INVITED PAPER

Reliability Options for Data Communications in the Future Deep-Space Missions

Reliability options that are available for application in deep-space missions are surveyed in this paper.

By Tomaso de Cola, *Student Member IEEE*, Enrico Paolini, *Member IEEE*, Gianluigi Liva, *Member IEEE*, and Gian Paolo Calzolari

ABSTRACT | Availability of higher capacity for both uplinks and downlinks is expected in the future deep-space missions on Mars, thus enabling a large range of services that could eventually support human remote operations. The provisioning for deep-space links offering data rate up to several megabits per second will be a crucial element to allow new services for the space domain along with the common telecommand and telemetry services with enhanced communication capabilities. On the other hand, also the geometry proper of this scenario with orbiting and landed elements sharing only partial visibility among them and towards Earth provides another challenge. This paper surveys the reliability options that are available in the Consultative Committee for Space Data Systems (CCSDS) Protocol Stack for application in the deep-space missions. In particular, the solutions implemented from the physical up to the application layer are illustrated in terms of channel coding and Automatic Retransmission reQuest (ARQ) schemes. Finally, advanced reliability strategies possibly applicable in next-generation deep-space missions are explored as well.

KEYWORDS | Channel coding; Consultative Committee for Space Data Systems (CCSDS); deep-space communications; delay/disruption tolerant network (DTN); erasure coding

Manuscript received October 21, 2010; revised March 17, 2011; accepted May 22, 2011. Date of publication August 12, 2011; date of current version October 19, 2011.
T. de Cola and **G. Liva** are with the German Aerospace Center (DLR), Institute of Communications and Navigation, 82234 Weßling, Germany (e-mail: Tomaso.decola@dlr.de; Gianluigi.liva@dlr.de).
E. Paolini is with the Department of Electronics, Computer Science and Systems (DEIS), University of Bologna, 47521 Cesena (FC), Italy (e-mail: e.paolini@unibo.it).
G. P. Calzolari is with the European Space Operations Centre, European Space Agency, 64293 Darmstadt, Germany (e-mail: Gian.Paolo.Calzolari@esa.int).

Digital Object Identifier: 10.1109/JPROC.2011.2159571

I. INTRODUCTION

Since early 1980s, the Consultative Committee for Space Data Systems (CCSDS) [1] has been playing a central role in space missions, mainly in terms of protocol recommendations and overall standardization activities. In particular, the tasks carried out by CCSDS have not only addressed communication and networking aspects, but also the interoperability of technological solutions and the interaction between different space agencies within the same space mission in terms of cross support [2].

Over the years, CCSDS has standardized a set of space–Earth and space–space communication protocols, coding schemes, and modulations based on state-of-the-art techniques. Indeed, since the needs of missions differ depending upon their profile, a single type of standard would not satisfy all the needs and possible selections will be offered to the users.

For instance, deep-space missions operate at low data rate and have, in general, rather mild bandwidth constraints; on the other hand, link performances are crucial and high coding gains are required while the large propagation delay imposes stringent requirements and operability constraints. Conversely, near-Earth missions, be they for space research, for space operations, or for Earth exploration, may operate at high or very high data rates on their telemetry link and require, in general, a compromise between coding gain and bandwidth expansion. At the same time, fast reception and reaction can be essential to operate the spacecraft and fast processing (at least part) of the huge amount of data is required.

Today, the demand in channel capacity of space missions is steadily rising, calling for a move to Ka-band in the coming years, for wider bandwidth and higher data rate capabilities. Ka-band is already used for deep-space missions (Mars, Mercury, etc.), and future near-Earth missions have identified Ka-band as the optimum range for their high rate telemetry transmissions. This is creating new mission profiles that involve highly dynamic links such as Earth exploration satellites operating on low Earth orbits with an increasing amount of data to dump to Earth during the short contacts. Consequently, ground stations require ever increasing data throughput. Besides, techniques based on variable coding and modulation (VCM) are means to account for the dynamics of the link geometry while more advanced algorithms (e.g., data compression) and protocols will take other aspects to ensure efficient exploitation of space and ground systems.

Space science and exploration have been a major target of space agencies since the very beginning of their activities. Missions to, e.g., Venus, Mars, and Saturn's moon Titan, with the exploration of the Solar System highly increased our understanding about Earth's relationship with the other planets and this research is going to continue in the next decades to augment the knowledge of our own Solar System. Such deep-space missions require long times for preparation, implementation, and operation and the current world situation makes inconceivable such future activities without international cooperation both in space and on ground. As a consequence, future space mission scenarios are envisioned with richer topologies involving multiple spacecraft and with data flowing across multiple hops and over multiple paths to achieve end-to-end data transfers. This may imply that a spacecraft or a lander will rely on "foreign" space or ground elements for relay operations that will allow information exchange with possible delays or losses.

This evolution of space exploration scenarios towards more complex communications topologies (encompassing more intermediate nodes, thus offering alternate communication paths) together with the steadily rising demand in channel capacity of space missions regardless of onboard storage capabilities do create the need for new alternatives within those standards produced and maintained by CCSDS. From this standpoint it is worth noting that ensuring reliable communications, for common telemetry/telecommand messages along with data transfers (e.g., images, files), is one of the primary requirements of space missions. In this respect, this paper focuses on the reliability options currently available within the CCSDS protocol stack, by also analyzing advanced features that could be applied in future space missions.

This paper is structured as follows. Section II is devoted to the description of the overall CCSDS protocol stack, from the physical up to the application layer. The details about the reliability options available from each protocol in terms of channel coding and Automatic Retransmission reQuest (ARQ) are provided in Section III, whereas the future trends are introduced in Section IV, where overall conclusions are also drawn.

II. THE CCSDS PROTOCOL ARCHITECTURE

An overview of the CCSDS communication stack is depicted in Fig. 1 [3]. Originally (early 1980s), the CCSDS

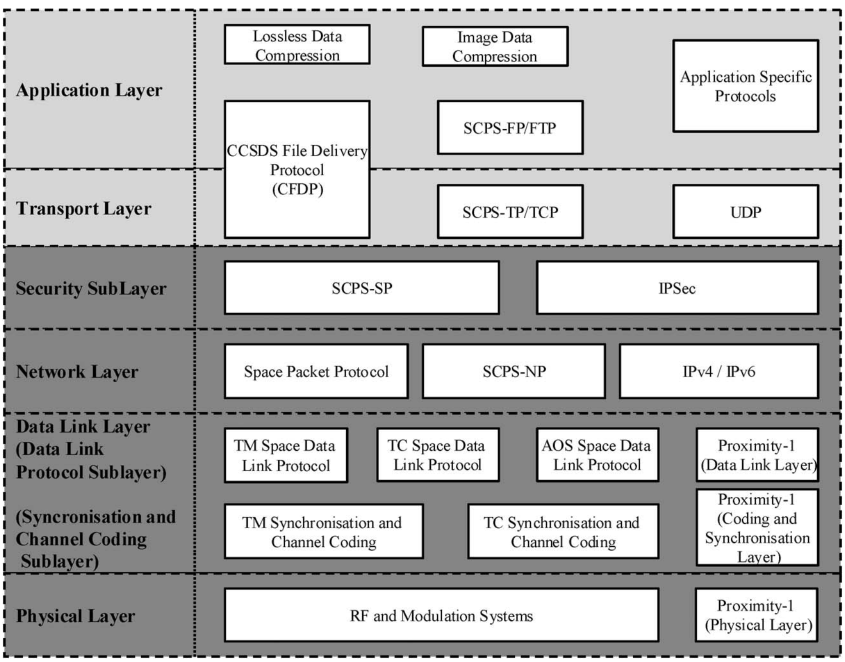

Fig. 1. ***CCSDS protocol stack.***

communication protocols were developed to define a standard for packet telemetry (TM) [4], [5], where data generated on a spacecraft are transmitted to the ground as a stream of fixed-length transfer frames, and a standard for packet telecommand (TC), where commands are sent to a spacecraft in a stream of sporadic and variable-length transfer frames. A few years later, a standard for both space-to-ground and ground-to-space links known as advanced orbiting systems (AOS) [6], specifically designed for the International Space Station (ISS) operations [7], was added to include "online" data (such as video and voice) to the packet telemetry. The current CCSDS recommendations for TM, TC, and AOS are available in [8]–[10], respectively. The corresponding CCSDS specifications for synchronization and channel coding to be used over synchronous (TM and AOS) and asynchronous (TC) links can be found in [11] and [12], respectively, while the physical layer specifications are available in [13]. Moreover, the space packet protocol (SPP), to be used over the above data link layer protocols, is specified in [14]. Originally designed as an application layer protocol to receive data from onboard applications or to send commands to them in a point-to-point context, the SPP was promoted to a network protocol (through the concept of logical path) after AOS was developed.

The CCSDS data link layer protocols are completed by the more recent Proximity-1 space link protocol [15], to be mainly used for radio communications between landers, rovers, and orbiting constellations/relays. The protocol is characterized by both dedicated synchronization and channel coding functions [16], and a dedicated physical layer [17].

Before the introduction of Proximity-1, the CCSDS communication stack had been already enriched with the so called space communications protocol specifications (SCPSs), a set of four Internet-like protocols spanning from the network to the application layer. Out of them, only the SCPS transport protocol (SCPS TP) [18] is still part of the CCSDS recommendation, the others being obsolete [19]–[21]. Transport layer functionalities are provided by the CCSDS file delivery protocol (CFDP) [22], which offers end-to-end transport services for the file transfer to/from onboard mass memories, and which is intended for use over TM, TC, and AOS link services. Finally, traditional Internet transport layer protocols such as TCP [23] and UDP [24] may be used over space links, on top of IP Version 4 (IPv4) [25] and IP Version 6 (IPv6) [26]. Actually, TCP is being used only on short-delay links, whereas UDP has a potential broader applicability, though at cost of unreliable data delivery.

Interoperability of IP-based protocols with CCSDS space links is ensured by the CCSDS encapsulation packet service (IPoC) [27] and the CCSDS IP extensions (IPE) [28]. IPoC has actually a broader applicability and provides a service to transfer data units that are not directly handled by the space data link protocols. IPE provides an interoperable way of identifying the Internet protocols encapsulated by the CCSDS encapsulation service. IPoC and IPE together constitute the building blocks for the implementation of IP over CCSDS space protocols. Their standardization is under finalization.

Finally, a note has to be reserved to internetworking services for space missions, which have been recently investigated by the Space Internetworking Strategy Group (SISG) to provide the Solar System with networking functionalities [29]. In particular, the store/forwarding features available from the delay/disruption-tolerant networking (DTN) protocol architecture [30], [31] standardized within the Internet Engineering Task Force (IETF) are now considered by the CCSDS for the adoption of the DTN architecture in futures deep-space missions.

A. Physical and Datalink Layers

The CCSDS physical layer specified in the recommendation [13] is common to TM, TC, and AOS links. The allowed modulations are binary phase-shift keying (BPSK), (offset) quadrature phase-shift keying (OQPSK and QPSK, respectively) for low-rate links ($<$ 2 Msymbol/s), while for high data rate links ($>$ 2 Msymbol/s) more spectrally efficient modulations are considered, such as Gaussian minimum shift keying (GMSK) with a time-bandwidth product $BT_s = 0.25$, filtered OQPSK, or 8-PSK. The Proximity-1 space link protocol foresees a pulse code modulation bi-phase-encoded (PCM/Bi-ϕ), which renders the carrier acquisition and tracking easier [32], [33]. The reader interested in the technical details of modulation issues, spectral efficiency, and carrier recovery in CCSDS can refer to [34].

The CCSDS data link layer is split into two sublayers, namely, synchronization and channel coding sublayer and data link protocol sublayer, a subdivision which holds for TM, TC, AOS, and Proximity-1. The TM space data link protocol (TM SDLP) [8], the TC space data link protocol (TC SDLP) [9], and the AOS space data link protocol (AOS SDLP) [10] provide several link services for data transmission in space-to-ground and ground-to-space links. Analogous functions for space-to-space links are provided by the Proximity-1 SDLP [15].

The TM SDLP employs data units called TM transfer frames (TM TFs), and relies on the functions offered by the TM synchronization and channel coding sublayer, such as frame delimiting, frame synchronizing, bit transition generation and removal, and error control coding. The synchronous stream of TM TFs includes only idle frames (OIFs) if necessary. The length of TM TFs is constant during an entire mission phase.

