalized throughput value varies from 0.956 to 0.949. The impact of the block size on the performance is topical: raising up the BER value, the number of blocks required to achieve the best performance decreases accordingly. This behavior is due to the fact that, in presence of low BERs, the performance is influenced by the overhead, which is lower when the number of blocks to be retransmitted is smaller. On the other hand, when the BER value is higher, the dominant effect is represented by the number of lost blocks and the probability to receive a corrupted block is lower when small blocks are transmitted. In fact: when BER = 10^{-9} the best performance has been measured for 4200 blocks, but the effect of the block size is not evident. If BER = 10^{-8}, 21000 blocks guarantee the highest throughput; the importance of the block size is clearer: the throughput value ranges from 0.90, measured when 420 blocks are transmitted, to 0.954, obtained with 21000 blocks. The performance gap in dependence of the number of blocks is outstanding

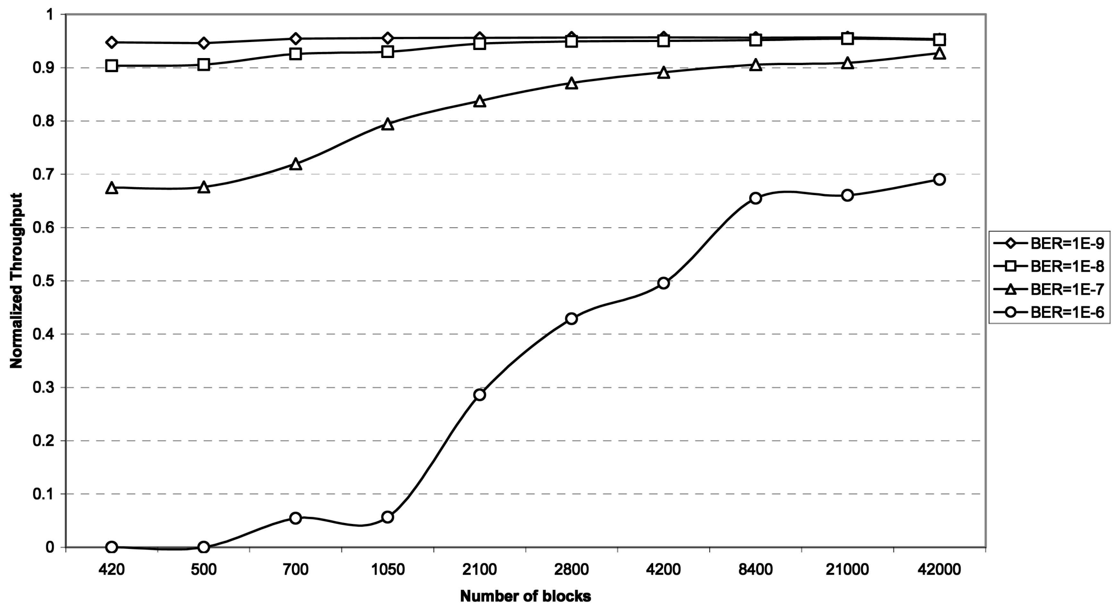

Fig. 14. Normalized throughput for $p = 0.01$.

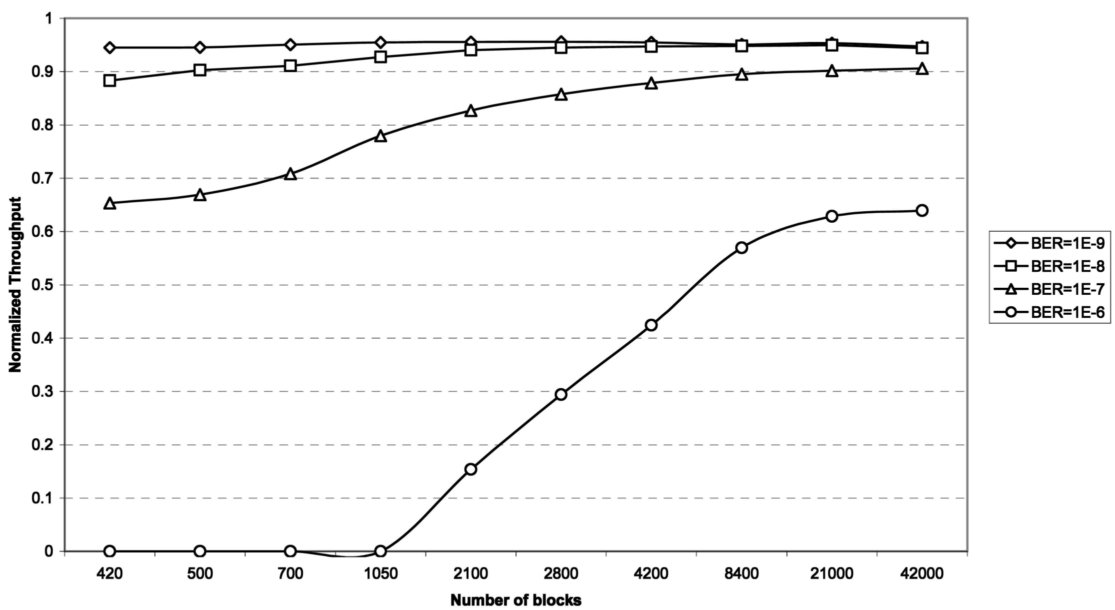

Fig. 15. Normalized throughput for $p = 0.001$.

when BER is equal to 10^{-7} and 10^{-6}. When BER = 10^{-7}, if the file transfer is performed with only 420 blocks, the obtained normalized throughput is 0.67. It lifts up when the number of blocks, in which the information is structured, is increased and reaches 0.927 with 42000 blocks. The same behavior is measured for BER = 10^{-6} but, in this case, the difference is between service guaranteed and nonguaranteed; if the number of blocks is below 700, it is not possible to complete the file transfer. The maximum throughput (0.89) is measured with 42000 blocks.

A particular observation arises from the fact that, even if the return link is not continuously available, satisfying performance results are offered by "CFDP reliable NAK asynchronous" solution with a proper choice of the blocks' length. This behavior can be explained considering that a probability of link availability of 0.1 is enough to assure retransmissions without excessive delay.

