Replacing the CCSDS Telecommand Protocol with the Next Generation Uplink (NGU)

Greg J. Kazz[1] and Ed Greenberg[2] and Scott C. Burleigh[3]
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

The current CCSDS Telecommand (TC) Recommendations[1-3]have essentially been in use since the early 1960s. The purpose of this paper is to propose a successor protocol to TC. The current CCSDS recommendations can only accommodate telecommand rates up to approximately 1 mbit/s. However today's spacecraft are storehouses for software including software for Field Programmable Gate Arrays (FPGA) which are rapidly replacing unique hardware systems. Changes to flight software occasionally require uplinks to deliver very large volumes of data. In the opposite direction, high rate downlink missions that use acknowledged CCSDS File Delivery Protocol (CFDP)[4] will increase the uplink data rate requirements. It is calculated that a 5 mbits/s downlink could saturate a 4 kbits/s uplink with CFDP downlink responses: negative acknowledgements (NAKs), FINISHs, End-of-File (EOF), Acknowledgements (ACKs). Moreover, it is anticipated that uplink rates of 10 to 20 mbits/s will be required to support manned missions. The current TC recommendations cannot meet these new demands. Specifically, they are very tightly coupled to the Bose-Chaudhuri-Hocquenghem (BCH) code in Ref. 2. This protocol requires that an uncorrectable BCH codeword delimit the TC frame and terminate the randomization process. This method greatly limits telecom performance since only the BCH code can support the protocol. More modern techniques such as the CCSDS Low Density Parity Check (LDPC)[5] codes can provide a minimum performance gain of up to 6 times higher command data rates as long as sufficient power is available in the data. This paper will describe the proposed protocol format, trade-offs, and advantages offered, along with a discussion of how reliable communications takes place at higher nominal rates.

I. Introduction

In the 1970s, as spaceborne digital technology was emerging, digital flight hardware was heavy, cumbersome, and power hungry thus limiting the complexity that could then be implemented in spaceborne transceivers and command decoders. The drivers behind the standardization of telecommand protocols stem from that era, namely: 1) Simple uplink coding, employed primarily to detect transmission errors, that was exclusively limited to the use of hard symbols provided by the flight receiver. 2) Command rates were very limited, thus short commands were implemented to provide Earth based support to control the spacecraft especially during spacecraft emergency events. 3) On-board flight controllers were simple with little or no memory thus requiring few commands to operate them and 4) The design of the telecommand protocol needed to emphasize both high communication and component reliability, limiting the inflight processing to a minimum and again stressing the need for very short commands and simplicity of implementation.

II. NGU Rationale

The CCSDS Telecommand Protocols evolved from the early NASA telecommand protocols. The CCSDS added the capability to extend the size of command messages and added both a Cyclic Redundancy Check (CRC) and a simple Automatic Repeat ReQuest (ARQ) protocol to improve the reliability for larger message sets. Current and future spacecraft require the use of uplink communication for support of a much wider variety of uses. While telecommand continues to be an essential application, both in normal and emergency situations, there is increasing demand for transmitting larger volumes of data to a spacecraft. There are various sources of these new demands on uplinks and with those demands come concerns of reliability, security, low-latency, and interoperability:

[1] Group Supervisor, End-to-End Information Systems Group, MS 301-490.
[2] Principal Engineer, End-to-End Information Systems Group, MS 301-490.
[3] Principal Engineer, End-to-End Information Systems Group, MS 301-490.

1) Onboard applications are tending to require larger volume uplinks than in the past. State of the art on-board telecommunication systems are deploying “software radios” implemented in FPGAs that can be easily reprogrammed in flight and support quantized bit outputs (required for higher performance codes). Moreover, changes to flight software applications occasionally require the uplink to deliver very large volumes of data due to reprogramming.

2) A growing practice is the use of selective repeat protocols on the uplink such as CFDP. Telecommand provides reliability mechanisms via a CRC and its own ARQ mechanism, via the Communications Operation Procedure 1 (COP-1) protocol (see Ref. 3). COP-1 is deliberately based on a very simple go-back-n repeat mechanism, but for that reason it is unsuitable for missions with very long signal propagation delays. Long round trip delays result in equally long interruptions in the command sequence, so that in practice, COP-1 can only be used inefficiently in deep space missions. Higher rate downlink missions that use selective repeat (ARQ) require higher uplink data rates. There are many typical uses for file uploads to spacecraft. For example, modern flight equipment can be reprogrammed, both with software for microprocessors and “logicware” for FPGAs.