The main functions offered by the TC SDLP may be summarized as segmentation/blocking and transmission control. Segmentation consists of breaking large service data units into small pieces, while blocking consists of grouping small service data into longer pieces. The transmission control function is performed through an automatic retransmission mechanism known as communications operation procedure (COP) [35]. Two types of data units are used: variable-length TC transfer frames (TC TFs) to send commands on the uplink to the spacecraft and communications link control words (CLCWs) to transmit reports on the downlink to the ground about the status of TC TFs delivery.[1] The TC SDLP exploits the synchronization and error control coding services offered by the TC synchronization and channel coding sublayer.

Similarly to the TM SDLP, the AOS SDLP can operate on top of the TM synchronization and channel coding sublayer, and uses data units whose length is maintained constant during each mission phase. The AOS SDLP data units are called AOS transfer frames (AOS TFs), or version-2 TFs, and their format is only slightly different from that of TM TFs. In all three data link protocols (TM, AOS, and TC), the stream of TFs is organized into virtual channels (VCs), where TFs belonging to the same VC share common requirements.

B. Networking and Application Layers

Unlike physical and data-link layer protocol architecture, the specification of the higher layers is less organized, because of a large number of protocol candidates, either devised within CCSDS or inherited from IETF [36] and Internet Research Task Force (IRTF) [37].

1) Network Layer: Two protocols have been designed directly within CCSDS: space packet protocol [14] and space communication protocol standards-network protocol (SCPS-NP) [19], although the second one is no longer a CCSDS standard. Both are responsible for addressing and routing operations, by means of path, end system addresses, and other specific identifiers [14], [28]. The first makes use of application identifiers (APID), whereas the second builds on system and path addresses for the addressing and consequent routing operations. On the other hand, the Internet protocol (versions 4 and 6) [25], [26] is the solution standardized by IETF and actually the well-known standard *de facto* for Internet.

2) Transport Layer: It is worthwhile noting that, though recommendations for the transport layer have been produced within CCSDS, the use of transport protocols is not mandatory in CCSDS. In practice, most applications, such as CFDP, do not require running over a transport protocol, but can work directly on top of the network layer. The CCSDS developed the SCPS transport protocol (SCPS-TP) [18] to provide end-to-end reliable communication, based on TCP [23] and improved for the deep-space environment.

3) Application Layer: CFDP [22] is designed to get reliable transfers of files by following an FTP-like paradigm [38]. Its implementation spans the application and transport layers. In addition to file transfer, also event-driven asynchronous message exchange is expected in the future for deep-space communications, established among spacecraft and remote stations. In this perspective, the CCSDS has developed an asynchronous message service (AMS) [39], conceived to provide a messaging layer over which the protocol messages of the mission operation services may be carried. AMS is effective to streaming engineering (housekeeping) data, real-time commanding, and allowing for continuous collaborative operations among robotic crafts.

[1]CLCWs are transmitted in telemetry. Specifically, a field of TM TFs called operational control field (OCF) is used to transfer CLCWs.

III. RELIABILITY OPTIONS

As already introduced in Sections I and II, performing data communications in deep space requires accurate protocol configuration at different layers and on all the involved nodes. This necessity stems from the very large propagation delays, the scarce bandwidth availability, the limited onboard storage, and the time-constrained visibility window between spacecraft and ground stations. This has resulted in the design of specific CCSDS reliability mechanisms running from physical up to application layer, with parameter configurations (e.g., code-rate, block length, max number of retransmissions, etc.) very much depending on the mission peculiarities.

A. Coding at the Datalink Layer

1) AOS/TM/TC/Proximity-1: AOS and TM links share the same synchronization and channel coding sublayer, specified in [11], whose services are employed by AOS in uplink as well. The channel coding schemes currently supported by the standard are convolutional, Reed–Solomon (RS), and turbo codes.

The CCSDS convolutional code (CC) is a transparent, rate-1/2, binary code with constraint length 7 and octal generators (171,133). Higher rate CCs are obtained by puncturing the rate-1/2 code, leading to the code rates 2/3, 3/4, 5/6, and 7/8.

Two RS codes on the Galois field GF(256) are recommended, namely, a (255,223) code with error-correction capability of 16 symbols, and a (255,239) code with error-correction capability of eight symbols. Concatenated codes using as an outer code one of the above-mentioned RS codes, and as an inner code the rate-1/2 CC (or one of its punctured versions) are recommended with block, byte-wise interleaving, via interleaving matrices of $n \times I$ bytes, with $n = 255$ and $I = 1, 2, 3, 4$ or 5. The family of CCSDS convolutional turbo codes comprises the code rates 1/2, 1/3, 1/4, and 1/6, and interleaver lengths of 1784, 3568, 7136, 8920, or 16384 bits. The turbo codes are obtained by puncturing a parallel concatenation of two rate-1/4, 16-states recursive systematic CCs. For a thorough description of the properties and performance of all

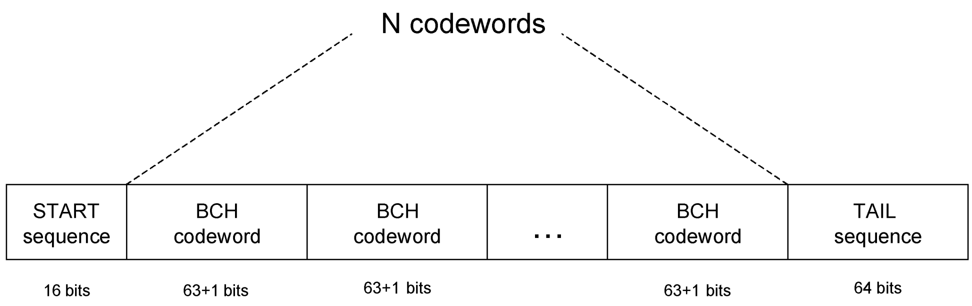

Fig. 2. ***Structure of a TC CLTU. A "0" bit is appended to each of the contiguous BCH codewords in the CLTU data field so that the overall data field length is a multiple of 64. Each codeword is obtained by breaking one TC TF, or a few TC TFs, into 56-bit blocks (with the addition of padding in the last block) and encoding them through the BCH encoder.***

above-mentioned codes, we refer the reader to [40], and also to [41] for turbo codes.

The CCSDS is currently finalizing the inclusion of low-density parity-check (LDPC) codes [42] in the TM recommendation [43]. The CCSDS LDPC codes have code rates 1/2, 2/3, 4/5, and 7/8 and information block lengths 1024, 4096, and 16 384. For a description of the parity-check matrix construction and of the performances offered by CCSDS LDPC codes, the reader is referred to [41] and [44].

The TC link in deep-space missions is usually characterized by data rates much lower than TM. For TC, error detection is at least as important as error correction: Undetected errors are highly undesirable, since they cause forwarding of dangerous, wrong commands to the spacecraft. The current CCSDS TC recommendation [12] adopts a (63,56) Bose, Ray–Chaudhuri, Hocquenghem (BCH) code with generator polynomial

$$g(X) = X^7 + X^6 + X + 1 = (X^6 + X + 1)(X + 1). \quad (1)$$

The code is an expurgated (63,57) Hamming code, obtained by allowing only the even-weight codewords, and its minimum distance is $d_{\min} = 4$. A TC TF is divided into N blocks, each of which is 56 bits long (padding is added for the last block if needed) and is individually BCH encoded. The obtained N contiguous BCH codewords, each one followed by a "0" bit for a total of 64 N bits, is encapsulated between a 16-bit start sequence (for synchronization purposes) and a 64-bit tail sequence to form a communication link transmission unit (CLTU), i.e., a TC synchronization and channel coding sublayer data structure (see Fig. 2).[2] The (63,56) BCH code may be operated in two manners: Single-error-correction (SEC) mode or triple-error-detection (TED) mode. In SEC mode, single errors are corrected and double errors detected, while in TED mode the BCH code acts simply as an error detection code, allowing the detection of all single, double, and triple errors and of all odd-weight error patterns. The TED mode is recommended whenever extremely low undetected error probabilities are required. In both modes, a further 16-bit cyclic redundancy check (CRC, see Section III-A3) may be employed at a frame level to further protect the TC TF,[3] where a frame error control field (FECF) is mandatory.

The main metric adopted to measure the reliability of the TC scheme is the *undetected* error probability, i.e., the probability that an erroneous TC TF is accepted as correct by the spacecraft and its data field delivered to the upper layers. The undetected error probability shall not exceed 10^{-9}. A second performance metric is represented by the rejection probability, defined as the probability that a TC TF is not accepted due to detected (but uncorrectable, in SEC mode) errors, where an entire CLTU is deleted if at least one of its N BCH codewords is marked as erroneous. This probability shall not exceed 10^{-3}. An analysis of the two performance metrics is provided in [46]. The requirement on the undetected error probability is achieved in TED mode without employing the CRC and for a wide range of values of N, while in SEC mode the use of the CRC is fundamental to meet the 10^{-9} requirement, already for N in the order of a few tens. When the 16-bit CRC is used (see Section III-A3), the estimated values of the undetected error rates are in the order of 10^{-22} in TED mode and of 10^{-17} in SEC mode.

In Proximity-1 links, the (7,1/2) CCSDS convolutional code is employed, with a mandatory 32-bit CRC (see Section III-A3). Optionally, a concatenated (204,188) Reed–Solomon/(7,1/2) convolutional code is supported [16], [33], [47], [48].

2) Advances—Next Generation Uplink: Motivated by the need of faster uplinks for future Lunar exploration missions, the CCSDS next generation uplink (NGU) working

[2]A CLTU may also correspond to more than one TC TFs.

[3]It is worthwhile noting that, while the CCSDS recommendation includes both SEC and TED modes, only the SEC mode (with mandatory CRC) is recommended by the European Cooperation for Space Standardization (ECSS) [45].

group is currently investigating updates of the uplink specifications to allow data rates higher than 1 Mb/s (up to 10 Mb/s in case of manned missions).

The current TC specification is based on a the aforementioned (63,56) BCH code, which suffers for modest coding gains ($\sim$2 dB in SEC mode with respect to an uncoded BPSK transmission). To achieve higher data rates, new codes offering larger coding gains will be employed. Two possibilities are considered for achieving a substantial coding gain over the (63,56) BCH code: Use of longer codes and reduction of the coding rate. The decoding complexity shall be as limited as possible, resulting in limitations on the considered block sizes. For TC and emergency uplinks, codes with input block size k of 64, 128, and 256 bits and with code rate 1/2 are currently considered. Among the investigated solutions, short, near-regular binary protograph-based LDPC codes have been proposed [49], which allow operating within 2 dB from Shannon's sphere packing bound [50].

In the short block-length regime, algebraic codes and terminated convolutional codes with maximum-likelihood (ML) soft-decision decoding can remarkably outperform short LDPC codes under belief-propagation (BP) decoding with similar (n, k) parameters. An example is provided in Fig. 3, where the codeword error rate (CER) over the additive white Gaussian noise (AWGN) channel for codes with block length $n = 128$ bits and dimension $k = 64$ bits are compared, namely:

- a (128,64) binary protograph LDPC code from [49];
- a (128,64) extended BCH (eBCH) code with minimum distance $d_{\min} = 22$ from [51];
- a (128,64) LDPC code constructed on a Galois field of order 256 [52], [53], decoded via BP exploiting fast Walsh–Hadamard (FWH) transform at the check nodes [54], [55].