When the probability of availability of the return link reduces to 0.01, as shown in Fig. 14, the behavior is similar to the previous case but the performance is accordingly affected, mainly when the channel conditions are very critical (BER = 10^{-6}). In this case, the best performance in terms of the normalized throughput is 0.69 and it is measured when the transmission is performed using 42000 blocks. In the cases of lower BER values ($10^{-9} \div 10^{-7}$), the performance results are still acceptable: the maximum

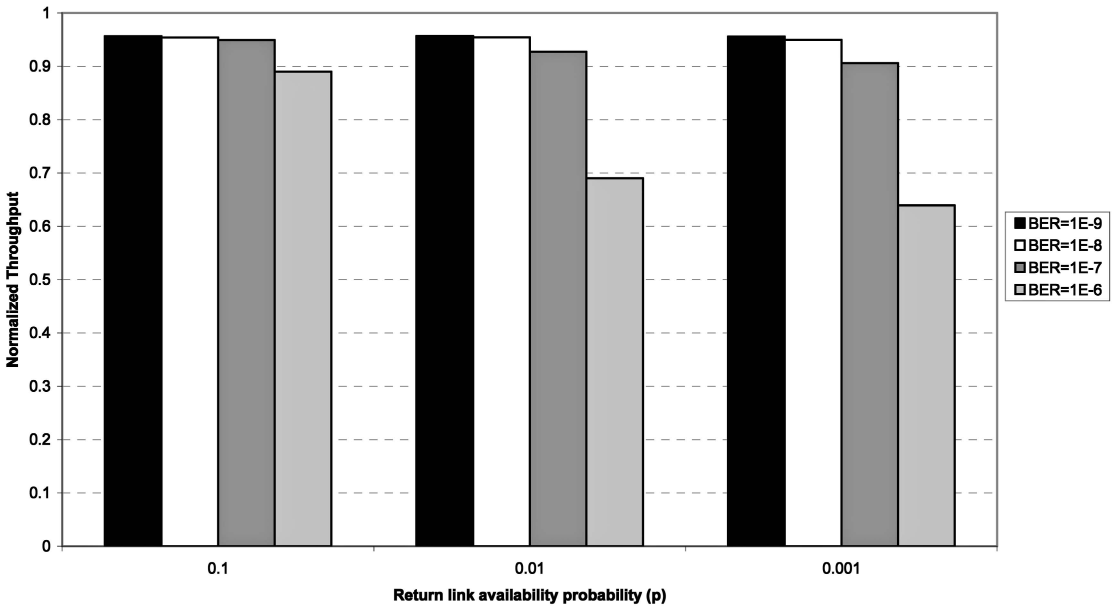

Fig. 16. Normalized throughput for investigated p values.

normalized throughput values range from 0.957 to 0.927.

Concerning the case of probability of link availability equal to 0.001 (Fig. 15), each retransmission is performed with a large delay, because the limited availability of the return link introduces an extra latency in the issuance of NAK PDUs. The performance slightly degrades also in correspondence of low BER values ($10^{-9} \div 10^{-7}$) that impose a normalized throughput ranging from 0.96 to 0.91 in the best configurations. When the BER value is 10^{-6}, the maximum normalized throughput value is 0.64.

The role of the blocks' dimension may be commented similarly to the previous cases, even if the importance of a correct choice is stressed.

The results are summarized in Fig. 16, where the normalized throughput versus the probability (p) of return link availability is shown, by choosing for each value of BER and p, the block dimension providing the best performance. The channel bandwidth utilization is excellent when BER values range from 10^{-9} to 10^{-7} independently of the limited channel availability on the return link. When the channel experiences BER of 10^{-6}, the utilization is very high if $p = 0.1$ and decreases when $p = 0.01$ (normalized throughput 0.69) and $p = 0.001$ (normalized throughput 0.64). Even if the absolute values of 0.69 and 0.64 can seem low for bandwidth utilization, it is also true that, with BER = 10^{-7}, BER = 10^{-6} and a return link available only for 1% and 0.1% of the overall time, a completely reliable communication has been completed in reasonable time (approximately 1735 s and 1873 s) and 69% and that 64% of the overall bandwidth has been used. It is a completely satisfying result.

CFDP-repeat: An alternative approach, aimed at mitigating the performance degradations due to the limited availability of the return link, is to adopt a heuristics based on repeated transmission. The basic idea is to transmit the same block for N times consecutively, in order to ensure that the data block is likely to be received correctly.

Concerning the protocol stack, as highlighted in the previous section, full CCSDS-based structure is considered again, and the transfer of data between the DAVID satellite station and the end terminals is performed by means of the CFDP protocol, as indicated in Fig. 12. CCSDS file delivery protocol is configured in unreliable mode in this case; its functionalities have been extended by the authors in order to manage repeated transmissions, as required by the heuristics adopted.

Knowledge about the presence of a return link for information feedback is no longer necessary, since all the operations are performed on the downlink channel. The number of transmissions employed and the size of each transmitted block plays an important role. The former has a double impact: raising up the number of transmissions increases the probability that the blocks are delivered correctly but implies some channel bandwidth wastage. The latter strongly affects the performance in terms of percentage of corrupted blocks, since the probability of receiving blocks correctly strictly depends on the BER value and on the block size itself, as already shown before.

Two metrics have been adopted in order to evaluate the performance in this approach: the ratio of success registered during the data transfer and the normalized throughput, which is defined as in the previous cases but it is weighted by the ratio of

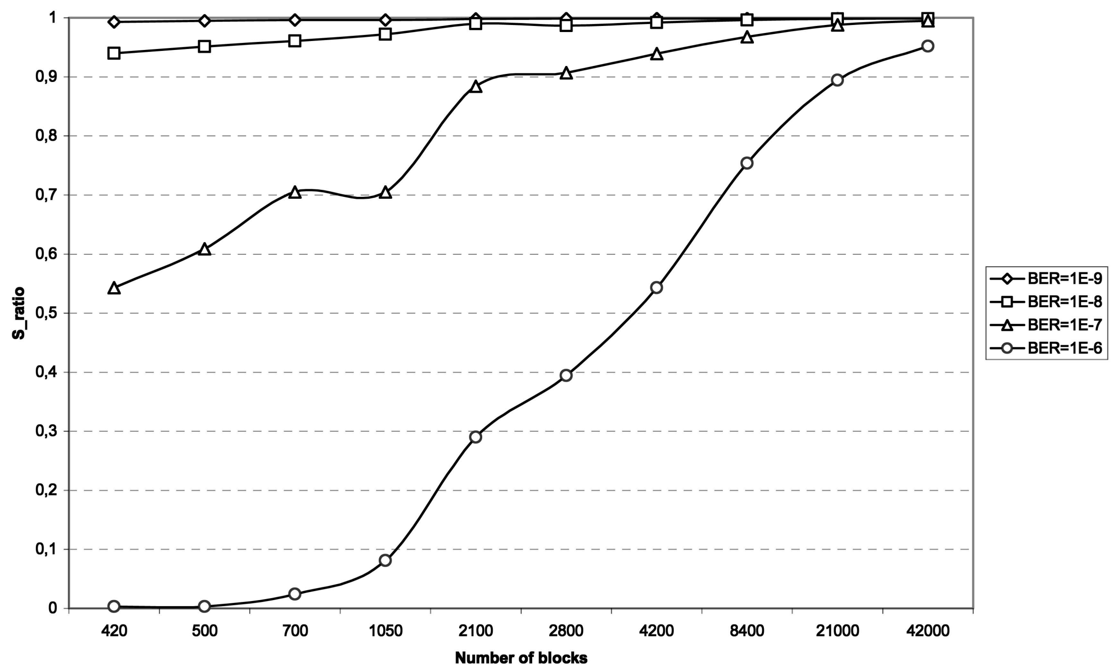

Fig. 17. Ratio of success versus number of blocks for 1 transmission.

success *S_ratio* (defined in (3)), where transfer size and transfer time are the amount of data exchanged during the communication and the time required by the transaction itself, respectively. Each single block (both received and transmitted) is counted only once. The quantity *S_ratio* defines the percentage of file that actually arrives at the destination. It is very meaningful because, for some applications (i.e., video-streaming), it is not necessary to get the reception of 100% of the sent information.

$$S\_ratio = \frac{\text{number of blocks received correctly}}{\text{number of blocks transmitted}}$$
$$N\_thr = \frac{\text{transfer size}}{\text{transfer time}} \cdot \frac{1}{\text{available bandwidth}} \cdot S\_ratio. \quad (3)$$

The investigation has been carried on for BER values ranging from 10^{-6} up to 10^{-9} by varying the number of blocks from 420 to 42000 and the number of times a single block is transmitted (i.e., the number of transmissions).