3) Advances in technology have provided large improvements in performance over the existing TC baseline. New software transponders can operate at much lower Signal to Noise Ratios (SNR) and output quantized bits. With the advent of modern coding techniques, very considerable improvement in performance as measured in coding gain is possible over the existing BCH code. Providing for more powerful codes is a central element of the new Next Generation Uplink protocol.

4) There is an emerging need for added uplink security adding considerable size to the minimum command size. The CCSDS Space Data Link Layer Security (SDLS) working group is separately addressing space link communication security concerns and providing standard services for authentication, encryption, and authenticated encryption. Although optional today, the emerging Space Link Security Protocol[6] will provide a standard solution to space link security.

5) There is an emerging need of higher rate uplinks in the manned space program. Unlike robotic missions that only require data, manned missions will also require voice, video and Internet access. Voice brings with it additional isochronous communication requirements including the need for low latency. Video and Internet will require the highest uplink rates possible.

III. Current Telecommand Architecture

Deep space telecommanding currently deals with the same challenges inherent in all deep space link communications, while focusing on the special issues of uplink: relatively low-rate links, episodic commanding, and commanding in emergencies. One issue of particular interest is the inherent sensitivity of spacecraft commanding, since an undetected command error can spell disaster for the spacecraft. Reliability is paramount for telecommand with the principal metric being an extremely low undetected command bit error rate on the order of 10^{-9}.

There are many common issues with space communications links that all space link protocols must address, usually with broadly similar solutions. The problems of noisy, long-delay links indicate the use of forward error correction techniques. The need to detect the beginning of a frame in order to initiate the error detection/correction process leads to the use of a reserved bit pattern as a Message Start Sequence (MSS) or synchronization marker. The problem of detecting bit transitions in the radio signal is aided by randomization techniques and complicated by high symbol error rates due to the low symbol signal to noise ratio (SSNR). And in general, each of these potential solutions must be weighed against the risk of imposing an overly burdensome computational, storage or power demands on the spacecraft. Given the wide range of choices within these categories, each with its own set of trade-off issues, it is not surprising that specific integrated solutions can vary greatly in the details. For example, the details of the Telecommand architecture vary substantially from the details of the Telemetry architecture. This section will explore those distinctive features of the uplink architecture seen as relevant to the Next Generation Uplink recommendations.

The TC protocol is designed to reliably deliver a delimited command without extra processing. The protocol is designed so that each command can be placed into a single link layer transfer frame, thus once a frame is accepted its contents can be delivered directly. Thus the TC protocol allows the frame to be of variable length matching the size of the command. The TC protocol frame consists of a MSS (to delimit the beginning of the frame), a primary transfer frame header (to provide the routing and accounting information for the protocol), a data field to carry the contents of the command and an optional CRC word (to lower the undetected error rate). On the spacecraft, the link layer process delimits the frame and then starts decoding the series of codewords that comprise the frame. When the non-decodable codeword that is added to the command frame by the protocol to signal the end of the Communications Link Transmission Unit (CLTU) is encountered, then the frame contents is ready to be acted upon by the receiver. Thus prime elements of the TC protocol are the MSS and the method used to delimit the CLTU.

The current telecommand standards recommend the use of the forward error correction/detection (64,56) Bose-Chaudhuri-Hocquenghem (BCH) code. In fact, this code is currently deployed in two different modes based upon the environment. For deep space missions, it is used in single-error-correction/double-error-detection (SEC/DED) mode since forward error correction is most efficient for long one-way light time transmissions. For near earth missions, it is primarily used in triple-error-detection (TED) mode since it reduces the undetected error rate and the use of COP-1 to retransmit erred frames is fairly efficient. While the BCH code is not an efficient code by today's standards, it has some important virtues from the standpoint of uplink. The BCH makes use of very short codewords (64 bits), making it well suited for composing a variable-length transfer frame (codeblock) with a variable series of codewords. Further, the BCH offers one of the simpler onboard implementations. The preference for implementation simplicity in order to provide reliability remains a key metric, though clearly more so for legacy spacecraft.