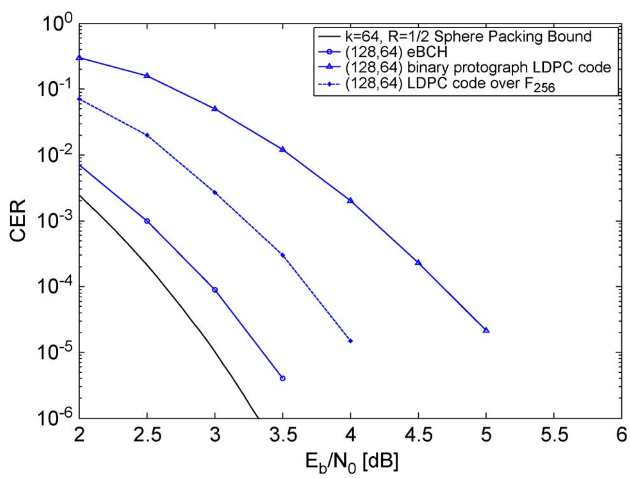

Fig. 3. ***Codeword error rates for various (128,64) codes, compared with the sphere packing bound.***

For the three codes, BPSK modulation is considered. The eBCH code under ML decoding approaches the sphere packing bound by about 0.2 dB. The binary protograph LDPC code shows a coding gain loss of about 1.7 dB at $\mathrm{CER} = 10^{-4}$ with respect to the eBCH code. In between, the nonbinary LDPC code gains roughly 1 dB over the binary LDPC code, and is outperformed by about 0.7 dB by the eBCH code.

Although at a first glance the eBCH code may look like the best choice from a pure CER performance viewpoint, the LDPC codes bring advantages that are not immediately visible. First, their decoding algorithms are simpler than that used for ML decoding of the eBCH code (this is even more true when considering the binary LDPC codes). Second, and even more important, LDPC codes are decoded via BP, an inherently *incomplete* decoding algorithm that offers an error detection mechanism. Thus, when an LDPC decoder fails, with high probability the output word is detected as erroneous without any additional frame validation test. This error detection capability can be further strengthened by comparing the decoder outputs and inputs according to a suitable metric [56], and discarding all decisions that fail either in this or in the parity-check matrix test. By doing that, the CER performance is further degraded, but the undetected error probability is highly reduced. For the eBCH code of the example above, all the decoding errors are virtually undetected due to the *complete* nature of its decoding algorithm.

Binary protograph-based LDPC codes are currently considered for the upgrade of the Proximity-1 protocol as well.

3) Frame Validation: The 16-bit FECF of TM, TC, and AOS TFs consists of the redundant bits of a CRC code, i.e., a binary systematic shortened cyclic code used to detect bit errors after transmission [57]. The FECF is calculated based on all the other fields of the TF, excluding the attached synch mark (ASM) used for frame synchronization purposes. (The term CRC refers to the parity bits produced by the encoding circuit and appended to the message before transmission. In this sense, CRC and FECF will be used interchangeably.) The CCSDS-recommended CRC for TM, TC, and AOS links is the CRC CCITT, whose generating polynomial is given by

$$\begin{aligned} g(X) &= X^{16} + X^{12} + X^5 + 1 \\ &= (X^{15} + X^{14} + X^{13} + X^{12} \\ &\quad + X^4 + X^3 + X^2 + X + 1) \cdot (X + 1). \end{aligned} \tag{2}$$

CRC encoding and error detection may be performed through a linear circuit based on a shift-register with feedback connections whose number of storage cells is equal to the degree of $g(X)$ [57]. Although the value to which the shift register storage cells are preset prior to encoding (on the transmitter side) and prior to syndrome calculation (on the receiver side) has in principle no effect on the undetected error probability, there are practical considerations leading to prefer an initial word to another one. For example, if all the storage cells in the CRC encoder are initialized to "0," the encoder has no state transition when an all-zero message is input, so that a nonzero initial word is preferred. Indeed, an all-one initial word is recommended in the TM, TC, and AOS recommendations ([8]–[10], respectively), where all the shift-register storage cells are preset to "1" prior to encoding and prior to decoding. Sometimes, this is referred to as "modified CRC" [58].

The error detection capability of any CRC depends on its generating polynomial $g(X)$ and on the length k of the message from which the CRC is calculated (and to which it is appended to form a codeword of length n). The minimum distance of the CRC-CCITT code is equal to 4 for $2 \leq k < 32\,752$ and to 2 for $k \geq 32\,752$. Then, if $2 \leq k < 32\,752$ any single, double, or triple error is detected. Since $(X+1)$ is a factor of $g(X) = X^{16} + X^{12} + X^5 + 1$, any pattern of an odd number of errors is also detected.

If the bit errors take place independently of each other with a sufficiently small probability P_e, then for $2 \leq k < 32\,752$ the undetected error probability of the CRC CCITT code can be approximated as

$$P_u \approx A_4 \cdot P_e^4 \cdot (1-P_e)^{n-4} \tag{3}$$

where $n = k + 16$ and where the values of A_4 as a function of k may be found in [59]. It is worthwhile noting that, however, in many real situations the errors are not independent but occur in bursts (this may be the case when using some types of error correcting codes). By definition, the received sequence is said to be affected by a burst error of length b when the error pattern only occurs over a span of b encoded bits. A CRC obtained by shortening a cyclic code is capable of detecting any error burst of length $b \leq n - k$ and large fractions of longer bursts. In the specific case of the CRC CCITT, the code is capable of detecting any burst error of length 16 or less. An analysis of the undetected error probability of cyclic and shortened cyclic codes affected by probabilistic burst errors can be found in [60] and [61].

A 32-bit CRC with generating polynomial

$$\begin{aligned} g(X) &= X^{32} + X^{23} + X^{21} + X^{11} + X^2 + 1 \\ &= (X^{21}+1) \cdot (X^{11} + X^2 + 1) \end{aligned} \tag{4}$$

is recommended for Proximity-1 links [16]. This polynomial generates a (42 987, 42 955) Fire code [62]. For all $2 \leq k \leq 42\,955$ the associated shortened code is capable to detect all single, double, and triple errors, all error patterns with odd multiplicity, all single error bursts of length 32 bits or less, and all two error bursts provided the shorter burst has length not greater than 11 and the sum of the two burst lengths is not greater than 22.

B. Automatic Retransmission reQuest (ARQ)

1) TC COP-1: Differently from CCSDS TM, AOS, and Proximity-1, CCSDS TC also implements an ARQ mechanism [9], [63] at the SDLP sublayer, complementing the channel coding schemes available from the synchronization and channel coding sublayer. CCSDS TC defines two data services: Sequence-controlled service and expedited service. Both are managed by the communication operations procedure-1 (COP-1) [35] that is actually responsible for reliable data delivery when needed.

As to sequence-controlled service, COP-1 implements a go-back-n ARQ mechanism, aimed at providing a reliable telecommand service, avoiding frame losses and duplications. Upon frame out-of-sequence detection, the receiver solicits the sender to retransmit the missed frames through a CLCW report, as described in Section II-A, and drops all the subsequent incoming TC frames. This mechanism is implemented through explicit retransmission request or expiration of a timer, set at the sender side. Clearly, this additional level of protection is worthwhile in low-delay scenarios (e.g., planet exploration or near-Earth environments). However, in case of large delays common in deep-space communications, this mechanism could severely degrade the telecommand service performance, because of the increased latency. Alternatively, it is possible to perform a repeated transmission of multiple frames, increasing the robustness of telecommand service delivery, provided the transmission timers are tuned consequently [64].

On the other hand, the management of expedited frames is lighter and does not include any retransmission mechanism. This differentiation with respect to the sequence-controlled service stems from the fact that expedited frames are defined for an "immediate" service requested in urgent space operations, as occurring during spacecraft recovery. In this case, higher priority and immediate telecommand frame delivery is requested, whereby telecommand delivery has to be ensured by channel coding [64].

2) TCP/SCPS-TP: The use of TCP over space links has been extensively studied in the last two decades by the scientific community, in order to identify possible optimizations or enhancements of the transport layer [65]–[67]. The peculiarities of interplanetary environments, such as very long delays (even up to hours), significant error rates, and link disruption or service unavailability [68], make the use of TCP very problematic in such an environment. On the other hand, an alternative protocol stack (defined from the network up to the application layer) to mitigate the performance impairments observed in such environment was proposed at the end of the 1990s through the SCPS body, that also developed a transport protocol called SCPS-TP [18], now standardized within CCSDS too.

In fact, SCPS-TP essentially builds on TCP, proposing some extensions, implemented as TCP options, capable to increase the robustness against link errors and long delays. In particular, the use of selective negative acknowledgment (SNACK) messages allows a faster response to link errors and lower delays than TCP in triggering the recovery mechanisms, where three duplicated acknowledgment or timeout expiration are required. In addition, different congestion control strategies are implemented, namely, Van Jacobson, Vegas, rate-based [69]–[71], which make SCPS-TP more flexible than TCP in both terrestrial and space environments [72]. In spite of these attracting advantages, this protocol, however, has been seldom applied in space missions, because of its TCP heritage which determines some degraded performance in case of very long-delay networks.

3) Bundle Protocol and Licklider Transmission Protocol: CCSDS is currently promoting the adoption of the DTN architecture [73]–[79] for the future space missions, by drawing a recommendation about the requirements of such missions in terms of delay and disruption tolerance [80]. The bundle protocol (BP) and Licklider transmission protocol (LTP) [81]–[83] are the core of the next-generation deep-space communication and for more general disrupted networking operations, where an architecture providing solid internetworking services in the Solar System Internet (SSI) will be required. To this end, some of the main functionalities already offered by CFDP (see next section) have been distributed over these two protocols, in order to have one overlay protocol offering the internetworking service and a different one working on a point-to-point basis offering reliability features, where needed. Hence, this separation allows offering internetworking capabilities not only to CFDP, but in general to all services running on top of BP. The general application of DTN architecture for deep-space networking is depicted in Fig. 4.

BP acts as overlay over the underlying protocol layers, thus allowing the implementation of dedicated protocols, depending on the characteristics of the applied environment. The philosophy of the BP is to decompose the whole network in an internet of internets [84], through which data delivery is performed by exploiting a store–carry–forward approach.

BP implements a message-switching service: Application data units (ADUs) are encapsulated in the BP data units (BPDUs), commonly referred to as bundles [85]. BP offers different encapsulation service towards the underlying layer, to ensure interoperability with different protocols, depending on the specific environment where the DTN architecture is applied. This is achieved by the so-called convergence layers, which allow BP to work over TCP [86], UDP, and LTP [87].

As emerged in the previous section, the use of TCP is not recommended over interplanetary networks, whereas UDP and LTP are more appropriate, although the former does not offer any delivery guarantee. The latter is specifically designed for deep space and therefore deserves a specific attention. Its essentials will be drawn in the last part of this section.

A particular attention has to be reserved to the reliability options which make BP a formidable solution to contrast link disruption and service unavailability. As already reported, store/forward functionality is implemented, thus providing BP with the possibility to suspend and then resume a message delivery when a link is again available or when the quality is no longer degraded. To this end, dedicated mass memories have to be available in order to store all the incoming bundles during the message transfer interruption phase. Besides, a set of notifications sent from the intermediate nodes or the destination to the source nodes, allow tracking the status of an ongoing message transfer, by eventually signaling the application layer whether the delivery was successful or failed.

An additional protection against bundle losses is represented by the custodial transfer option, through which BP exploits the "mailman" principle. Nodes elected custodians [88] (statically or through notifications) are responsible for

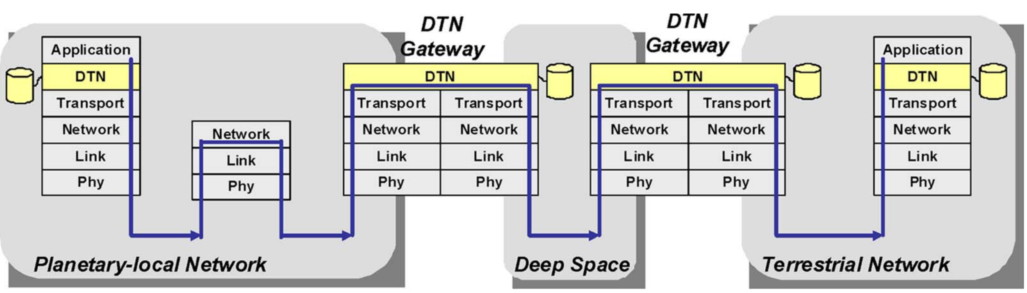

Fig. 4. *DTN architecture in deep-space scenarios.*

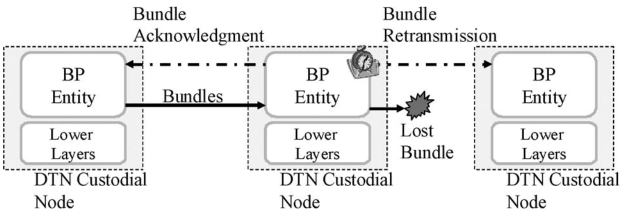

Fig. 5. ***Custodial transfer option and retransmission of a lost bundle upon bundle transmission timer expiration.***

forwarding bundles towards the next BP node. From a protocol implementation viewpoint, each custodian implements an ARQ mechanism to ensure that bundles are correctly routed to the next node. For each delivered bundle, an acknowledgment is sent back to notify the correct receipt of the bundle. In case of missed or incorrect delivery, a fail signal notification or a bundle timeout event occurs, forcing the retransmission of the missed bundles. The retransmission mechanism applied when the custodial transfer option is enabled is depicted in Fig. 5.

