The Consultative Committee for Space Data Systems

Research and Development for
Space Data System Standards

SHORT BLOCK LENGTH LDPC CODES FOR TC SYNCHRONIZATION AND CHANNEL CODING

EXPERIMENTAL SPECIFICATION

CCSDS 231.1-O-1

ORANGE BOOK

April 2015

The Consultative Committee for Space Data Systems

Research and Development for Space Data System Standards

SHORT BLOCK LENGTH LDPC CODES FOR TC SYNCHRONIZATION AND CHANNEL CODING

EXPERIMENTAL SPECIFICATION

CCSDS 231.1-O-1

ORANGE BOOK

April 2015

AUTHORITY

Issue:	Orange Book, Issue 1
Date:	April 2015
Location:	Washington, DC, USA

This document has been approved for publication by the Consultative Committee for Space Data Systems (CCSDS). The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

FOREWORD

This document is a CCSDS Experimental Specification for a set of short block length Low Density Parity Check (LDPC) codes intended for telecommand applications. It was contributed to CCSDS by NASA and DLR.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the e-mail address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a CCSDS Experimental Specification. Its Experimental status indicates that it is part of a research or development effort based on prospective requirements, and as such it is not considered a Standards Track document. Experimental Specifications are intended to demonstrate technical feasibility in anticipation of a 'hard' requirement that has not yet emerged. Experimental work may be rapidly transferred onto the Standards Track should a hard requirement emerge in the future.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 231.1-O-1	Short Block Length LDPC Codes for TC Synchronization and Channel Coding, Experimental Specification, Issue 1	April 2015	Original issue

CONTENTS

CONTENTS (continued)

Figure Page

Table

1 INTRODUCTION

1.1 BACKGROUND

With the advent of modern coding techniques, considerable improvement is possible over the existing BCH codes specified in *TC Synchronization and Channel Coding* (reference [1]). Moreover, spacecraft and their communications protocols continue to become more complex, so there is a corresponding demand for more sophisticated uplink coding capabilities.

Traditionally, uplink communication has been used primarily for telecommand, where short command sequences were transmitted to unmanned spacecraft a few times per day to a few times per month. Current and future spacecraft use uplink communication for 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 spacecraft. Flight equipment can be reprogrammed, both with software for microprocessors and 'logicware' for Field Programmable Gate Arrays. Manned missions benefit from live uplink video and Internet access.

The space telecommunications environment has also evolved. In some cases there may be many spacecraft in a small solid angle as observed from Earth, such as in the Mars vicinity, or at a lunar outpost, and there is interest in serving Multiple Spacecraft Per Antenna (MSPA). As computation becomes cheaper relative to large mechanical structures, there is interest in retiring the 70 meter Deep Space Network (DSN) antennas, thus reducing the maximum available transmit power, and compensating with more elaborate protocols and signal processing algorithms. Despite these changes, some facts hold constant: distances in deep space are immense, and received signal power is correspondingly miniscule. Power efficiency remains a dominant priority. When a spacecraft emergency affects the telecommunications system, delivering any information at all can be difficult, and there must be a method to deliver short commands at an extremely low data rate when necessary.

1.2 PURPOSE AND SCOPE

The Next Generation Uplink (NGU) (reference [3]) initiative identifies a greater variety of uplink needs, categorized into four modes as summarized in table 1-1. These different applications place different demands on the telecommunications link, and hence the error correcting codes chosen for each application may also be different.

Table 1-1: Modes of Operation for Uplink Codes[1]

Mode	Purpose	Block Length (kb)	Throughput (kb/s)
A	Emergency	0.1	0.01
B	Command/ARQ	0.1–1	1–4
C	File Upload	1–4	1000
D	Human Support	>4	20000

In this document, several sets of three short block length LDPC codes are presented as appropriate candidates for Modes A and B. These codes may be substituted in place of the BCH code specified in *TC Synchronization and Channel Coding* for improved performance, with virtually no impact on other aspects of the telecommand system. Section 2 presents a family of three binary LDPC codes that have been reasonably well characterized. Section 3 presents several design alternatives for families of non-binary LDPC codes. These options all operate at a lower Signal-to-Noise Ratio than the binary codes, but at the cost of more complex decoders.