As is typical for trade-offs, while the use of error-correcting mode helps raise the frame acceptance rate, it comes at the cost of raising the undetected error rate. For situations like this, the telecommand standard provides an option of adding a cyclic redundant check (CRC) in order to further reduce the probability of an undetected error.

When error-correction fails (or is not sufficient), the current telecommand standard optionally utilizes COP-1 as the ARQ process to deliver reliable and complete commands as an optional procedure. Referred to as a command operating procedure or COP-1, it essentially returns to the set of ordered commands at the point of the error and restarts the transmission. This very simple technique is described as "go-back-n" as compared to the more targeted technique described as "selective repeat". Despite its relative simplicity, the COP-1 does require many cooperating elements: a Frame Operating Procedure (FOP) to administer source frame transmission, a Frame Acceptance and Reporting Mechanism (FARM) for detecting and reporting missing or out-of-order frames, and a Communications Link Control Word (CLCW) for reporting error frames conveyed back to the source by means of the CCSDS telemetry protocols[7-8]. Again, the relative simplicity of the algorithm is a virtue that comes at the cost of efficiency. The COP-1 technique can result in a lengthy halt of the command sequence in cases where the round-trip time is long. These long round-trip times are characteristic of deep space missions, and so in practice, these missions do not use the retransmission mechanism of COP-1, but instead use duplicate commands to increase the probability of acceptance of a command sequence.

IV. NGU Architecture

There have been numerous technology advances in recent decades that are substantially changing the character and context for newer missions along with the mix of potential solutions. This section will focus principally on technology advances and trade offs that can offer new choices for the development of a better uplink architecture. The trade offs/emerging technologies examined by this paper are:

1) Selection of the NGU retransmission technique;
2) Selection of the data link layer protocol;
3) Selection of a Forward Error Correcting (FEC) Code family;
4) Selection of the MSS to delimit the data link frame;
5) Selection of the CLTU termination method.

A. Selection of the Retransmission Technique

Recent decades have seen important developments in space link and transport protocols, such as the CCSDS Advanced Orbiting System (AOS), Proximity-1 (Prox-1)[9], CFDP, the emerging Delay Tolerant Networking (DTN)[10] architecture and Licklider Transport Protocol (LTP)[11]. Among other things, these new protocols bring various new ARQ techniques. This is a recurring theme where ARQ serves as the reliability mechanism of last resort when the various forward error correction schemes fail to deliver a data unit successfully. As previously explained, the COP-1 ARQ mechanism is very inefficient under conditions of long round-trip delays or repeated commands, so some form of selective-repeat-ARQ is more attractive. Generally, a selective-repeat-ARQ will target the missing data unit(s) and not cause any delay in concurrent transmissions. Among these selective-repeat mechanisms, the ARQ of LTP appears to be the most promising. LTP offers an optional selective-repeat-ARQ that can be adapted to any of the underlying space data link protocols, while remaining low enough in the stack to offer maximum flexibility for use at higher layers. For example, the CFDP ARQ is specialized to entire files while LTP is neutral with respect to the ultimate data format.

Most of the design concepts of LTP are inherited from the design of the CFDP. Rather than applying these concepts to the transmission of a file, however, LTP procedures accomplish the reliable transmission of an arbitrary *block* of data; the block transmitted by a single LTP *session* might comprise an entire file, a portion of a file, multiple complete or partial files, or indeed one or more data structures that are not files at all.

LTP block transmission begins with the explosion of the block, which may be arbitrarily large, into some number of data *segments*, each of which is small enough to fit into a single protocol data unit (PDU) of the underlying link service (LS) protocol.[4] Each segment begins with a small header that identifies the session of which this segment is a part and indicates the nature of the segment. Any number of segments, starting with the first, may be designated "red data", meaning they are subject to ARQ; any remaining segments are designated "green data", meaning they are subject only to "best efforts" transmission procedures. The segments are encapsulated in LS PDUs and transmitted via the LS protocol. The last "red" segment of the block is flagged as a *checkpoint*, and a retransmission timer is set at the moment the checkpoint segment is transmitted. If that timer expires before a *report* segment responding to this checkpoint is received, then the checkpoint is retransmitted. Timeout intervals are dynamically adjusted in response to announced changes in the transmission state of the LTP sender and/or receiver.