As far as the Licklider transmission protocol (LTP) is concerned, it is implemented beneath the bundle protocol and is essentially a point-to-point protocol. It was conceived for data transport over deep-space links in order to overcome the performance limitations introduced by TCP-based protocols because of long delays and large error ratios.

LTP data units are usually referred to as segments and they handle data blocks forwarded by BP. Depending on the service requirements or the quality of service demanded by each block, LTP defines red and green parts for a given block. The former corresponds to parts which require to be delivered reliably, whereas the latter may require delivery immediacy. In order to properly handle these two types of profiles, LTP implements a negative acknowledgment (NAK)-based recovery scheme for red parts, whereas no protection mechanisms are enabled for the green. In particular, when LTP segments carry red parts, the retransmission of the missing parts is performed upon reception of a negative acknowledgment issued by the receiving LTP entity. The missed reception of a retransmitted segment is detected at the receiving side through a NAK-timer expiration, triggering the issuance of a NAK to invoke a new retransmission loop.

4) CFDP: The CCSDS file delivery protocol (CFDP) [22] aims at transferring files from one file store to another, located in spacecraft and space stations [89]. It implements two operative procedures: core and extended. The former allows the data transfer between two topologically consecutive file stores, without any caching system in between. The latter instead allows the exchange of files through intermediate CFDP nodes, which implement store/forward capabilities, useful to suspend and then resume the data transfer in case of disruptions or scheduled transmission windows. The "extended procedure" configuration, though attracting for the inherent increased reliability, has been seldom used in space missions because of the rather complex implementation of CFDP, thus resulting in some performance degradation. By contrast, CFDP is today used as the file transfer protocol for space missions when configured in "core procedure," without relying on any suspend/resume functionality implemented within intermediate CFDP nodes, features which are instead provided by the DTN architecture.

A file to be transmitted is encoded into a file delivery unit (FDU), composed of the file itself and of metadata necessary for the data management. The CFDP entity splits the FDU into CFDP protocol data units (PDUs) of variable length. CFDP PDUs are structured into a payload, containing up to 65 536 B, and a header, containing CFDP source and destination identifiers, transfer file sequence number as well as other fields necessary to allow the reconstruction of the FDU at destination. Data transmission is performed by CFDP entities according to two operative modes: unacknowledged and acknowledged.

The unacknowledged mode implements no mechanisms to ensure a complete data delivery; communication reliability, where required, should be ensured by proper mechanisms implemented within the underlying layers. The acknowledged one provides reliable delivery of data by means of ARQ strategies, relying on NAKs. The detection of missing CFDP PDUs is performed by the receiver, which notifies the loss of data to the sender, by issuing NAKs, according to four different algorithms: immediate, deferred, asynchronous, and prompted. In the first case (see Fig. 6), a NAK issuance is performed as soon as the loss of CFDP PDUs is detected. Deferred mode allows postponing the issuance of NAKs to the end of the file transfer. As far as asynchronous and prompted modes are concerned, the detection of missing blocks is triggered by external events, such as explicit (asynchronous mode) or periodical (prompted mode) requests by the sender. In particular, the asynchronous mode is of interest in case human operators checked the effectiveness of the data transfer, thus triggering the CFDP recovery function in case of information loss. On the other hand, the periodic mode may be applied to verify the status of the file transfer at the beginning of a new transmission window, which is usually scheduled in advance.

The scientific community mostly focused on the case of immediate and deferred retransmissions [90], [91]. The application of the former looks particularly promising in the case of long delays and random losses. In this case, it is worth recovering from isolated losses as soon as they are detected and notified to the sender side. In case of correlated losses, the use of a deferred retransmission strategy looks more effective, as it is expected to avoid running consecutive recovery loops because of the loss of adjacent

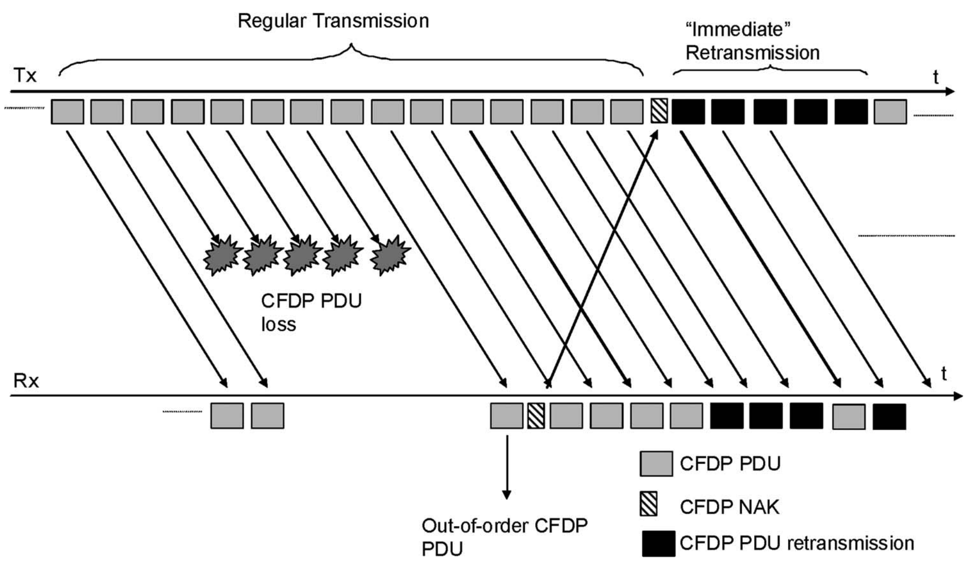

Fig. 6. ***CFDP applying immediate retransmission algorithm.***

blocks. Finally, when a short delay is experienced by the network both strategies are effective, since the feedback mechanism to transport NAK is fast enough to solicit the sender side to retransmit the missing CFDP PDUs with satisfactory performance achievements independently of the loss pattern (independent or correlated losses) [92], [93].

5) *Overall Summary:* As described in the previous sections, upper layer protocols provide different recovery tools, all based on ARQ principles. It was observed that retransmission functionality is optional in TC COP-1, CFDP, bundle protocol, and LTP, whereas TCP and SCPS-TP are implementing the well-known fast retransmission, fast recovery, and retransmission upon timeout procedures. Besides, it can be remarked that the TCP and SCPS-TP can be used mainly on short-delay links, in order to avoid length retransmission cycles, whereas the others have an overall applicability over deep-space networks. Finally, it is also worth noting that the interactions among the different recovery procedures have a great importance, making the use of some of them not mandatory, depending on profile of the service being handled over the interplanetary network.

An overall summary of the sketched protocol options is reported in Table 1.

Table 1 ARQ Mechanisms Summary

Protocol	Operating Mode	Retransmission Mechanism
TC COP-1	Sequence-controlled, expedited	Sequence-controlled implements go-back-n ARQ procedure. Expedited does not implement any recovery procedure
TCP	Not applicable	Fast Retransmit, Fast Recovery, timeout-based retransmission
SCPS-TP	Not applicable	Same as TCP with SNACK
BP	Store/ Forward, Custodial Transfer Option	suspend/resume facilities; Custodial Transfer
LTP	Red and Green parts	Red parts: NAK-based ARQ Green parts: no reliability guarantee
CFDP	Core, Extended procedures Unacknowledged/ Acknowledged/	Extended: store/forward Core: no store/forward Acknowledged: NAK-based ARQ with immediate, deferred, prompted and asynchronous algorithms. Unacknowledged: no reliability guarantee

C. Coding at the Networking and Application Layers

1) Main Principles: The traditional channel codes employed in space links (and reviewed in Section III-A), together with data link layer frame validation procedures, only deliver to the upper layers those data units for which integrity is recognized. The discarding of frames marked as "bad" at the receiver side is perceived as data units losses by the upper layers. This phenomenon is typically caused by the noise affecting the transmitted codewords, as well as brief outage conditions due to weather, loss of frame synchronization, changes on the fly of the channel code rate or the modulation, or turbulences of the propagation medium in free-space optical links. An effective channel model from the upper layers perspective is then a packet erasure channel (PEC), where whole packets of bits are either correctly received or lost. Packet-oriented erasure correcting codes may be adopted to enable an automatic recovery, on the receiver side, of the missing data units.

A set of k fixed-length input segments, each of length T bytes, is encoded to obtain n encoded segments, each again of length T bytes. The encoded segments are composed of the k input segments followed by $m = n - k$ checksum segments. In the receiver, the n encoded segments, affected by some erasure pattern, are processed by an erasure decoder to recover the k input segments. The implementation of a packet-oriented code at some of the upper layers is not aimed at replacing the bit-oriented channel code, i.e., the two coding schemes can coexist in the same system.

The aforementioned k input segments are obtained from a certain number of PDUs belonging to the protocol stack layer in which the erasure code is implemented. These PDUs are known as source packets. The encoding process starts by filling with the source packets an encoding table, also called the source block, consisting of $n = k + m$ rows each of T bytes, indexed from 1 to n. The first k source block rows are filled with the source packets (which may be of either constant or variable length) in a progressive order. Usually, only the payload of a source packet is introduced into the source block, together with a few additional data necessary on the decoder side (usually, the length of the payload being inserted and the flow of PDUs to which the packet belongs[4]). Each source packet occupies a certain number of rows of the source block, where the last row associated with a source packet is completed by padding bits if needed. Each of the first k rows of the source block, of length T bytes, corresponds to an input segment. It is regarded as a sequence of $L = 8T/\log_2(q)$ symbols of a finite field F_q of order q, where q is assumed to be a power of 2.

After the first k source block rows have been filled with a certain number of source packets, encoding is performed to generate $m = n - k$ checksum segments.[5] In more detail, each checksum segment is calculated as a symbol-wise linear combination (in F_q) of a subset of the k input segments. (For example, if the erasure code is constructed over F_2, each checksum segment is calculated as a bit-wise exclusive-OR of a subset of the k input segments.) Once the checksum segments have been generated, a set of repair packets is formed. A repair packet is a PDU belonging to the same layer of the protocol stack to which source packets belong, and its payload is composed of a certain number of checksum segments. (The number of checksum segments forming the payload of a repair packet is a design parameter.) Repair packets are then transmitted, as well as source packets, by encapsulating them into the payload of lower layer PDUs. Each source/repair packet carries the information about the source block it is associated with (since several source blocks may exist in parallel), about the index of the first row of the source block it occupies, and about its length. This signaling is usually transmitted through a dedicated header.

Each source block in the transmitter corresponds to a decoding table in the receiver, which has the same number n of rows and the same width (T bytes) as the source block. When a source packet or a repair packet is received, its payload is inserted in the corresponding decoding table, starting from the row whose index is available in the packet. Note that the correctly received source packets can be also directly forwarded to the upper layers, besides being inserted in the decoding table. Due to the aforementioned impairments, some source or repair packets may be not delivered, so that they are missing in the decoding table. Decoding starts upon a time trigger, and its goal is to recover as many input segments as possible. It is successful when all the unknown rows out of the first k rows of the decoding table have been recovered. A decoding failure is declared otherwise.

A study group called CCSDS Long Erasure Codes (LEC)/Birth of Feather (BOF) was established in 2003 to investigate the potential benefits of erasure correcting schemes in CCSDS space exploration missions [43], [94].

The simplest reference channel model used for the design of powerful erasure correcting codes is the

[4]For example, if the erasure code is implemented at SPP level over TM, TC, or AOS SDLP, source and repair packets are space packets and this flow identifier is the identifier of a virtual channel. If the implementation is at UDP layer, this would be the UDP port, and so on.