1.3 NOMENCLATURE

1.3.1 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

[1] Cf. reference [3].

1.3.2 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.4 CONVENTIONS

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'Bit 0', the following bit is defined to be 'Bit 1', and so on up to 'Bit *N*-1'. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., 'Bit 0' (see figure 1-1).

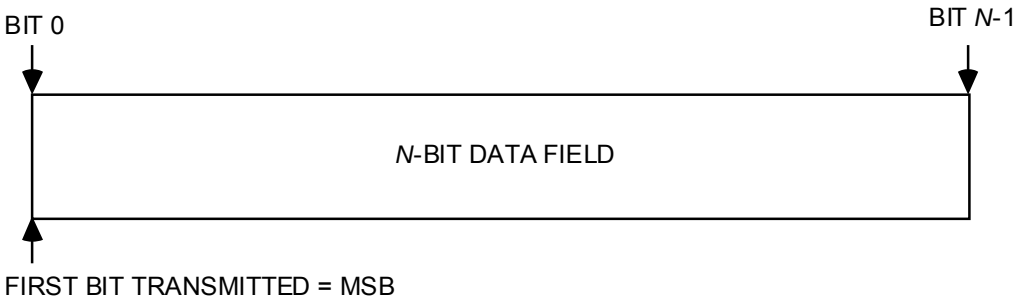


Figure 1-1: Bit Numbering Convention

The convention for matrices differs from that for bit fields. Matrices are indexed beginning with the number '1'.

In accordance with standard data-communications practice, data fields are often grouped into 8-bit 'words' which conform to the above convention. Throughout this Specification, such an 8-bit word is called an 'octet'.

The numbering for octets within a data structure starts with '0'.

1.5 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *TC Synchronization and Channel Coding*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-2. Washington, D.C.: CCSDS, September 2010.

[2] *TC Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-2. Washington, D.C.: CCSDS, September 2010.

[3] *Next Generation Uplink*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 230.2-G-1. Washington, D.C.: CCSDS, July 2014.

2 LOW DENSITY PARITY CHECK CODE SPECIFICATION

2.1 DESCRIPTION OF THE CODES

The three Low-Density Parity-Check (LDPC) codes defined here are systematic. All are transparent, so phase ambiguities may be resolved either by using frame markers (which are required for codeblock synchronization) or by another means after decoding.

LDPC codes may be used to obtain greater coding gain than that provided by the BCH code.

NOTES

1 LDPC coding, by itself, cannot guarantee sufficient bit transitions to keep receiver symbol synchronizers in lock. Therefore the Pseudo-Randomizer defined in reference [1] is required.

2 These LDPC codes possess relatively large minimum distance for their block length, and undetected error rates lie several orders of magnitude below detected frame and bit error rates for any given operating signal-to-noise ratio. This property, combined with the Frame Error Control Field (FECF) specified in reference [2] assures an extremely low undetected error rate.

2.2 SPECIFICATION

2.2.1 GENERAL

These LDPC codes are specified indirectly by an m-by-n parity-check matrix H consisting of m linearly independent rows. A coded sequence of n bits must satisfy all m parity-check equations corresponding to the m rows of H. An encoder maps an input frame of $k=n-m$ information bits uniquely into a codeblock of n bits.

Three LDPC codes are specified, with codeblock lengths (n=128, k=64), (n=256, k=128), and (n=512, k=256), so that each has code rate $r=k/n=1/2$.

2.2.2 PARITY CHECK MATRICES

The H matrices are constructed from $M \times M$ sub-matrices, where $M=k/4=n/8$. They are specified as:

$$H_{64\times128} = \begin{bmatrix} I_M \oplus \Phi^7 & \Phi^2 & \Phi^{14} & \Phi^6 & 0_M & \Phi^0 & \Phi^{13} & I_M \\ \Phi^6 & I_M \oplus \Phi^{15} & \Phi^0 & \Phi^1 & I_M & 0_M & \Phi^0 & \Phi^7 \\ \Phi^4 & \Phi^1 & I_M \oplus \Phi^{15} & \Phi^{14} & \Phi^{11} & I_M & 0_M & \Phi^3 \\ \Phi^0 & \Phi^1 & \Phi^9 & I_M \oplus \Phi^{13} & \Phi^{14} & \Phi^1 & I_M & 0_M \end{bmatrix}$$