The number of transmissions ranges from 1 to 15 and the behavior is similar in all the tests: if BER is equal to 10^{-9}, a rate of success very close to 1 may be obtained independently of the number of blocks, but, when the BER value increases, only small blocks together with many transmissions, guarantee a satisfying performance. The cases for 1, 2, 5, and 15 transmissions have been chosen, among the others, to show this behavior. Figs. 17–20 contain the ratio of success (*S_ratio*) versus the number of blocks for 1, 2, 5, and 15 transmissions, respectively. It is interesting to note that, when BER is equal to 10^{-6}, the file may be transmitted almost entirely only by using 15 transmissions and 42000 blocks. A particular attention has to be reserved to the trade-off between block size and number of transmissions. Figs. 17–20 show that, for low BER values ($10^{-9} \div 10^{-8}$), the ratio of success is almost insensitive to both the transmission number variation and the block size; this is due to the fact that such BER values have a lower impact on the performance result in terms of lost block event and, consequently, raising up the number of blocks along with the number of transmissions does not produce meaningful differences. On the other hand, when more severe BER values are concerned ($10^{-7} \div 10^{-6}$), the performance is strongly dependent of the block size; higher the number of blocks more promising are the performance results since loss events are less critic when blocks of small size are applied. Even in this case, the role played by the number of transmissions is less evident, since the pure repetition of data blocks does not assure an improvement of the rate of success in presence of high BER values.

On the other hand, increasing the number of transmissions, independently of the state of the transmission because there is no knowledge about it, delays the completion of the overall operation automatically. The overall time necessary to complete the file transmission increases linearly with the number of transmissions. The value slightly varies depending on the number of blocks. It ranges from 1250.72 s with 420 blocks to 1255.63 s with 42000 when only 1 attempt to send the file is used; it raises up to 18760 s (420 blocks) and 18834 s (42000 blocks) with 15 transmissions.

The lack of efficiency is measured also by the value *N_thr*. Figs. 21–24 contain this measure versus the number of blocks for 1, 2, 5, and 15

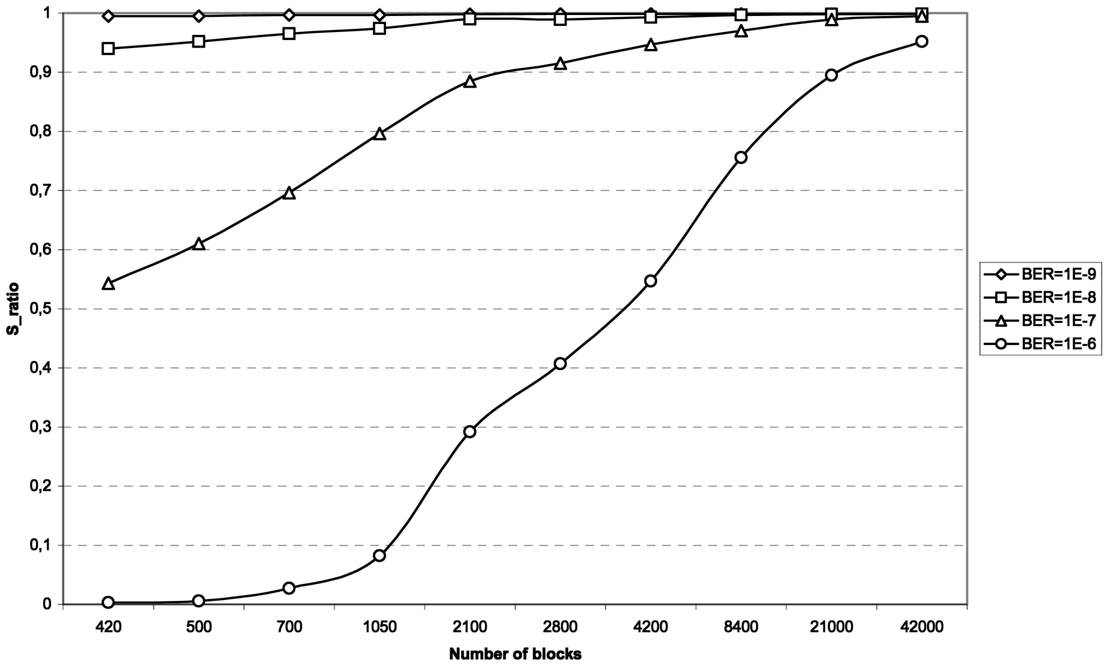

Fig. 18. Ratio of success versus number of blocks for 2 transmissions.

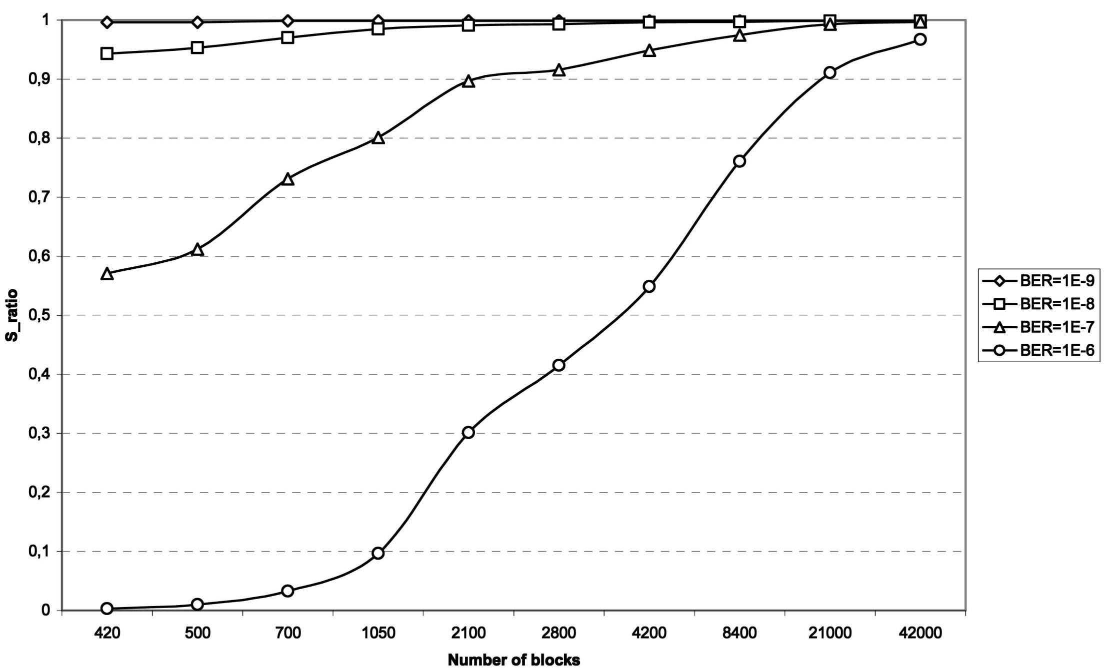

Fig. 19. Ratio of success versus number of blocks for 5 transmissions.