[5]Note that encoding is not necessarily performed after the first k source block rows have been completely filled. For example, if no additional source packets are available and there are delay constraints, encoding is imposed by a time trigger. In this case, the rows of the source block out of the first k rows, which have not yet been filled, are set to all-zero. This is equivalent to *shortening* of the erasure code, and the information about which rows have been filled with source packets and which ones have been filled with padding must be made available at the receiver (for instance, by means of a dedicated in-band signaling).

memory-less (packet) erasure channel, where each packet is lost independently from the others with probability ε. The limit rate R at which packets can be sent over this channel with vanishing error probability, is the Shannon capacity $C = 1 - \varepsilon$ (data packets/codeword packets).

Consider an (n, k) linear block erasure code generating n encoded segments from k input segments, where $n = k + m$ and $m > 0$. Typically, the set of n encoded segments contains of the k input segments, i.e., the erasure code is systematic. In such a case, the codeword vector $\mathbf{x}$ can be partitioned as $\mathbf{x} = [\mathbf{u}|\mathbf{p}]$, where $\mathbf{u}$ is the vector of the k input segments, and $\mathbf{p}$ is the vector of the m checksum segments. The code rate R is here given by k/n. The code parity-check matrix $\mathbf{H}$ is an $(m \times n)$ matrix with elements in F_q, which defines a set of m parity-check equations that must be fulfilled by the encoded segments composing the codeword. Specifically, we have

$$\mathbf{x}\mathbf{H}^T = \mathbf{0} \tag{5}$$

where $\mathbf{0}$ is a vector made by null (all-zero) packets. The linear constraints imposed by the code parity-check matrix are exploited at the decoder side to perform erasure recovery. More in detail, erasure decoding is performed by solving the system of equations

$$\mathbf{x}_{\bar{K}}\mathbf{H}_{\bar{K}}^T = \mathbf{x}_K\mathbf{H}_K^T. \tag{6}$$

Here, $\mathbf{x}_{\bar{K}}$ denotes the vector of erased encoded segments (unknowns), and $\mathbf{x}_K$ the vector of nonerased encoded segments. Analogously, $\mathbf{H}_K$ is the matrix composed of the columns of $\mathbf{H}$ corresponding to $\mathbf{x}_K$, while $\mathbf{H}_{\bar{K}}$ is the matrix composed of the columns of $\mathbf{H}$ corresponding to $\mathbf{x}_{\bar{K}}$.

Many decoding algorithms attempt to solve the equation system (6) with different approaches and with a different complexity. Among them, ML decoding consists of solving the system by Gaussian–Jordan elimination (GJE) performed on the matrix $\mathbf{H}_{\bar{K}}$. ML decoding offers the best possible performance, but its complexity is in general cubic with the dimension of the system, so that the overall complexity is $O(n^3)$. Hence, it becomes impractical for large block lengths, whereas it is known that the channel capacity can be approached by using long linear block codes.

For this reason, most of the attempts in the design of long efficient erasure correcting codes led to the classes of linear block codes with sparse parity-check matrices, for which the solution of (6) can be tackled iteratively, with linear $O(n)$ complexity [95], [96]. In particular, iterative (IT) decoding of LDPC codes consists of solving (6) by recursively processing one equation at time. For large block lengths, effective LDPC code design techniques, for example, based on extrinsic information transfer (EXIT) charts, are available, allowing the design of low-complexity, long codes with near-optimum performance [96]–[101]. Some classes of binary LDPC code ensembles under IT decoding have been shown to be able to asymptotically approach with an arbitrarily small gap the memory-less erasure channel capacity [102]–[104].

Some problems, however, arise when constructing moderate (and practical) length LDPC codes (i.e., codes with $n < 10\,000$) according to the asymptotically optimal ensembles proposed so far. Owing to the suboptimality of IT decoding, at high error rates the performance curve, although usually good, denotes a coding gain loss with respect to that of the same code under ML decoding [105], [106]. Moreover, at lower error rates the performance curve typically exhibits a high error floor caused by the presence of small size stopping sets [107]. In general, lowering the IT error floor implies a sacrifice in terms of coding gain at high error rates. To counteract this issue, efficient ML decoding algorithms have been developed for LDPC codes, which exploit their parity-check matrix sparseness [108], [109]. Thanks to these reduced-complexity ML approaches, decoding speeds above 1 Gb/s have been demonstrated on satellite links for block sizes in the order of a few thousands symbols [109].

In [105] and [106], LDPC code design techniques are provided, which target simultaneously near-optimum performance and low ML decoding complexity. In Fig. 7, the performance of a rate-1/2 (2048,1024) irregular repeat accumulate (IRA) code [110] from [109] is provided, in terms of codeword error rate (CER) versus channel erasure probability. The performance is provided for both IT and ML decoding, and is compared with the Singleton bound, which lower bounds the block error probability of an (n, k)

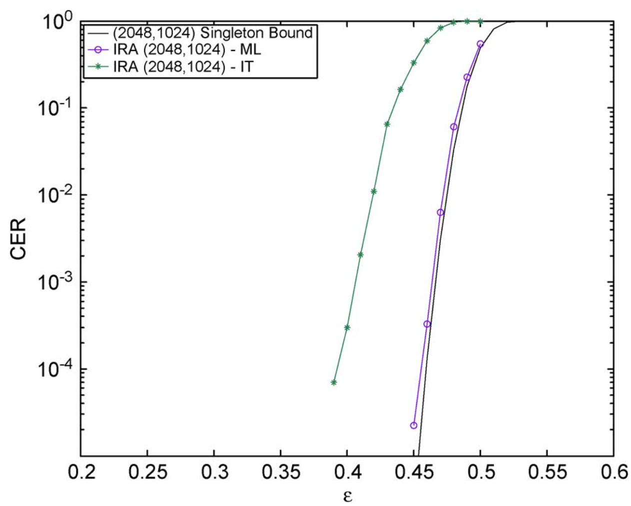

Fig. 7. ***Performance in terms of codeword error rate (CER) for a (2048,1024) IRA code from [109].***

linear block code ($n = 2048$ and $k = 1024$, in the example) according to

$$P_B \geq \sum_{i=n-k+1}^{n} \binom{n}{i} \varepsilon^i (1-\varepsilon)^{n-i}. \tag{7}$$

The IRA code, under ML decoding, allows tightly approaching the Singleton bound, with an evident gain with respect to the IT decoding performance. Note that, for this code, ML decoding speeds larger than 1.5 Gb/s have been recorded.

Similar performances are achieved on correlated erasure channels [111], [112], while extensions of the efficient ML approaches of [108] and [109] have been proposed for error-and-erasure channels (where errors are sporadic, i.e., much less frequent than the erasures) in [113].

Within CCSDS, three main erasure code solutions have been proposed and investigated until now, i.e.:

1) the above-presented solution, based on families of LDPC codes designed for ML decoding [114];
2) a class of protograph-based LDPC codes [115], [116], designed for conventional IT decoding;
3) a scheme based on interleaving of short Reed–Solomon codes [117].

The last solution is suited to the case where erasures take place in bursts, and shows nonnegligible losses either with uncorrelated erasures or with channels with mixtures of bursty and sparse erasure patterns. The first two LDPC-based solutions allow operating close to the theoretical limits in presence of both correlated and uncorrelated erasures.

2) Integration Within the CCSDS Protocol Stack: The advantages offered by erasure codes, pointed out in previous section, support the idea of implementing a packet-layer coding strategy within the CCSDS protocol stack. The use of erasure codes in space communications could be beneficial to recover from frame losses that otherwise would trigger long retransmission periods performed by the CFDP entity, thus penalizing the overall system performance.

Four possible coding strategies can be taken in consideration: 1) pure FEC, 2) type-I hybrid ARQ, 3) type-II hybrid ARQ, and 4) weather genie [94]. The first one consists in the generation and transmission of information and redundancy units over the forward link. Solutions 2) and 3) combine advantages of FEC and ARQ strategies: type-I hybrid ARQ allows retransmitting the information symbols that could not be recovered at the destination through erasure decoding; type-II hybrid ARQ consists in sending additional redundancy symbols upon notification of failed erasure decoding at the receiver side. Weather genie approach exploits the availability of a return channel to acquire information about the deep channel state and to adapt the coding strategy accordingly.

It is immediate to see that some challenges can arise particularly for solutions 3) and 4), in virtue of the need for a return channel and for a protocol specifically designed to use it. Hybrid ARQ-II demands for a dedicated protocol such that the receiver side application can request additional redundancy symbols. Likewise, weather genie requires a dedicated protocol able to estimate the channel state and to transmit it to the sender side. On the other hand, type-I hybrid ARQ, though demanding for the return link as solutions 3) and 4), can be implemented within layer protocols that already implement retransmission procedures (e.g., CFDP, BP, LTP) to recover from information losses.

According to these observations and taking into account the limited implementation complexity allowed on space nodes, solutions 3) and 4), though attractive, look harder to be adopted in deep-space environments. In particular solution 3) will be partially addressed in Section III-C2 as part of the possible protocol extensions in terms of erasure coding. In fact, given the high complexity of space mission design and the necessity to avoid a burdensome management of the space operations, the introduction of new advanced features in preexisting protocols will be performed gradually and carefully. As a result, in the space mission perspective it is desirable to have simple and effective protocol implementations, thus making the adoption of solutions 3) and 4) not appealing at the moment. On the other hand, solutions 1) and 2) easily meet the technological requirements of space nodes.

Another important issue to be addressed is to position the implementation of a given packet-level coding within the protocol stack, by identifying the most suitable protocol layer.

Actually, this approach can be applied at different layers of the protocol stack, from the application down to the network layer, where actually a "packet" unit may be defined [113].

- *Application/transport layer erasure coding*. It is applied on end-to-end basis: The coding strategy can be configured according to the content carried by data packets and to the error protection they may need. This approach allows keeping unmodified the underlying protocol stack, offering several advantages in terms of flexibility and modularity of the whole deep space communication system design.
- *Network layer erasure coding* [119]. It works on a point-to-point basis, thus allowing contrasting efficiently packet erasures experienced with different loss patterns in a multihop environment. The main drawback is represented by the necessity to modify the different network layer protocol specifications that may be present on the network segments, depending on the space missions. This may be too burdensome from an implementation point of view.

CCSDS has dedicated some effort in identifying the most appropriate protocol layer where the implementation of erasure codes would be beneficial. According to the observations raised above, it is possible to consider CCSDS SPP, LTP, BP, and CFDP the most appropriate ones.

In particular, by taking under consideration that different protocol alternatives are available at the network layer (CCSDS SPP, IPv4, and v6), it is difficult to propose the implementation of erasure codes for a given protocol, whose use generally depends on the specific space mission. On the other hand, the implementation of a packet-level coding at the application layer, namely within the CCSDS file delivery protocol [115], has some pros and cons. Concerning the advantages, it is possible to select the most appropriate coding strategy according to the file transfer characteristics, thus matching quality-of-service requirements. For instance, it is very likely that transmission of images could require a reduced level of redundancy, as images are usually already encoded at the application layer with a sufficient level of protection. On the other hand, data file transfer could call for additional reliability options, which will be then provided by the implementation of erasure codes within CFDP. Such an approach introduces some limitations on the usual benefits of erasure codes. Actually, underlying protocol layers (e.g., bundle protocol and Licklider transmission protocol) already implement recovery functions, which definitely help reduce the probability of CFDP PDU erasure. In this respect, it is immediate to see that the use of erasure codes in this context would increase the CFDP implementation complexity, with minor gains from a performance point of view, unless interlayer optimization of the defined recovery function is carried out.

As far as bundle protocol is concerned, it would be possible to think about dedicated protocol extensions to allow the integration of erasure codes at this layer [116]. As already observed, the main advantage here stems from the possibility to offer increased robustness on hop-to-hop basis, according to the philosophy of the DTN overlay architecture. However, it is worth remarking that the bundle protocol already implements an ARQ scheme (once the custodial transfer option is enabled) and a store/forward mechanism, whereby the additional implementation of erasure coding functionalities should be carried out very attentively. An additional challenge to be properly addressed is the buffer management [122] at this layer that should take into account the memory space required for the retransmission operations (in case of custodial transfer), store/forward, and erasure encoding. It has been observed [123] that the interaction between these functions cannot be neglected, as it might give rise to congestion events because of bundle layer buffer overflow. In this perspective, a careful design of the overall protocol extensions has to be carried out by also bearing in mind the present spacecraft constraints in terms of onboard storage.