The receiving LTP engine reassembles the block from received segments. On reception of the "checkpoint" segment, LTP at the receiver prepares and transmits a report segment stating which portions of the block's "red" data were received and which were lost or corrupted in transmission and therefore must be retransmitted. In symmetry with the sender, a retransmission timer is set at the moment the report segment is transmitted. If that timer expires before a *report acknowledgment* segment responding to this report is received, then the report is retransmitted.

On reception of the report segment, the sending LTP engine immediately sends a responding report acknowledgment segment and then retransmits any data segments that the report indicates were not successfully received; the last retransmitted data segment is flagged as another checkpoint, for which a timer is set as before.

This dialogue continues until either the receiving LTP engine has received all "red" data in the block – and has noted this successful reception in a final report segment, which has been acknowledged – or a limit on retransmission has been reached at either the sender or the receiver and the transmission session is canceled.

Importantly, any number of block transmission sessions may be in various stages of progress between two LTP engines concurrently. That is, loss of data in one session does not delay the transmission of more data in a subsequent session. This parallelism minimizes latency in the delivery of whatever data have been successfully acquired at the receiving engine.

Moreover, only those data that are actually lost in transmission – not all data transmitted after the lost segments, as in a "go back N" algorithm – are retransmitted. This minimizes waste of bandwidth due to retransmission of data, which were in fact successfully received.

Although LTP is somewhat more complex than the current Telecommand protocol's COP-1 procedures, it offers a number of operational advantages:

1) As noted above, it accomplishes reliable transmission of command data even when operating over extremely large signal propagation delays, without imposing unnecessary latency in data delivery and without incurring unnecessary bandwidth consumption.
2) Because LTP acknowledgment is at block granularity, the volume of acknowledgment traffic can be reduced by merely increasing block size. This enables LTP to operate effectively even over highly asymmetrical links.
3) LTP's ability to dynamically revise timeout intervals enables reliable data delivery despite lengthy, potentially irregular interruptions in link service.
4) Finally, an extension mechanism is built into the LTP specification. This enables mission-specific adaptations to be implemented in a conformant manner, and it also enables future extensions to LTP functionality to be readily standardized. Currently, for example, several simple security measures have been integrated into LTP.

Taken together, these advantages enable uplink data to be reliably conveyed to spacecraft at higher data rates than COP-1.

[4] LTP may be used over any link-layer service protocol, e.g., CCSDS Telecommand or AOS, PPP, IEEE 802.11, etc.

B. Selection of the Data Link Layer Protocol

A data link layer protocol per the appropriate environment needs to be chosen. The options are: 1) TC Space Data Link Protocol or 2) AOS Space Data Link Protocol.

	Telecommand (TC)	**AOS**
Mission Usage	Unmanned Missions	Manned Missions
Synchronous/Asynchronous	Asynchronous – Idle allowed	Synchronous – No Idle uses Fill Frames when no data
Payload	One message per frame Message within a packet or segment or byte stream	Bit stream, Byte stream or Packet based data transfer
Short Commands	Advantage – low latency & small code word size	N/A
Long Commands	N/A	Advantage – use larger size code word to obtain better performance
CCSDS CLTU Service	No changes required to add LDPC code capability	Change to synchronous uplink (fill frames) required
Voice	No provision	Insert Zone provides for low latency and meets data needs

Table 1. Data Link Layer Protocol Trade-off

The TC protocol was designed for unmanned missions that typically require low rate commanding (8 to 2000 bits/s). The requirement for short emergency commands when data rates are limited favors the TC protocol. As long as space agencies cannot relax that requirement and continue to require the use of short commands i.e., approximately 56 to 256 bits for emergency commanding, then TC seems to be the prudent choice. The driver behind the use of short commands in the emergency case is the available window period involved in receiving these commands. Window requirements depend on code block size and data rate. A code block could be composed of one code word at low rates and a series of concatenated code words allowing the frame size to be larger for higher rate deliveries with small implementation cost. Similarly, one could concatenate a series of smaller code words into a larger code block i.e., telecommand frame. See Table 1 above.

AOS commanding has emerged as the prime method for supporting manned missions that require much higher uplink rates (.01 to 20 mbits/s). AOS provides a more efficient synchronous data delivery commensurate with the higher desired rates. See Table 1 above.