transmissions, as done for Figs. 17–20, respectively. If, on one hand, the repetitions assure a high degree of accuracy also in the presence of channel degradation, on the other hand, the wastage of time introduced by them implies a decrease of the bandwidth utilization. In particular, when 1 transmission is applied, the performance results, as expected, are quite satisfactory and range from 90.8% (BER = 10^{-6} and 42000 blocks) to 95.6% (BER = 10^{-9} and 2800 blocks). On the other hand, the impact of the increased repetitions barely affects the performance results, because of the high delay required by the transmission of the file. In more detail, also with a proper choice of the block length, the efficiency ranges from 45.1% (BER = 10^{-6} and 42000 blocks) to 47.8% (BER = 10^{-9} and 2800 blocks), if 2 transmissions are applied, and from 18.4% (BER = 10^{-6} and 42000 blocks) to 19% (BER = 10^{-9} and 700 blocks), with 5 transmissions. When 15 transmissions are tested, the efficiency ranges from 61.8% (BER = 10^{-6} and 42000 blocks) to 63.8% (BER = 10^{-9} and 420 blocks).

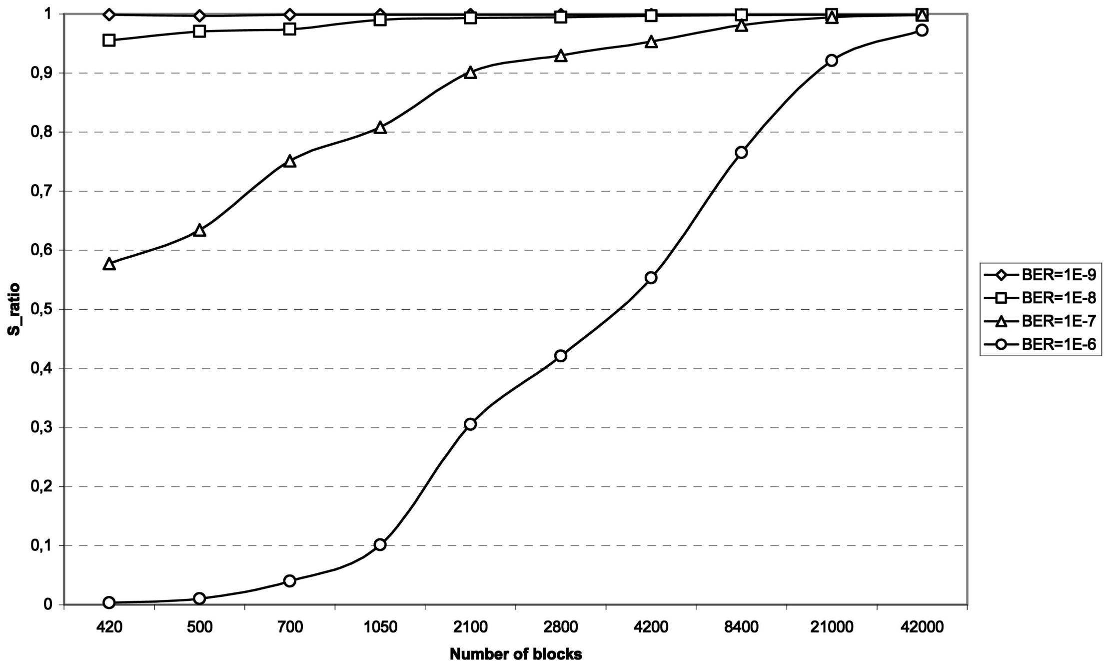

Fig. 20. Ratio of success versus number of blocks for 15 transmissions.

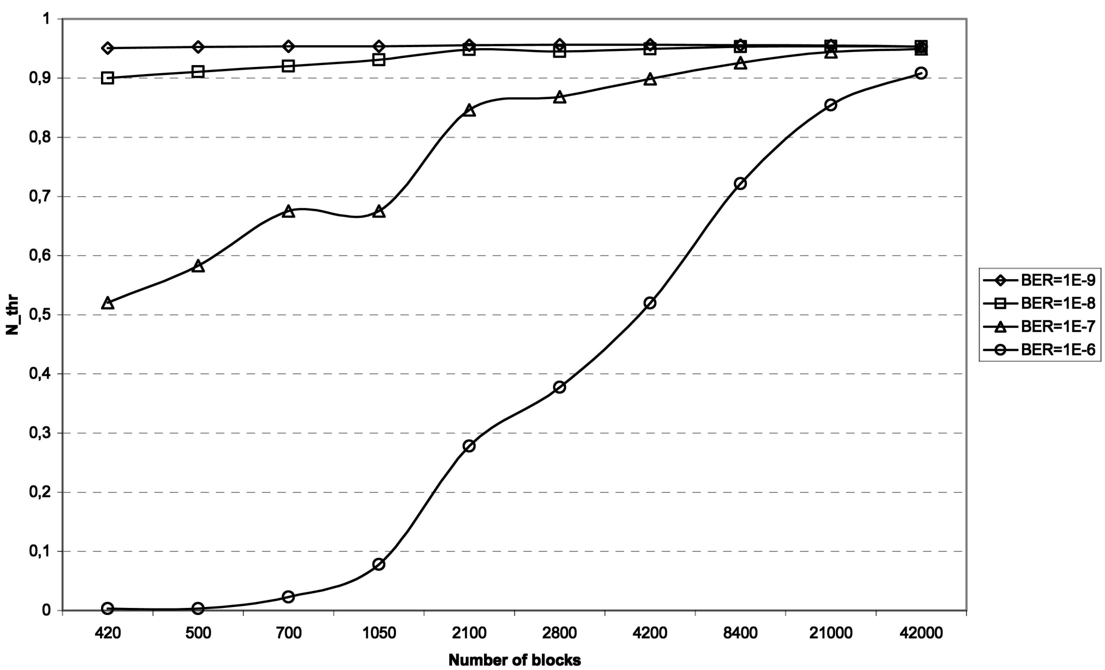

Fig. 21. Normalized throughput versus number of blocks for 1 transmission.

The role of the repetitions may be better understood from Fig. 25, where the ratio of success is shown versus the number of transmissions. The number of blocks giving the best performance in terms of *S_ratio* have been sketched for each value of the number of transmissions. Fig. 25 may be also operatively used to choose the number of times the file needs to be retransmitted in real environments, having some a priori knowledge about the channel conditions and about the application requirements (i.e., the percentage of losses that can be tolerated). The normalized throughput corresponding to the configurations of Fig. 25 is shown in Fig. 26.