Finally, the Licklider transmission protocol is probably the best candidate [124] for the integration of erasure codes, since it is the "first" protocol layer above the CCSDS datalink layer, where recovery functions are implemented, thus complementing the protection mechanisms provided by channel coding at lower layers. As illustrated in the previous sections, LTP already implements an ARQ mechanism based on negative acknowledgments, enabled in case of red LTP blocks. Nevertheless, there are scenarios where the uplink is very much bandwidth constrained, thus possibly making the transport of acknowledgment a bit problematic. Actually, a limited data-rate on the uplink would result in much delayed delivery of acknowledgments to the sender side, then provoking an increased latency in the recovery operations. This is particularly evident in the case of the International Space Station (ISS) [125], [126], where the uplink can currently offer up to few hundred bits per second, thus degrading the overall system performance. Consequently, the implementation of erasure codes within LTP would be helpful and also feasible with reduced implementation complexity, by exploiting the protocol extensions. Besides, in case of failed erasure decoding, the data communication reliability would be ensured by the ARQ available directly within the LTP protocol or even at the bundle protocol.

3) Advances: Erasure Coding With ARQ: The only use of erasure codes can be sometimes not sufficient to ensure the complete delivery of data because of strongly degraded signal quality. In this case, it is worth thinking about additional recovery mechanisms which can be provided in terms of ARQ. The joint combination of erasure codes and ARQ mechanisms can be the key solution to enable reliable data delivery. This is important in scenarios where the presence of long delays and link errors makes the disjoint use of ARQ and erasure codes not really effective. On the contrary, the employment of a two-level protection scheme, where retransmission procedures help recover the information losses not handled by the erasure codes is a promising solution. These considerations reinforce the interest towards combined packet-level coding and ARQ mechanisms. Nevertheless, it has to be also observed that the interaction between erasure codes and other recovery mechanisms implemented within different protocol layer cannot be neglected, but on the contrary deserves some attention.

Different approaches can be pursued in this context, depending on the system design constraints. Actually, the receiver upon decoding-fail detection can request to the sender additional redundancy "packets" to help complete the decoding procedure. The redundancy packets can be generated "on-fly" in case of CFDP, where a copy of a file is stored unless an acknowledgment is received. Another approach more suitable to the cases where persistent memory storage is not available consists in keeping only a copy of the redundancy "packets."

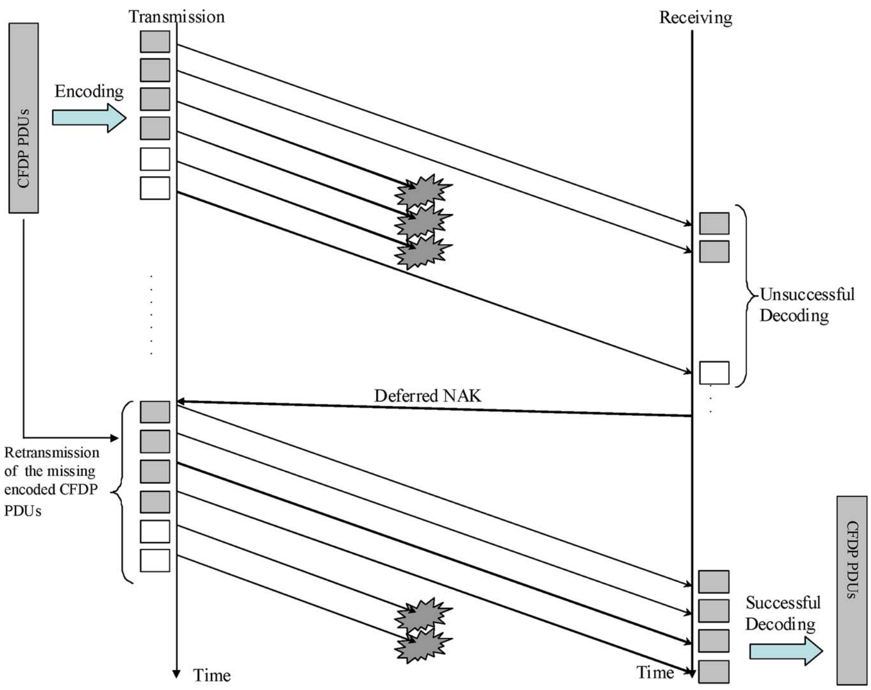

Fig. 8. ***Integration of erasure codes into CFDP, with deferred retransmission.***

Alternatively, the retransmission of the set of packets can be requested, in order to increase the likelihood of correct delivery. The second approach has been already proposed in the literature for the use within CFDP [127], [128], whereas the former was investigated in the context of the bundle protocol, to understand the interaction between erasure codes and the custodial transfer option [123]. In the first case, CFDP running in acknowledged mode with deferred retransmission algorithm was analyzed. CFDP PDUs are aggregated together in a data unit and then segmented in small encoding data units, which are used in the packet-level coding procedure. At destination, once the decoding procedure cannot complete because insufficient number of received encoded data units, a negative acknowledgment is issued to solicit the sender to retransmit the whole aggregate of CFDP PDUs together with the redundancy units. The case of CFDP integrating erasure codes with deferred retransmission is sketched in Fig. 8. On the other hand, in the case of bundle protocol, the interaction of erasure codes and the custodial transfer option has to be investigated. The encoding procedure works on information bundles, generating a number of redundancy bundles. The destination upon failed decoding operation issues an acknowledgment to request the sender to transmit the missing redundancy bundles. In addition, the custody transfer mechanism is upgraded so that retransmission timers now refer to the whole set of information and redundancy bundles, differently from the protocol specification where a retransmission timer is set for each single bundle. The overall protocol behavior is depicted in Fig. 9.

IV. CONCLUSION AND FUTURE TRENDS

This paper has surveyed the communication and networking protocols currently available from the CCSDS protocol stack, with particular attention to the reliability options. Different features are implemented from the lower up to the application layer. In particular, the channel coding schemes implemented within CCSDS TM, TC, AOS, and proximity-1 have been illustrated, by also shedding some lights on the CRC issues, which may play some issues in the recovery capabilities implemented in deep-space missions. In addition to channel coding mechanisms, ARQ schemes have been also illustrated by considering the relevant cases of TC COP-1, BP, LTP, and CFDP protocols. A special note has been reserved to the application of

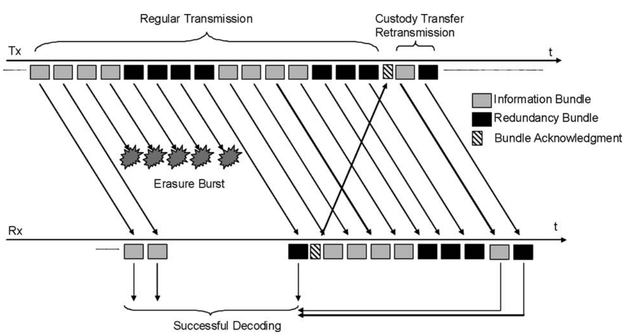

Fig. 9. ***Integration of erasure codes into bundle protocol, with custodial transfer option enabled.***

erasure codes, which could play an important role in the future deep-space communications in order to provide complimentary protection tools implemented at the higher layers. It is remarkable that such an approach would be beneficial in space missions characterized by very much resource-constrained uplink, or in case of optical link communications [129], where short fading events (1–100 ms) can be contrasted by the use of erasure codes. Finally, the ultimate frontier for ensuring reliable data delivery is represented by the joint use of ARQ and erasure codes, in order to provide a two-level protection against short and moderately long fading events, by also taking advantage of the store/forward capabilities offered by the higher protocol layers (e.g., bundle protocol). ■

Acknowledgment

The authors would like to thank J. L. Gerner, former Consultative Committee for Space Data Systems (CCSDS) SLS Area Director, for the coordination of channel coding activities, and E. Vassallo for the suggestions to improve the quality of the paper. E. Paolini and G. Liva would like to also thank Prof. M. Chiani for introducing them to channel coding, for his precious teachings, and constant guidance.

REFERENCES

[1] *Consultative Committee for Space Data Systems (CCSDS)*. [Online]. Available: http://www.ccsds.org

[2] M. A. Allen, "Management and control of CCSDS cross-support services," *Proc. IEEE*, vol. 78, no. 7, pp. 1304–1310, Jul. 1990.

[3] *Overview of Space Link Protocols*, Consultative Committee for Space Data Systems (CCSDS), Washington, DC, Report Concerning Space Data System Standards, 130.0-G-2. Green Book. Issue 2, Dec. 2007.

[4] R. D. Carper and W. H. Stallings, III, "CCSDS telemetry systems experience at the Goddard space flight center," *IEEE Netw.*, vol. 4, no. 5, pp. 17–21, Sep. 1990.

[5] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1138–1144, Aug. 1990.

[6] A. J. Hooke, "CCSDS advanced orbiting systems: International data communications standards for the space station freedom," *IEEE Netw.*, vol. 4, no. 5, pp. 13–16, Sep. 1990.

[7] "International space station (ISS)—A star is born," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 15, no. 10, pp. 84–88, Oct. 2000.

[8] *TM Space Data Link Protocol*, Recommendation for Space Data Systems Standards, 132.0-B-1. Blue Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Sep. 2003.

[9] *TC Space Data Link Protocol*, Recommendation for Space Data System Standards, 232.0-B-2. Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Sep. 2010.

[10] *AOS Space Data Link Protocol*, Recommended Standard, 732.0-B-2. Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Jul. 2006.

[11] *TM Synchronization and Channel Coding*, Recommendation for Space Data System Standards, 131.0-B-1. Blue Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Sep. 2003.

[12] *TC Synchronization and Channel Coding*, Recommendation for Space data System Standards, 231.0-B-2. Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Sep. 2010.

[13] *Radio Frequency and Modulation Systems—Part 1: Earth Stations and Spacecraft*, Recommended Standard, 401.0-B. Blue Book, Issue 20, Consultative Committee for Space Data Systems (CCSDS), Apr. 2009.

[14] *Space Packet Protocol*, Recommendation for Space Data System Standards, 133.0-B-1, Blue Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Sep. 2003.

[15] *Proximity-1 Space Link Protocol—Data Link Layer*, Recommended Standard, 211.0-B-4. Blue Book, Issue 4, Consultative Committee for Space Data Systems (CCSDS), Jul. 2006.

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

[17] *Proximity-1 Space Link Protocol—Physical Layer*, Recommended Standard, 211.1-B-3. Blue Book, Issue 3, Consultative Committee for Space Data Systems (CCSDS), Mar. 2006.

[18] *Space Communications Protocol Specification (SCPS)—Transport Protocol (SCPS-TP)*, Recommended Standard, 714.0-B-2. Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Oct. 2006.

[19] *Space Communications Protocol Specification (SCPS)—Network Protocol*, Retired Standard, 713.0-B-1-S. Silver Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), May 1999.

[20] *Space Communications Protocol Standards (SCPS)—Security Protocol (SCPS-SP)*, Retired Standard, 713.5-B-1-S. Silver Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), May 1999.

[21] *Space Communications Protocol Specification (SCPS)—File Protocol (SCPS-FP)*, Retired Standard, 717.0-B-1-S. Silver Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), May 1999.

[22] *CCSDS File Delivery Protocol (CFDP)*, Recommended Standard, 727.0-B-4. Blue Book, Issue 4, Consultative Committee for Space Data Systems (CCSDS), Jan. 2007.

[23] J. Postel, *Transmission Control Protocol*, STD 7, [RFC 793], Sep. 1981.

[24] J. Postel, *User datagram protocol*, STD 6, [RFC 768], Aug. 1990.

[25] J. Postel, *Internet Protocol*, IETF, Request for Comments, RFC 791, Sep. 1981.

[26] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, IETF, Request for Comments, RFC 2460, Dec. 1998.

[27] *IP Over CCSDS Space Links*, Draft Recommended Practice for Space Data System Standards, 702.1-R-4. Red Book, Issue 4, Consultative Committee for Space Data Systems (CCSDS), Jul. 2010.