C. Selection of the FEC Code Family

One or multiple FEC code(s), with associated code rate, and code word size, need to be chosen. The options are:

	Code SNR operating point Frame(TC)/AOS	**Performance Impact**
BCH (56,64)	~9 dB (TC)	Current Baseline
LDPC ½, (128,64)	~5 dB (TC)	~4 dB better than BCH
LDPC ½, (256,128)	~4 dB (TC)	~5 dB better than BCH
LDPC ½, (512,256)	~3 dB (TC)	~6 dB better than BCH Send (64,56) code 4x faster But same window as current TC
Non-binary LDPC ½, (512,256)	~2 dB	~7 dB improvement More complex implementation
LDPC ½, (2048,1024)	~1.5 dB (AOS)	~7.5 dB better than BCH data rate 5x faster but larger code word needs 3x larger window

LDPC ½, (8192,4096)	~1.2 dB (AOS)	~7.8 dB better than BCH
LDPC ½, (16384,8192)	~1.0 dB (AOS)	~8 dB better than BCH

Table 2. FEC Code Selection Trade-off

Modern coding techniques offer very large improvements in performance. In particular, the Low Density Parity Check (LDPC) family of codes can approach the theoretical limits of efficiency as established by the Shannon sphere-packing bound. While the LDPC codes were discovered in the 1960s, it has been the development of practical decoding algorithms and hardware advancements that now makes LDPC so attractive. These new decoding algorithms, described as Belief Propagation (BP), provide near-optimum performance with manageable complexity. In comparison to the BCH codes presently used by TC, an LDPC code can achieve the same error floor with a reduction in power requirements of between 2.5 and 8.5 dB.

Per Table 2, clearly the largest coding gain is achieved using the largest LDPC code block size of 16384 bits. However that comes at the price of increasing the latency of reception of the commands. An interesting comparison can be seen between the LDPC (512, 256) and the BCH (56,64) codes. The LDPC (512,256) code can, with the same configuration and operating condition, accommodate a four times higher data rate than the current BCH code and thus deliver a 248 bit command (256 information bits- 8 bits of the signaling byte) within the same time period currently required to deliver a 56 bit command. The minimum command period would be 248 information bits, 8 bits for signaling the end of the code block, and ~64 bits for frame synchronization (total 320 bits). Note that the emerging CCSDS Space Data Link Security protocol requirements could also be accommodated. The current minimum emergency command size that is often quoted is 56 bits based upon the information size of a BCH code word. The use of the LDPC (512, 256) code would accommodate this 64-bit command and would allow for the inclusion of 192 bits for link security see Ref. 6. The use of the LDPC (2048,1024) code would provide a gain of 7.5 dB over the BCH baseline, but would require about 3 times the radiation time period.

An even larger LDPC code can be used with this same approach but it seems that it would be better suited to AOS where the frame size is fixed and the current emergency mode requirement for reception of an emergency command within a short receiving time window is no longer required.

Emerging studies by B. Chang, D. Divsalar, and L. Dolecek (see Ref 12) conclude that there is about a 1 dB improvement in performance for the rate ½, short (64, 128, 256) non-binary protograph LDPC codes over the binary protograph codes. However, the decoding complexity of these non-binary codes is much higher than their binary counterparts, by a factor of approximately 64 that makes implementation at this time unlikely for such a small performance gain.

We are left with some important trade-off issues/questions to be considered when choosing new high performance code(s) for TC protocol standardization:

1) Delimiting of the CLTU requires the selection of a higher performance Message Start Sequence along with the method for signaling the end of the CLTU.

2) Selection of the FEC code family for uplink.

3) Should a single FEC code satisfy the totality of a mission's needs (e.g., latency, EIRP) or should multiple codes be implemented on a single mission?

D. Selection of the Message Start Sequence (MSS, Synchronization Word)

For the NGU protocol we propose to modify the current CLTU delimiting method. The use of high performance FEC codes provides much better performance but significantly increases the symbol error rate output by the receiver. Thus a new higher performance MSS is required in order to enable delimiting the beginning of the command frame. The size of the new MSS will depend upon the FEC code used because the symbol error rate will differ based on the code selected. The current candidate MSS options based upon CCSDS standards are: 1) 16 bit (BCH) and 2) 64 bit (LDPC). The selection of the synchronization word size depends upon the selected code and the operating symbol SNR point and whether or not idle is allowed between frames. A very high probability of obtaining synchronization in one synchronization word is required for telecommand.