C. Comparison of Results

This section is addressed to the comparison of the behavior experienced by the envisaged protocol solutions, namely “CFDP reliable NAK asynchronous” and “CFDP-repeat,” in terms of the normalized throughput. The aim of this discussion is to highlight the main elements arising from the

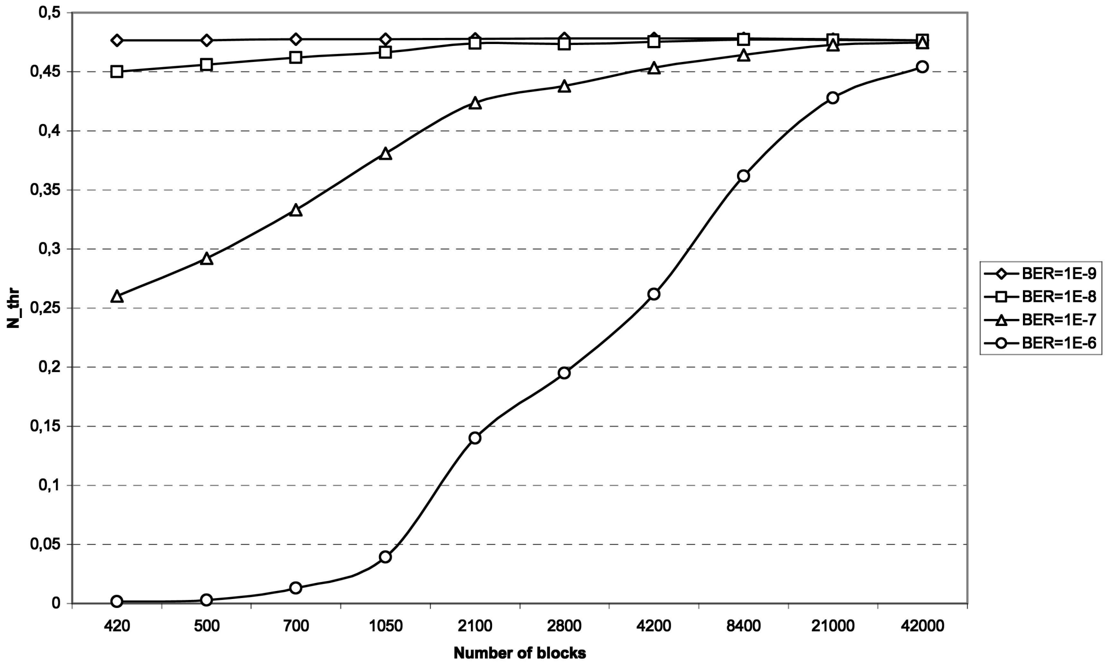

Fig. 22. Normalized throughput versus number of blocks for 2 transmissions.

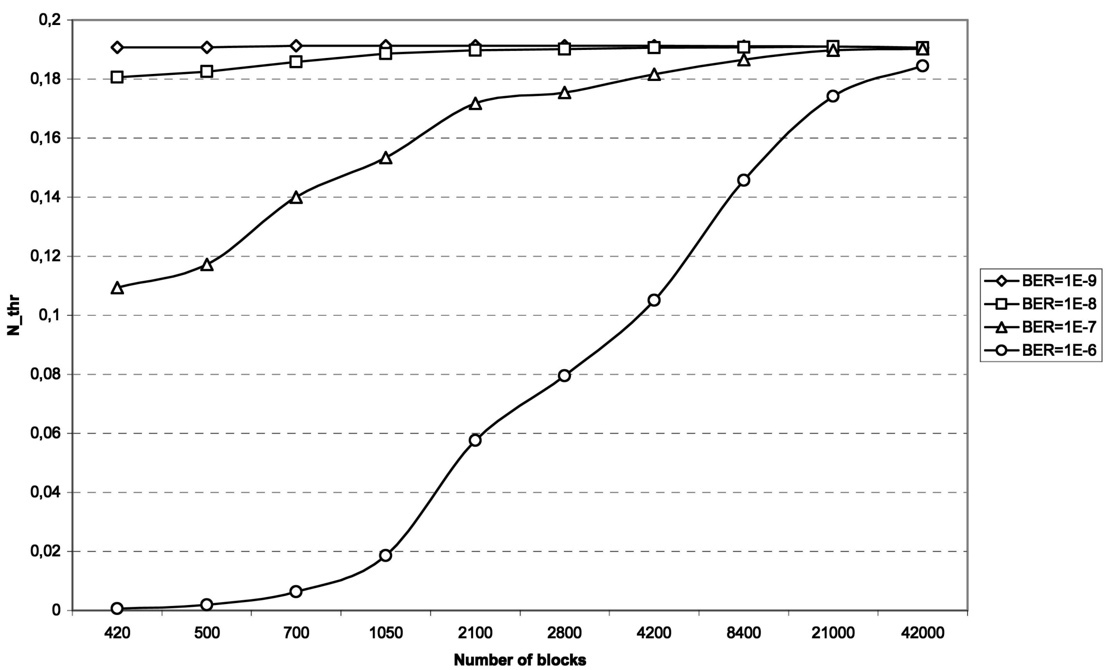

Fig. 23. Normalized throughput versus number of blocks for 5 transmissions.

analysis of data communication performed on the LEO/GEO environment, focusing on the impact that the lack of a return link has on the performance.

Fig. 27 summarizes the results of the investigation, showing the normalized throughput versus the algorithm used. Concerning "CFDP-repeat," the solutions assuring the highest normalized throughput have been selected for each BER value. "CFDP reliable NAK asynchronous" (simplified in "CFDP asynchronous" in Fig. 27) has been studied for different values of p (i.e., the probability of return link availability) and the "CFDP-repeat" behavior has been investigated for different number of transmissions (indicated in Fig. 27 as "CFDP-repeat" followed by the number of transmissions). Reported results show that the adoption of "CFDP reliable NAK asynchronous" (where 100% of the information always arrives at the destination) is very promising in cases with low BER values. Concerning "CFDP-repeat," Fig. 27 highlights how the tests performed with only 1 transmission are quite efficient, while, as the number of transmissions increases, the

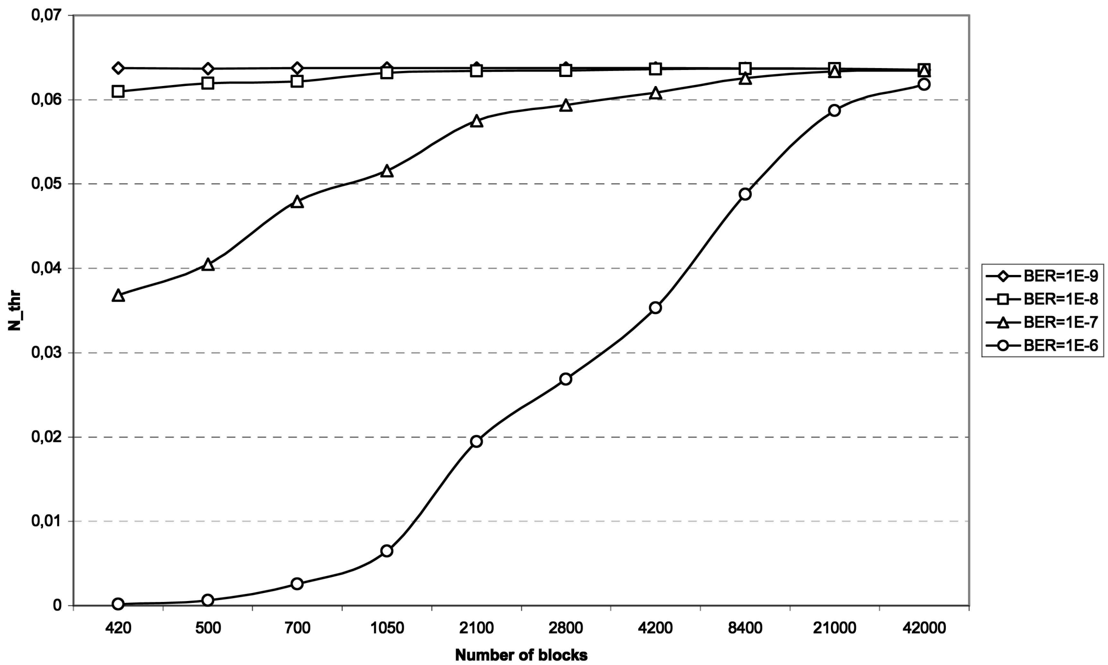

Fig. 24. Normalized throughput versus number of blocks for 15 transmissions.