[28] *Space Link Identifiers*, Recommendation for Space Data System Standards, 135.0-B-4. Blue Book, Issue 4, Consultative Committee for Space Data Systems (CCSDS), Oct. 2009.

[29] Space Internetworking Strategy Group, "Recommendations on a strategy for space internetworking," Tech. Rep., Oct. 2009.

[30] V. Cerf, S. Burleigh, A. Hooke, L. Torgersen, R. Durst, K. Scott, K. Fall, and H. Weiss "Delay-tolerant networking architecture," IETF RFC 4838, Apr. 2007.

[31] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[32] A. J. Barbieri, S. Butman, M. J. Danos, E. Greenberg, P. A. Ilott, G. J. Kazz, J. L. Torgerson, A. Vaisnys, W. R. Adams, C. E. Johnson, M. Dapore, and D. Merz, "Development and flight performance of CCSDS Proximity-1 on Odyssey and the Mars exploration rovers," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 2005, pp. 1444–1454.

[33] J. Taylor, A. Makovsky, A. Barbieri, R. Tung, P. Estabrook, and A. G. Thomas "Mars exploration rover telecommunications," DESCANSO Design & Performance Summary Series.

[34] *Bandwidth-Efficient Modulations: Summary of Definition, Implementation, and Performance*, Report Concerning Space Data System Standards, 413.0-G-2. Green Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Oct. 2009.

[35] *Communications Operation Procedure-1*, Recommended Standard, 232.1-B-2. Blue Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Sep. 2010.

[36] *Internet Engineering Task Force (IETF)*. [Online]. Available: http//www.ietf.org

[37] *Internet Research Task Force (IRTF)*. [Online]. Available: http://www.irtf.org

[38] J. Postel and J. Reynolds, "File transfer protocol (FTP)," IETF, Request for Comments, RFC 959, Oct. 1985.

[39] *Asynchronous Message Service (AMS)*, Draft Recommended Standard, 735.1-R-2. Red Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Jul. 2009.

[40] *TM Synchronization and Channel Coding—Summary of Concept and Rationale*, Informational Rep., CCSDS 130.1-G-1. Green Book. Issue 1, Consultative Committee for Space Data Systems (CCSDS), Jun. 2006.

[41] K. Andrews, D. Divsalar, S. Dolinar, J. Hamkins, C. R. Jones, and F. Pollara, "The development of turbo and LDPC codes for deep-space applications," *Proc. IEEE*, vol. 95, no. 11, pp. 2142–2156, Nov. 2007.

[42] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[43] G. P. Calzolari, M. Chiani, F. Chiaraluce, R. Garello, and E. Paolini, "Channel coding for future space missions: New requirements and trends," *Proc. IEEE*, vol. 95, no. 11, pp. 2157–2170, Nov. 2007.

[44] *Low-Density Parity-Check Codes for Use in Near-Earth and Deep-Space Applications*, Experimental Specification, CCSDS 131.1-O-2. Orange Book, Issue 2, Consultative Committee for Space Data Systems (CCSDS), Sep. 2007.

[45] ESA Requirements and Standards Division, "Space data links—Telecommand protocols synchronization and channel coding," Noordwijk, The Netherlands, ECSS-E-ST-50-04C, 2008.

[46] *TC Synchronization and Channel Coding—Summary of Concepts and Rationale*, Informational Rep., CCSDS 230.1-G-1. Green Book. Issue 1, Consultative Committee for Space Data Systems (CCSDS), Jun. 2006.

[47] *Proximity-1 Space Link Protocol—Rationale, Architecture, and Scenarios*, Informational Rep., CCSDS 210.0-G-1. Green Book. Issue 1, Consultative Committee for Space Data Systems (CCSDS), Aug. 2007.

[48] J. Hamkins and M. K. Simon, *Autonomous Software-Defined Radio Receivers for Deep Space Applications*. New York: Wiley, 2006.

[49] D. Divsalar, S. Dolinar, and C. Jones, "Short protograph-based LDPC codes," in *Proc. IEEE Military Commun. Conf.*, Orlando, FL, Oct. 2007, pp. 1–6.

[50] C. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell Syst. Tech. J.*, vol. 38, pp. 611–656, 1959.

[51] A. Valembois and M. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 796–810, May 2004.

[52] C. Poulliat, M. Fossorier, and D. Declercq, "Design of regular $(2, d_c)$-LDPC codes over $GF(q)$ using their binary images," *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, Oct. 2008.

[53] L. Costantini, B. Matuz, G. Liva, E. Paolini, and M. Chiani, "On the performance of moderate-length non-binary LDPC codes for space communications," in *Proc. 5th Adv. Satellite Multimedia Syst. Conf.*, Cagliari, Italy, Sep. 2010, pp. 122–126.

[54] D. Declercq and M. Fossorier, "Decoding algorithms for non-binary LDPC codes over $GF(q)$," *IEEE Trans. Commun.*, vol. 55, no. 4, pp. 633–643, Apr. 2007.

[55] G. D. Forney, Jr., "Transforms and groups," in *Codes, Curves and Signals: Common Threads in Communications*, A. Vardy, Ed. Norwell, MA: Kluwer, 1998, pp. 79–97.

[56] S. Dolinar, K. Andrews, F. Pollara, and D. Divsalar, "Bounds on error probability of block codes with bounded-angle maximum-likelihood incomplete decoding," in *Proc. Int. Symp. Inf. Theory Appl.*, Auckland, New Zealand, Dec. 2008, pp. 634–639.

[57] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," *Proc. IRE*, vol. 49, no. 1, pp. 228–235, Jan. 1961.

[58] T. G. Berry, "A note on the modified CRC," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 20, no. 5, pp. 12–17, Oct. 1990.

[59] D. Fiorini, M. Chiani, V. Tralli, and C. Salati, "Can we trust in HDLC?" *ACM SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 61–80, Oct. 1994.

[60] V. K. Agarwal and A. Ivanov, "Computing the probability of undetected error for shortened cyclic codes," *IEEE Trans. Commun.*, vol. 40, no. 3, pp. 494–499, Mar. 1992.

[61] J. K. Wolf and D. Chun, "The single burst error detection performance of binary cyclic codes," *IEEE Trans. Commun.*, vol. 42, no. 4, pp. 11–13, Jan. 1994.

[62] P. Fire, "A class of multiple-error-correcting binary codes for non-independent errors," Sylvania Electric Products, Inc., Mountain View, CA, Rep. RSL-E-2, Mar. 1959.

[63] D. Baum and L. Lenzini, "Analysis of command operation procedures (COPs) for reliable data transfer in the European space data network," *IEEE J. Sel. Areas Commun.*, vol. 9, no. 2, pp. 233–240, Feb. 1991.

[64] *Space Data Link Protocols—Summary of Concept and Cationale*, Informational Rep., CCSDS 130.2-G.1, Green Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Dec. 2007.

[65] Ö. B. Akan, J. Fang, and I. F. Akyildiz, "Performance of TCP protocols in deep space communication networks," *IEEE Commun. Lett.*, vol. 6, no. 11, pp. 478–480, Nov. 2002.

[66] C. Liu and E. Modiano, "On the performance of additive increase multiplicative decrease (AIMD) protocols in hybrid space-terrestrial networks," *Comput. Netw.*, vol. 47, no. 5, pp. 661–678, Apr. 2005.

[67] T. de Cola and M. Marchese, "Performance analysis of data transfer protocols over space communications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 4, pp. 1200–1223, Dec. 2005.

[68] I. F. Akyildiz, Ö. B. Akan, C. Chen, J. Fang, and W. Su, "The state of the art in interplanetary internet," *IEEE Commun. Mag.*, vol. 42, no. 7, pp. 108–118, Jul. 2004.

[69] J. Muhonen and R. C. Durst, "Performance of transport protocols over satellite communication links," in *Proc. IEEE Military Commun. Conf.*, Oct. 1998, pp. 263–269.

[70] R. Wang, S. Horan, S. Kota, and B. Sun, "An experimental evaluation of SCPS-TP over lossy GEO-space links," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, Nov./Dec. 2006, vol. 1, pp. 4294–4298.

[71] F. J. Lawas-Grodek, D. T. Tran, R. P. Dimond, and W. D. Ivancic, "SCPS-TP, TCP and rate-based protocol evaluation for high delay, error prone links," in *Proc. AIAA Int. Conf. Space Oper.*, Houston, TX, Jun. 2002, pp. 1–10.

[72] R. Wang and S. Horan, "Protocol testing of SCPS-TP over NASA's ACTS asymmetric links," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 790–798, Jun. 2009.

[73] K. Scott, "Space internetworking with IP and DTN," in *Proc. AIAA Int. Conf. Space Oper.*, Heidelberg, Germany, May 2008, pp. 1–8.

[74] G. P. Calzolari, W. Hell, P. Maldari, M. Schmidt, K.-J. Schulz, and C. Taylor, "Towards space internetworking: The ESA SISG-team view," in *Proc. AIAA Int. Conf. Space Operations*, Huntsville, AL, Apr. 2010, pp. 1–6.

[75] D. Israel, A. Hooke, K. Freeman, and J. Rush, "The NASA space communications data networking architecture," in *Proc. AIAA Int. Conf. Space Oper.*, Rome, Italy, Jun. 2006, pp. 1–9.

[76] E. Barkley, S. Burleigh, R. Gladden, S. Malhotra, and P. Shames, "Moving toward space internetworking via DTN: Its operational challenges, benefits and management," in *Proc. AIAA SpaceOps Conf.*, Huntsville, AL, Apr. 25–30, 2010, pp. 1–8.

[77] J. Schoolcraft, S. Burleigh, R. Jones, J. Wyatt, and L. Torgerson, "The deep impact network experiments—Concept, motivation and results," in *Proc. AIAA Int. Conf. Space Oper.*, Huntsville, AL, Apr. 2010, pp. 1–8.

[78] S. Fowell, D. Stanton, S. Farrell, O. Page, N. Peccia, and F. Flentge, "Study into the use of CFDP and DTN for future ESA missions," in *Proc. AIAA Int. Conf. Space Oper.*, Huntsville, AL, Apr. 2010, pp. 1–8.

[79] S. Burleigh, "Delay-tolerant networking for space flight operations: Design and development," in *Proc. AIAA Int. Conf. Space Oper.*, Heidelberg, Germany, May 2008, pp. 1–7.

[80] *Rationale, Scenarios, and Requirements for DTN in Space*, Informational Rep., CCSDS 130.2-G.1, Green Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS), Aug. 2010.

[81] K. Scott and S. Burleigh, "Bundle protocol specification," IETF RFC 5050, Nov. 2007.

[82] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider transmission protocol—Specification," IETF Request for Comments RFC 5326, Sep. 2008.

[83] R. Wang, T. Wang, and X. Wu, "Bundle protocol (BP) over Licklider transmission protocol (LTP) for cislunar communications," in *Proc. IEEE Global Telecommun. Conf.*, Honolulu, HI, Nov./Dec. 2009, pp. 1–5.

[84] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[85] K. Fall, W. Hong, and S. Madden, "Custody transfer for reliable delivery in delay tolerant networks," IRB-TR-03-030, Jul. 2003. [Online]. Available: http://www.dtnrg.org

[86] R. Wang, W. Xuan, W. Tiaotiao, L. Xueguan, and Z. Liulei, "TCP convergence layer-based operation of DTN for long-delay cislunar communications," *IEEE Syst. J.*, vol. 4, no. 3, pp. 385–395, Sep. 2010.

[87] S. Burleigh, "Delay-tolerant networking LTP convergence layer (LTPCL) adapter," draft-burleigh-dtnrg-ltpcl-02, IETF, Internet Draft, Feb. 2011.

[88] M. Demmer, E. Brewer, K. Fall, S. Jain, M. Ho, and R. Patra, "Implementing delay tolerant networking," IRB-TR-04-020, Dec. 28, 2004. [Online]. Available: http://www.dtnrg.org

[89] F. A. Sanders, G. Jones, and M. Levesque, "Transfer of files between the deep impact spacecrafts and the ground data system using CFDP: A case study," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 2007, pp. 1–5.

[90] W. Baek and D. C. Lee, "Analysis of CCSDS file delivery protocol: Immediate NAK mode," *IEEE Trans Aerosp Electron. Syst.*, vol. 41, no. 2, pp. 503–524, Apr. 2005.