E. Selection of Command Link Transmission Unit (CLTU) Termination Method

Terminating the CLTU can be accomplished by one of three techniques: 1) Use of a single non-decodable codeword, 2) use of a termination flag in each of the codewords or 3) use of the frame length field within the TC primary frame header. The use of the non-decodable code word is an acceptable method when the code word length is short and the frame is large but this method adds significant overhead when the codeword is long and the command frame is short (as desired for emergency commanding) – see Table 3 below. The use of a flag within each codeword to signal the last codeword delimiting the CLTU is very efficient for short command frames but each added codeword in a frame must carry the flag; this increases overhead for frames requiring many codewords. The use of the frame length requires no added overhead but uses the TC frame structure to delimit the frame and determine the last code word in order to start searching for the next frame. This method depends on an error free frame length field, so it is highly dependent upon the FEC code chosen.

Command Size	Signaling Byte per Code word Overhead	Erred Code word Overhead
256 bits	8/248 ~ 3.2%	256/256 = 100%
1024 bits	32/994 ~ 3.2%	256/1024 = 25%
8192 bits	256/7940 ~ 3.2%	256/8196 ~ 3%

Table 3. Code block termination method trade-off

V. Conclusion

1) By using a better (in comparison to the current BCH code) performing FEC code on the uplink, as long as sufficient data power is present one could take advantage of either higher uplink margins/lower error rates at the current uplink rates and/or higher data rates/lower error rates than are currently in use. Note: These higher rates could be used for relief of emergency communication margins/rates and not limited to improving top-end rate performance.
2) A higher performance uplink could also reduce the requirements on flight emergency antennas and/or the performance required from ground stations.
3) Use of a selective repeat ARQ protocol may increase the uplink design requirements but the resultant development is deemed acceptable, due the factor of 4 to 8 potential increase in uplink data rate.
4) Use of a selective repeat ARQ protocol allows one to work closer to the uplink margin, which enables higher uplink rates but incurs a frame loss penalty, requiring retransmission – acceptable as long as the frame error rate doesn't overwhelm the error correction/detection capability of the FEC code and sufficient symbol SNR is provided.
5) A higher performance uplink provides added bandwidth required for the security header/trailer defined by the emerging CCSDS Space Data Link Security Protocol.

Among the missions that we believe could immediately benefit from a NGU protocol if it were a standard today, would be missions whose low end uplink rates are severely constrained due to their operational environment. In the cases of low rate nominal operations and emergency uplinks, NGU could be used to relieve the most adverse uplink conditions that severely constrain mission operations. Some current missions that could benefit are: NASA Mars Science Laboratory (MSL), currently in operations, and the NASA Solar Probe Plus (SPP) mission, under development. Like most deep space missions, MSL's emergency uplink rate is 7.8125 bits/s. This rate severely limits recovery operations whenever the spacecraft falls into safe mode. Similarly, SPP is severely bandwidth limited whenever the spacecraft is close to the Sun. Capitalizing on the NGU performance improvements would enable SPP to utilize more bandwidth-intensive selective repeat ARQ techniques (e.g., CFDP, LTP) in order to meet its uplink completeness requirements.

Appendix A
Acronym List

ACK	Acknowledgement
AOS	Advanced Orbiting Systems
ARQ	Automatic Repeat ReQuest
BCH	Bose-Chaudhuri-Hocquenghem
BP	Belief Propagation
CCSDS	Consultative Committee for Space Data Systems
CER	Code word Error Rate
CFDP	CCSDS File Delivery Protocol

CLCW	Communications Link Control Word
CLTU	Command Link Transmission Unit
COP-1	Communications Operations Procedure 1
CRC	Cyclic Redundancy Check
DLR	Deutsche Luft und Raumfahrt
DTN	Delay/Disruption Tolerant Networking
EIRP	Effective Isotropic Radiated Power
FARM	Frame Acceptance and Reporting Mechanism (part of COP-1)
FEC	Forward Error Correction
FOP	Frame Operating Procedure (part of COP-1)
FPGA	Field Programmable Gate Array
IEEE 802.11	IEEE 802.11 Wireless Ethernet Standards
LDPCC	Low-Density Parity Check Codes
LTP	Licklider Transmission Protocol
LS	Link Service
MESSENGER	Mercury Surface, Space Environment, Geochemistry and Ranging
MSL	Mars Science Laboratory
MSS	Message Start Sequence
NAK	Negative Acknowledgement
NASA	National Aeronautics and Space Administration
NGU	Next Generation Uplink
PDU	Protocol Data Unit
PPP	Point to Point Protocol
PROX-1	Proximity-1 Space Link Protocol
RF	Radio Frequency
SDLS	Space Data Link Security
SEC/DED	Single-Error-Correction/Double-Error-Detection
SNR	Signal to Noise Ratio
SSNR	Symbol Signal to Noise Ratio
SPP	Solar Probe Plus
TC	Telecommand
TED	Triple-Error-Detection