$$H_{128\times256} = \begin{bmatrix} I_M \oplus \Phi^{31} & \Phi^{15} & \Phi^{25} & \Phi^0 & 0_M & \Phi^{20} & \Phi^{12} & I_M \\ \Phi^{28} & I_M \oplus \Phi^{30} & \Phi^{29} & \Phi^{24} & I_M & 0_M & \Phi^1 & \Phi^{20} \\ \Phi^8 & \Phi^0 & I_M \oplus \Phi^{28} & \Phi^1 & \Phi^{29} & I_M & 0_M & \Phi^{21} \\ \Phi^{18} & \Phi^{30} & \Phi^0 & I_M \oplus \Phi^{30} & \Phi^{25} & \Phi^{26} & I_M & 0_M \end{bmatrix}$$

$$H_{256\times512} = \begin{bmatrix} I_M \oplus \Phi^{63} & \Phi^{30} & \Phi^{50} & \Phi^{25} & 0_M & \Phi^{43} & \Phi^{62} & I_M \\ \Phi^{56} & I_M \oplus \Phi^{61} & \Phi^{50} & \Phi^{23} & I_M & 0_M & \Phi^{37} & \Phi^{26} \\ \Phi^{16} & \Phi^0 & I_M \oplus \Phi^{55} & \Phi^{27} & \Phi^{56} & I_M & 0_M & \Phi^{43} \\ \Phi^{35} & \Phi^{56} & \Phi^{62} & I_M \oplus \Phi^{11} & \Phi^{58} & \Phi^3 & I_M & 0_M \end{bmatrix}$$

where I_M and 0_M are the $M \times M$ identity and zero matrices, respectively, and Φ is the first right circular shift of I_M. That is, Φ has a nonzero entry in row i and column j if $j=i+1 \bmod M$. It should be noted that Φ^2 is the second right circular shift of I_M, etc., and $\Phi^0=I_M$. The $\oplus$ operator indicates modulo-2 addition.

2.3 ENCODING

One method for producing codeblocks consistent with the parity-check matrices is to perform matrix multiplication by block-circulant generator matrices. These matrices may be constructed as follows.

a) Let P be the $4M \times 4M$ sub-matrix of H consisting of the last $4M$ columns. Let Q be the $4M \times 4M$ sub-matrix of H consisting of the first $4M$ columns.

b) Compute $W = (P^{-1}Q)^T$, where the arithmetic is performed modulo-2.

c) Construct the matrix $G = \begin{bmatrix} I_{4M} & W \end{bmatrix}$, where I_{4M} is the $4M \times 4M$ identity matrix, and W is a dense matrix of circulants of size $4M \times 4M$.

For convenience, the matrices W can also be constructed from the hexadecimal values in tables 2-1, 2-2, and 2-3. Rows 1, $M+1$, $2M+1$, and $3M+1$ are found by converting the hexadecimal values to binary strings, and the other rows are constructed from right circular shifts of those strings.

Table 2-1: Generator Matrix for (*n*=128,*k*=64) LDPC Code

$G_{64\times128}$	
Row 1	0E69 166B EF4C 0BC2
Row 17	7766 137E BB24 8418
Row 33	C480 FEB9 CD53 A713
Row 49	4EAA 22FA 465E EA11