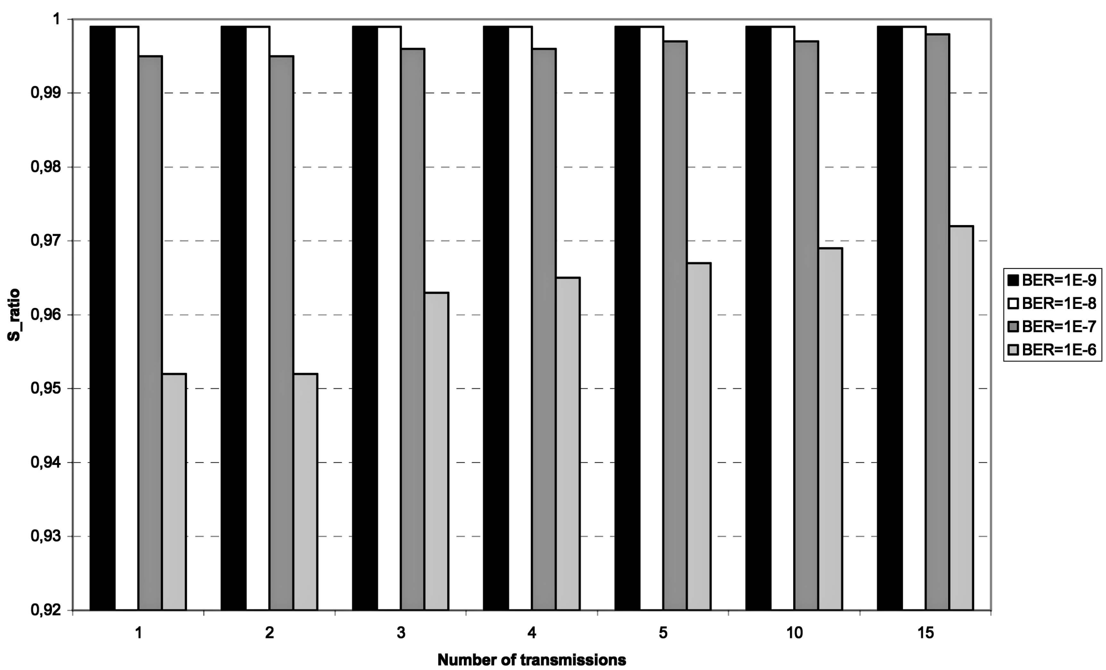

Fig. 25. Ratio of success versus number of transmissions.

performance collapses, registering a minimum for the normalized throughput of 0.06 for 15 transmissions.

It is worth considering the fact that the results related to "CFDP-repeat" do not depend on the return link availability and it is expected that the adoption of such solution would significantly extend the application environment. Moreover, in case of transmission of images, a ratio of success lower than 0.95 may be tolerated and hence "CFDP-repeat" may be a valid solution. Another aspect is also represented by the implementation complexity. "CFDP-repeat" is based on a trivial heuristics and its employment in space missions does not add any particular complexity to the system design. On the other hand, "CFDP-asynchronous" requires at least the partial availability of the return link and the emission of NAK notifications.

VI. END-TO-END PERSPECTIVE

Sections IV and V have pointed out the main impairments introduced by LEO and GEO portions

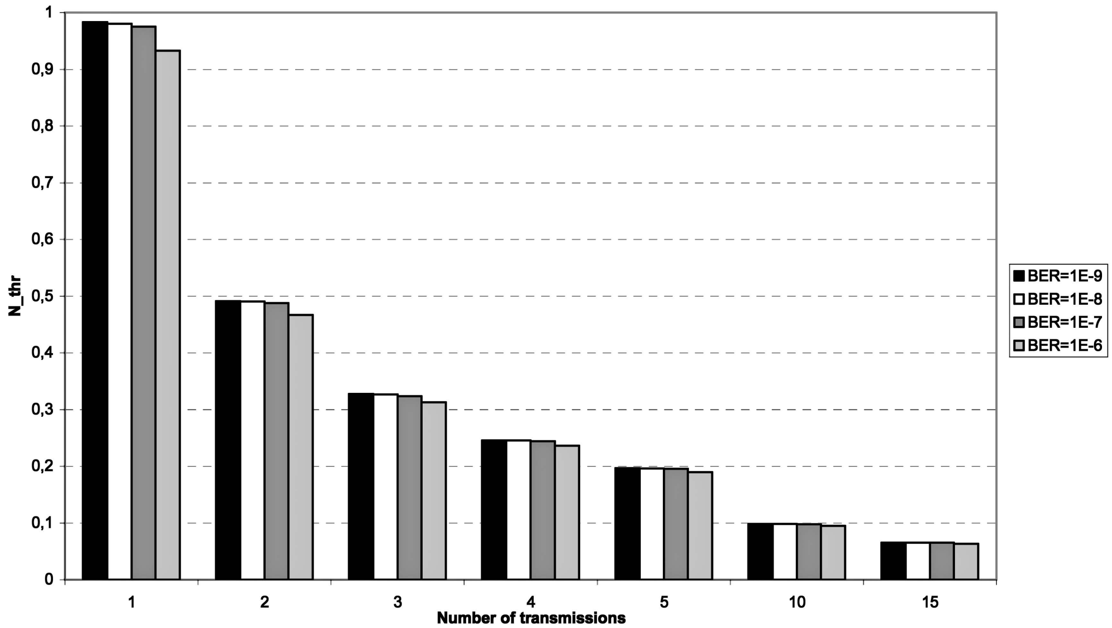

Fig. 26. Normalized throughput versus number of transmissions.