Appendix B
Glossary

Acknowledge Repeat ReQuest	An error-control method for data transmission that uses acknowledgements and timeouts to achieve reliable data transmission over an unreliable service.
CCSDS	An organization of Space Agencies that produces space data standards mainly for flight and ground systems and their interface to space systems.
CFDP	File Delivery Protocol intended for use on board spacecraft, such as its use on the uplink to NASA MESSENGER now orbiting Mercury.
Effective Isotropic Radiated Power	Amount of power that a theoretical isotropic radiator would emit to produce the peak power density observed in the direction of maximum antenna gain
Licklider Transmission Protocol	Intended to serve as a reliable convergence layer over single-hop deep-space radio frequency (RF) links.

Low-Density Parity Check Codes A linear error correcting code: a method of transmitting a message over a noisy transmission channel, and is constructed using a sparse bipartite graph.

Acknowledgments

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration. The authors thankfully acknowledge the collaboration of Richard Borgen as well as Kenneth S. Andrews, Jon Hamkins and Fabrizio Pollara for their work on LDPC codes.

References

[1]CCSDS TC Space Data Link Protocol, CCSDS 232.0-B-1. Blue Book. Issue 1. September 2003. Consultative Committee for Space Data Systems, www.ccsds.org.

[2]CCSDS TC Synchronization and Channel Coding, CCSDS 231.0-B-2. Blue Book. Issue 2. September 2010. Consultative Committee for Space Data Systems, www.ccsds.org.

[3]CCSDS Communications Operation Procedure-1, CCSDS 232.1-B-2. Blue Book. Issue 2. September 2010. Consultative Committee for Space Data Systems, www.ccsds.org.

[4]CCSDS File Delivery Protocol (CFDP), CCSDS 727.0-B-4. Blue Book. Issue 4. January 2007. Consultative Committee for Space Data Systems, www.ccsds.org.

[5]CCSDS TM Synchronization and Channel Coding, CCSDS 131.0-B-2. Blue Book. Issue 2. August 2011. Consultative Committee for Space Data Systems, www.ccsds.org.

[6]CCSDS Space Data Link Security Protocol, CCSDS 355.0-R-2. Red Book. Issue 2. February 2012. Consultative Committee for Space Data Systems, www.ccsds.org

[7]CCSDS TM Space Data Link Protocol, CCSDS 132.0-B-1. Blue Book. Issue 1. September 2003. Consultative Committee for Space Data Systems, www.ccsds.org

[8]CCSDS AOS Space Data Link Protocol, CCSDS 732.0-B-2. Blue Book. Issue 2. July 2006. Consultative Committee for Space Data Systems, www.ccsds.org

[9]CCSDS Proximity-1 Space Link Protocol – Data Link Layer, CCSDS 211.0-B-4. Blue Book. Issue 4. July 2006. Consultative Committee for Space Data Systems, www.ccsds.org

[10]CCSDS Rationale, Scenarios, and Requirements for DTN in Space, CCSDS 734.0-G-1. Green Book. Issue 1. August 2010. Consultative Committee for Space Data Systems, www.ccsds.org

[11]CCSDS Licklider Transmission Protocol (LTP), CCSDS 734.1-R-2. Red Book. Issue 2. February 2012. Consultative Committee for Space Data Systems, www.ccsds.org

[12]Ben-Yue B., Divsalar D., and Dolecek L., Jet Propulsion Laboratory, California Institute of Technology, "Non-binary Protograph LDPC Codes with Short Blocklengths." Presentation to CCSDS Next Generation Uplink Working Group, Spring Meeting 2012, Darmstadt, Germany.