[91] K. K. Choi, G. Maral, and R. Rumeau, "The implementation and validation of the new standard CCSDS file delivery protocol for multi-hopped space file transfer," in *Proc. IEEE Aerosp. Conf.*, Aspen, CO, Mar. 1999, pp. 153–163.

[92] R. Wang, B. L. Shrestha, W. Xuan, E. Tade, T. Wang, and X. Wang, "Experimental investigation of CCSDS file delivery protocol (CFDP) over cislunar communication links with intermittent connectivity," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 1910–1914.

[93] R. Wang, B. L. Shrestha, and X. Ma, "Channel delay impact on CCSDS file delivery protocol (CFDP) over space communications links," in *Proc. IEEE Int. Conf. Commun.*, Glasgow, Scotland, Jun. 2007, pp. 5201–5205.

[94] M. Chiani, G. Liva, and E. Paolini, "Long erasure correcting codes: A new appealing chance for space applications protocols?" presented at the Spring CCSDS Meeting, Montreal, QC, Canada, May 2004.

[95] J. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to reliable distribution of bulk data," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 8, pp. 1528–1540, Oct. 2002.

[96] M. Luby, M. Mitzenmacher, M. Shokrollahia, and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[97] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[98] T. Richardson and R. Urbanke, "The capacity of low-density parity check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[99] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inf. Theory*, vol. 48, no. 2, pp. 585–598, Feb. 2001.

[100] G. Liva, S. Song, L. Lan, Y. Zhang, S. Lin, and W. E. Ryan, "Design of LDPC codes: A survey and new results," *J. Commun. Softw. Syst.*, vol. 2, *Special Issue on Channel Coding in Wireless Systems*, no. 3, pp. 191–211, Sep. 2006.

[101] W. E. Ryan and S. Lin, *Channel Codes—Classical and Modern*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[102] P. Oswald and M. A. Shokrollahi, "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 12, pp. 364–373, Dec. 2002.

[103] H. Pfister, I. Sason, and R. Urbanke, "Capacity-achieving ensembles for the binary erasure channel with bounded complexity," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2352–2379, Jul. 2005.

[104] H. Pfister and I. Sason, "Accumulate-repeat-accumulate codes: Capacity-achieving ensembles of systematic codes for the erasure channel with bounded complexity," *IEEE Trans. Inf. Theory*, vol. 53, no. 6, pp. 2088–2115, Jun. 2007.

[105] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Generalized IRA erasure correcting codes for hybrid iterative/maximum likelihood decoding," *IEEE Commun. Lett.*, vol. 12, pp. 450–452, Jun. 2008.

[106] G. Liva, B. Matuz, E. Paolini, and M. Chiani, "Achieving a near-optimum erasure correction performance with low-complexity LDPC codes," *Int. J. Satell. Commun. Netw.*, vol. 28, no. 5–6, pp. 236–256, Sep.–Dec. 2010.

[107] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. Urbanke, "Finite length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.

[108] D. Burshtein and G. Miller, "An efficient maximum likelihood decoding of LDPC codes over the binary erasure channel," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2837–2844, Nov. 2004.

[109] B. Matuz, G. Liva, E. Paolini, and M. Chiani, "Pivoting algorithms for maximum likelihood decoding of LDPC codes over erasure channels," in *Proc. IEEE Global Telecommun. Conf.*, Honolulu, HI, Nov./Dec. 2009, pp. 1–6.

[110] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," in *Proc. 2nd Int. Symp. Turbo Codes Related Topics*, Brest, France, Sep. 2000, pp. 1–8.

[111] G. Liva, B. Matuz, Z. Katona, and E. Paolini, and M. Chiani, "On construction of moderate-length LDPC codes over correlated erasure channels," in *Proc. IEEE Int. Conf. Commun.*, Dresden, Germany, Jun. 2009, pp. 1–5.

[112] E. Paolini and M. Chiani, "Construction of near-optimum burst erasure correcting low-density parity-check codes," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1320–1328, May 2009.

[113] G. Liva, E. Paolini, B. Matuz, and M. Chiani, "A decoding algorithm for LDPC codes over erasure channels with sporadic errors," in *Proc. 48th Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 29–Oct. 1, 2010.

[114] G. Liva, T. De Cola, D. Giggenbach, N. Perlot, E. Paolini, M. Chiani, and G. P. Calzolari, "Packet loss recovery via maximum-likelihood LDPC decoding," presented at the CCSDS Fall Meeting, Berlin, Germany, Oct. 2008.

[115] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," JPL INP, Tech. Rep. 42-154, Aug. 2003.

[116] T. D. Divsalar, S. Dolinar, and C. Jones, "Protograph LDPC codes over burst erasure channels," in *Proc. IEEE Military Commun. Conf.*, Washington, DC, Oct. 2006, pp. 1–7.

[117] J. Hamkins, "Optimal codes for the burst erasure channel, IPN Progr. Rep. 42-174, Aug. 2008.

[118] T. de Cola and M. Marchese, "Reliable data delivery over deep space networks: Benefits of long erasure codes over ARQ strategies," *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 57–65, Apr. 2010.

[119] E. Paolini, M. Varrella, M. Chiani, and G. P. Calzolari, "Recovering from packet losses in CCSDS links," in *Proc. 4th Adv. Satellite Mobile Syst. Conf.*, Bologna, Italy, Aug. 2008, pp. 283–288.

[120] T. de Cola, H. Ernst, and M. Marchese, "Performance analysis of CCSDS file delivery protocol and erasure coding techniques in deep space environments," *Comput. Netw.*, vol. 51, no. 14, pp. 4032–4049, Oct. 2007.

[121] T. de Cola, "A protocol design for incorporating erasure codes within CCSDS: The case of DTN protocol architecture," in *Proc. 5th Adv. Satellite Multimedia Syst. Conf.*, Cagliari, Italy, Sep. 2010, pp. 68–73.

[122] S. Shambayati, "On selecting the size of a spacecraft's buffer," IPN PR 42-174, Aug. 2008, pp. 1–25.

[123] T. de Cola and M. Marchese, "Joint use of custody transfer and erasure codes in DTN space networks: Benefits and shortcomings," in *Proc. IEEE Globecom Conf.*, Miami, FL, Dec. 2010.

[124] T. de Cola, "Use of erasure codes in CCSDS upper layers: Motivation and implementation CCSDS Meeting, Portsmouth, VA, May 2004. [Online]. Available: http://cwe.ccsds.org/sls/docs/SLS-CandS

[125] K. Nichols, M. Holbrook, R. L. Pitts, K. K. Gifford, A. Jenkins, and S. Kuzminsky, "DTN implementation and utilization options on the international space station," in *Proc. AIAA SpaceOps Conf.*, Huntsville, AL, Apr. 25–30, 2010, pp. 1–8.

[126] E. Ackerman, "Interplanetary internet tested," *IEEE Spectrum*, vol. 46, no. 7, pp. 13–14, Jul. 2009.

[127] T. de Cola and M. Marchese, "High performance communication and navigation systems for interplanetary networks," *IEEE Syst. J.*, vol. 2, no. 1, pp. 104–113, Feb. 2008.

[128] T. de Cola and M. Marchese, "Study and performance analysis of ARQ-based and transport layer coding schemes over deep space networks," in *Proc. IEEE Globecom*, San Francisco, CA, Nov.–Dec. 2006, pp. 1–5.

[129] H. Hemmati, A. Biswas, and I. B. Djordjevic, "Deep-space optical communications: Future perspectives and applications," *Proc. IEEE*, to be published.

ABOUT THE AUTHORS

Tomaso de Cola (Student Member, IEEE) was born in Manosque, France, on April 28, 1977. He received the "Laurea" degree (with honors) in telecommunication engineering, in 2001, the Qualification degree as Professional Engineer in 2002, and the Ph. D. degree in Electronic and Computer Engineering, Robotics and Telecommunications in 2010 from the University of Genoa, Italy.

From 2002 until 2007, he worked with the Italian Consortium of Telecommunications (CNIT), University of Genoa Research Unit, as a Scientist Researcher. Since 2008, he has been with the German Aerospace Centre (DLR), Weßling, Germany, where he is involved in different European Projects focusing on different aspects of DVB standards, CCSDS protocols, and testbed design. He is coauthor of more than 30 papers, including in international conferences and journals. His main research activity concerns: TCP/IP protocols, satellite networks, transport protocols for wireless links, interplanetary networks as well as delay tolerant networks.

Dr. de Cola served on the Technical Program Committee at several IEEE International Conferences. He is a member of the IEEE Communications Society.

Enrico Paolini (Member, IEEE) received the Dr.Ing. degree (with honors) in telecommunications engineering and the Ph.D. degree in electrical engineering from the University of Bologna, Bologna, Italy, in 2003 and 2007, respectively.

While working toward the Ph.D. degree, he was a Visiting Research Scholar with the University of Hawaii at Manoa. From 2007 to 2010, he held a postdoctoral position with the Department of Electronics, Computer Science and Systems (DEIS), University of Bologna, where he is currently Assistant Professor. His research interests include error-control coding (with emphasis on LDPC codes and their generalizations, iterative decoding algorithms, reduced-complexity maximum-likelihood decoding for erasure channels), and radar sensor networks based on ultrawideband technology. In the field of error-correcting codes, has been involved since 2004 in several activities with the European Space Agency (ESA), in the framework of the Consultative Committee for Space Data Systems (CCSDS).

Dr. Paolini served on the Technical Program Committee at several IEEE International Conferences, and on the Organizing Committee of 2011 IEEE International Conference on Ultra-Wideband. He is a member of the IEEE Communications Society and of the IEEE Information Theory Society.

Gianluigi Liva (Member, IEEE) was born in Spilimbergo, Italy, on July 23, 1977. He received the Laurea degree in electronic engineering in and the Ph.D. degree from the Department of Electronics, Computer Science and Systems (DEIS), University of Bologna, Bologna, Italy, in 2002 and 2006, respectively.

He has been involved in the research of channel codes for high data rate Consultative Committee for Space Data Systems (CCSDS) missions since 2003, in collaboration with the European Space Operations Centre of the European Space Agency (ESA-ESOC). From October 2004 to April 2005, he was researching at the University of Arizona as visiting student, where he was involved in the design of low-complexity coding systems for space communication systems. He is currently with the Institute of Communications and Navigation, German Aerospace Center (DLR), Weßling, Germany. His main research interests include satellite communication systems, random access techniques, and error control coding.

Dr. Liva is active in the DVB-SH and in the DVB-RCS mobile standardization groups. In 2010 he has been appointed guest lecturer for channel coding at the Institute for Communications Engineering (LNT) of the Technische Universität München (TUM). He serves IEEE as reviewer for transactions, journals, and conferences.

Gian Paolo Calzolari was born in Latina, Italy, in 1958. He received the "Laurea in Ingegneria Elettronica" from the Università di Roma La Sapienza, Roma, Italy, in 1984.

He was a Software Engineer at DATAMAT Ingegneria dei Sistemi, Roma, Italy, from 1985 to 1986 and for ALCATEL-Face Standard Italia from 1986 to 1987. Since 1987, he has been with the European Space Agency (ESA) at the European Space Operations Centre (ESOC), Darmstadt, Germany. For several years he has been involved with definition of requirements, analysis, simulation, system design, and testing of the ESA ground stations equipment for the processing of telemetry, telecommand, and channel coding as well as planning, scheduling, monitoring, and control of ground stations and he is now member of the Technical Management and Support Office of the Missions and Operations Department of the Human Spaceflight and Operations (HSO), Directorate of ESA. His work also includes advanced studies and international standardization activities in the field of channel coding, communication protocols, and telemetry/telecommand processing.

Dr. Calzolari has been contributing to ESA Standardization activities since 1995. As of 1996, he was a member of the Consultative Committee for Space Data Systems (CCSDS) where, since 2000, he has been chairing the Working Group dealing with channel coding within the CCSDS space link services (SLS) area. He is also participating in the production of the E-50 series of standards for the European Cooperation for Space Standardization (ECSS). Sicne 2010, he has chaired the Standards Approval Board for Telemetry and Data-handling (STAB) of ESA and in 2011 he has been appointed CCSDS SLS Area Director.