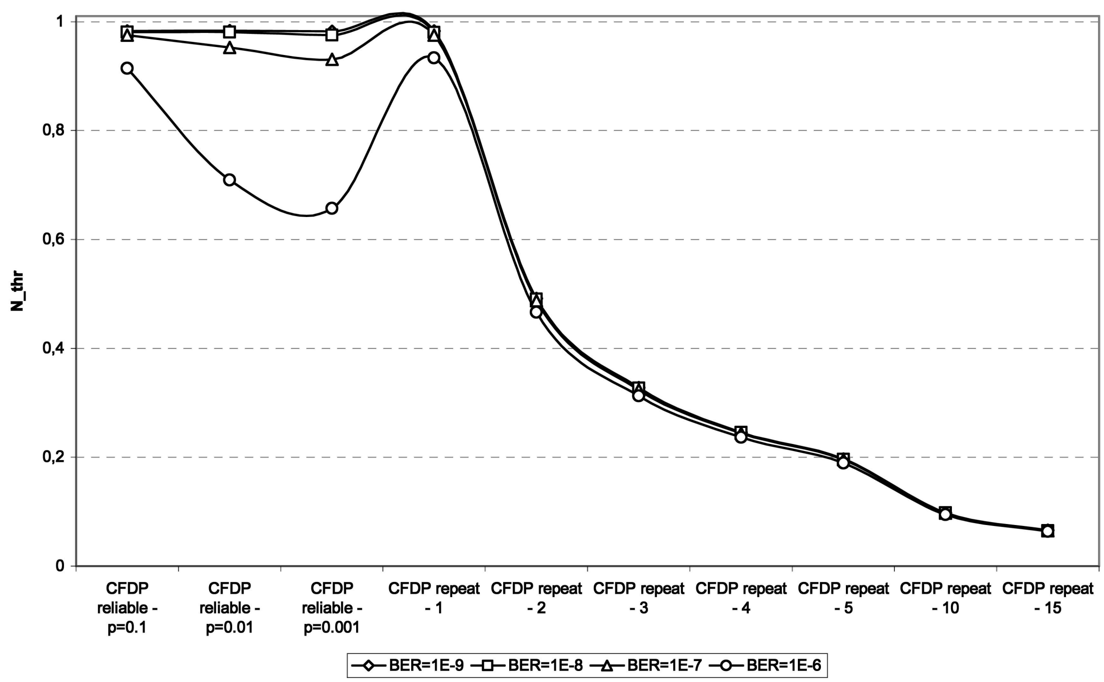

Fig. 27. Performance comparison: CFDP reliable NAK asynchronous and CFDP-repeat.

and possible solutions to mitigate these effects. In order to make the whole analysis more complete, the next step is to evaluate the performance of the end-to-end communication, taking as reference the protocol solutions that offer the best performance results, analyzed in the previous sections.

In the end-to-end perspective, a fundamental role is played by the limited visibility of DAVID satellite [32, 33] from the DAVID Earth station, which imposes the adoption of ad-hoc protocols. Suspend and resume mechanisms have to be guaranteed in order to "suspend" the data transaction whenever the amount of data exceeds the maximum quantity that can be transmitted in the period of LEO visibility. As a second issue, it is also important to take into account the limited visibility of the LEO platform with respect to the destination through the GEO satellite. Also in this case, suspend and resume capabilities are required in order to perform transmission operations only when the satellite link with ARTEMIS is available. Such store-and-forward operations make necessary some mass-storage equipment on board of the DAVID satellite.

The duration of visibility of the LEO from the DAVID Earth station ranges from 187 s to 309 s and a number of $2 \div 7$ passages is guaranteed for each site (visibility period and number of passages depends on the position of the LEO with respect to the Earth station) in order to assure the delivery of the amount of data. On the other hand, the duration of the period in which DAVID and ARTEMIS satellites are mutually visible is assumed to be 1638 s at maximum and a number of 10 passages is guaranteed. On the basis of the results collected in the performance evaluation phase and taking into account the capabilities of suspend/resume provided by CFDP implementation, one thinks of the CCSDS-based protocol stack as a promising solution. In more detail, considering the employment of "CFDP-reliable NAK immediate" on LEO link and of "CFDP reliable NAK asynchronous" on the other hop, a minimum transfer delay of 1905.7 s is registered when the channel is in very critical conditions (i.e., BER value of 10^{-6} on both links and probability of availability of the return link p equal to 0.001). In particular, this value is lower than the maximum available period, evaluated as the sum of the LEO and GEO visibility window durations.

The above consideration confirms the advantages provided by a full CCSDS-based protocol architecture, able to guarantee high performance results in terms of data delivery delay and to manage the communication even in the presence of hazardous network conditions, such as intermittent links, without deteriorating the overall data communication.

VII. CONCLUSIONS

The work carried out in this paper has focused on the study of protocol architectures able to cope with typical peculiarities of satellite channels, such as high bandwidth-delay product, relevant BERs, partial unavailability of the return channel and limited visibility periods of satellites. The whole investigation has been split into two separated frameworks, addressing communication issues for LEO and GEO satellite channels, respectively, and pointing out the advantages offered by CCSDS protocol stack in terms of reliability capability and high performance guarantees.

As far as LEO portion is concerned, "CFDP reliable NAK immediate" emerged as a good solution, offering satisfying results in terms of normalized throughput and overall transfer time, also in the presence of critical channel conditions (BER values of 10^{-7}, 10^{-6}), and assuring a lower complexity effort with respect to dedicated solutions (e.g. STP-IW86-320K), characterizing the splitting architecture approach and experiencing slightly lower performance results.

The employment of CCSDS-based protocol stack has been found to be also useful in the case of the GEO scenario, where the partial unavailability of the return channel, used for the transportation of feedback information, makes the adoption of TCP-IP protocol stack unpractical. The "CFDP reliable NAK asynchronous" solution exploits the limited availability of the return channel to perform ARQ schemes and provides satisfying results in terms of normalized throughput; consequently, it may be adopted for data communications presenting real-time constraints. On the other hand, "CFDP-repeat" solution, relying on a repeated transmission mechanism, presents the drawback of wasting the available channel bandwidth but it is preferred in the transmissions of images whose characteristics can often tolerate the partial arrival of the overall content and some degree of loss.

Finally, the impact of the limited visibility of satellites involved in the data communication has been envisaged highlighting the advantages offered by the CCSDS protocol stack in terms of suspend/resume features, necessary to assure the reliability of the end-to-end communication.

REFERENCES

[1] Zielinski, R. J., Wieckowski, W., and Bem, J.
Broadband satellite systems.
IEEE Communications Surveys & Tutorials.
3, 1 (First Quarter 2000), 2–15.

[2] Bonifazi, C., Ruggieri, M., Pratesi, M., Salome, A., Varacalli, G., Paraboni, A., and Saggese, E.
The DAVID satellite mission of the Italian Space Agency: High rate data transmission to Internet at W and Ka bands.
In *Proceedings of the IEEE International Conference on Communications* (ICC 2000), New York, **25**, 1 (May 2002), 3022–3026.

[3] Bonifazi, C., Ruggieri, M., and Paraboni, A.
The DAVID mission in the heritage of the SIRIO and ITALSAT satellites.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 4 (Oct. 2002), 1371–1376.

[4] Information Sciences Institute, University of Southern California
Transmission control protocol—Darpa internet program—Protocol specification.
IETF, RFC 793, Sept. 1981.

[5] Consultative Committee for Space Data Systems (CCSDS)
http://www.ccsds.org.

[6] Jacobson, V., Braden, R., and Borman, D.
TCP extensions for high performance.
IETF, RFC 1323, May 1992.

[7] Lakshman, T. V., and Madhow, U.
The performance of TCP/IP for networks with high bandwidth-delay products and random loss.
IEEE/ACM Transactions on Networking, **5**, 3 (June 1997), 336–350.

[8] Allman, M., Paxson, V., and Stevens, W. S.
TCP congestion control.
IETF, RFC 2581, Apr. 1999.

[9] Barakat, C., Altman, E., and Dabbous, W.
On TCP performance in a heterogeneous network: A survey.
IEEE Communications Magazine, **38**, 1 (Jan. 2000), 40–46.

[10] Henderson, T. R., and Katz, R. H.
Transport protocols for internet-compatible satellite networks.
IEEE Journal on Selected Areas in Communications, **17**, 2 (Feb. 1999), 326–344.

[11] Ludwig, R.
A case for flow-adaptive wireless links.
Technical report UCB//CSD-99-1053, University of California at Berkeley, May 1999.

[12] Allman, M., Glover, D., and Sanchez, L.
Enhancing TCP over satellite channels using standard mechanisms.
IETF, RFC 2488, Jan. 1999.

[13] Allman, M., Dawkinks, S., Glover, D., Griner, J., Henderson, T., Heidemann, J., Ostermann, S., Scott, K., Semke, J., Touch, J., and Tran, D.
Ongoing TCP research related to satellites.
IETF, RFC 2760, Feb. 2000.

[14] Akyildiz, I. F., Zhang, X., and Fang, J.
TCP-Peach+: Enhancement of TCP-Peach for satellite IP networks.
IEEE Communications Letters, **6**, 7 (July 2002), 303–305.

[15] Wang, R., Pau, G., Yamada, K., Sanadidi, M. Y., and Gerla, M.
TCP startup performance in large bandwidth delay networks.
Hong Kong, *INFOCOM 2004*, **23**, 1 (Mar. 2004), 795–804.

[16] Katabi, D., Handley, M., and Rohrs, C.
Internet congestion control for future high bandwidth-delay product environments.
ACM Sigcomm 2002, Pittsburgh, Aug. 2002, 89–102.

[17] Ramakrishnan, K., Floyd, S., and Black, D.
The addition of explicit congestion notification (ECN) to IP.
IETF, RFC 3168, Sept. 2001.

[18] Border, J., Kojo, M., Griner, J., Montenegro, G., and Shelby, Z.
Performance enhancing proxies intended to mitigate link-related degradations.
IETF, RFC 3135, June 2001.

[19] Ghani, N., and Dixit, S.
TCP/IP enhancement for satellite networks.
IEEE Communications Magazine, **37**, 7 (July 1999), 64–72.

[20] Bharadwaj, V. G., Baras, J. S., and Butts, N. P.
An architecture for internet service via broadband satellite networks.
International Journal of Satellite Communications, (special issue on IP), **19**, 1 (Jan./Feb. 2001), 29–50.

[21] Partridge, C., and Shepard, T. J.
TCP/IP performance over satellite links.
IEEE Network, **11**, 5 (Sept./Oct. 1997), 44–49.

[22] Daniel, L.
Satellite-link aware communication protocol (S-LACP).
http://sahara.cs.berkeley.edu/summer2003/index.html, Aug. 2003.

[23] Eddy, W., Ostermann, S., and Allman, M.
New techniques for making transport protocols robust to corruption-based loss.
ACM Computer Communication Review, **34**, 5 (Oct. 2004), 75–88.

[24] Allman, M., and Ostermann, S.
Multiple data connection FTP extensions.
Technical report TR-19971, Ohio University Computer Science, Feb. 1997.

[25] Postel, J., and Reynolds, J.
File transfer protocol (FTP).
IETF, RFC 959, Oct. 1985.

[26] Floyd, S., and Fall, K.
Promoting the use of end-to-end congestion control in the internet.
IEEE/ACM Transactions on Networking, **7**, 4 (Aug. 1999), 458–472.

[27] Lin, W. K., Chiu, D. M., and Lee, Y. B.
Erasure code replication revisited.
In *Proceedings of the Fourth IEEE International Conference on Peer-to-Peer Computing*, Zürich, Switzerland, Aug. 2004, 90–97.

[28] Consultative Committee for Space Data Systems (CCSDS)
Space Communications Protocol Specification-Transport Protocol, CCSDS 714.0-B-1, Blue Book, Issue 1, May 1999.

[29] Consultative Committee for Space Data Systems (CCSDS)
CCSDS File Delivery Protocol, CCSDS 727.0-B-2, Blue Book, Issue 2, Oct. 2002.

[30] Consultative Committee for Space Data Systems (CCSDS)
Proximity-1 Space Link Protocol—Coding and Synchronization Sublayer, CCSDS 211.2-B-1, Blue Book, Issue 1, Apr. 2003.

[31] Lee, D. C., and Baek, W.
Expected file-delivery time of deferred NAK ARQ in CCSDS file-delivery protocol.
IEEE Transactions on Communications, **52**, 8 (Aug. 2004), 1408–1416.

[32] De Fina, S., Ruggieri, M., and Bosisio, A. V.
Exploitation of the W-band for high capacity satellite communications.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 1 (Jan. 2003), 82–93.

[33] Ruggieri, M., De Fina, S., Pratesi, M., Saggese, E., and Bonifazi, C.
The W-band data collection experiment of the DAVID mission.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 4 (Oct. 2002), 1377–1387.

[34] Floyd, S., Henderson, T., and Gurtov, A.
The NewReno modification to TCP's fast recovery algorithm.
IETF, RFC 3782, Apr. 2004.

[35] Mathis, M., Mahdavi, J., and Romanow, A.
TCP selective acknowledgement options.
IETF, RFC 2018, Oct. 1996.

[36] Marchese, M.
TCP modifications over satellite channels: Study and performance evaluation.
International Journal of Satellite Communications, (special issue on IP), **19**, 1 (Jan./Feb. 2001), 93–110.

[37] Consultative Committee for Space Data Systems (CCSDS)
Overview of Space Link Protocols, CCSDS 130.0-G-1, Green Book, Issue 1, June 2001.

Tomaso de Cola was born in Manosque, France, on April 28, 1977. He received the "Laurea" degree (summa cum laude) in telecommunication engineering at the University of Genoa, Genoa, Italy, in 2001 and the Qualification degree as professional engineer in 2002.

Since 2002, he has been with the Italian Consortium of Telecommunications (CNIT), University of Genoa Research Unit, where he is a research scientist. His main research concerns are TCP/IP protocols, satellite networks, transport protocols for wireless links, interplanetary networks, and delay tolerant networks.

Mario Marchese (S'90—M'97) was born in Genoa, Italy in 1967. He received the his "Laurea" degree (cum laude) at the University of Genoa, Italy in 1992 and the Qualification degree as professional engineer in April 1992. He obtained the Ph.D. (Italian "Dottorato di Ricerca") degree in telecommunications at the University of Genoa in 1996.

From 1999 to 2004, he worked with the Italian Consortium of Telecommunications (CNIT), by the University of Genoa Research Unit, where he was head of research. Since February 2005 he has been associate professor at the University of Genoa, Department of Communication, Computer and Systems Science (DIST). He is the founder of and still the technically responsible for CNIT/DIST Satellite Communication and Networking Laboratory (SCNL) of the University of Genoa, which contains high value devices and tools and implies the management of different units of specialized scientific and technical personnel. His main research concerns are satellite networks, transport layer over satellite and wireless networks, quality of service over ATM, IP, and MPLS, and data transport over heterogeneous networks.

Dr. Marchese is vice-chair of the IEEE Satellite and Space Communications Technical Committee. He is author and coauthor of more than 80 scientific works, including international magazines, international conferences, and book chapters.