Table 2-2: Generator Matrix for (*n*=256,*k*=128) LDPC Code

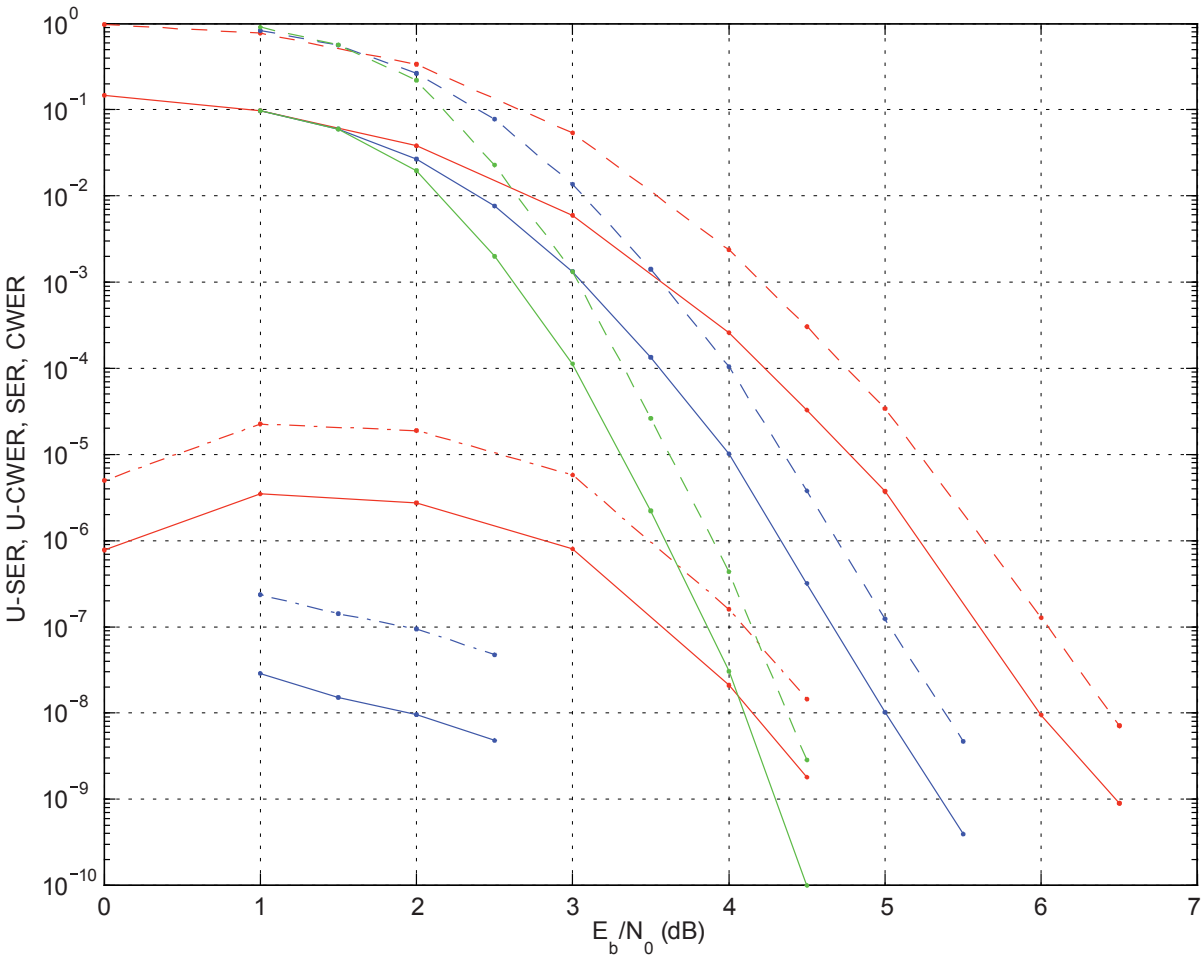
$G_{128\times256}$	
Row 1	73F5E839 0220CE51 36ED68E9 F39EB162
Row 33	BAC812C0 BCD24379 4786D928 5A09095C
Row 65	7DF83F76 A5FF4C38 8E6C0D4E 025EB712
Row 97	BAA37B32 60CB31C5 D0F66A31 FAF511BC

Table 2-3: Generator Matrix for (*n*=512,*k*=256) LDPC Code

$G_{256\times512}$	
Row 1	1D21794A22761FAE 59945014257E130D 74D6054003794014 2DADEB9CA25EF12E
Row 65	60E0B6623C5CE512 4D2C81ECC7F469AB 20678DBFB7523ECE 2B54B906A9DBE98C
Row 129	F6739BCF54273E77 167BDA120C6C4774 4C071EFF5E32A759 3138670C095C39B5
Row 193	28706BD045300258 2DAB85F05B9201D0 8DFDEE2D9D84CA88 B371FAE63A4EB07E

2.4 PERFORMANCE

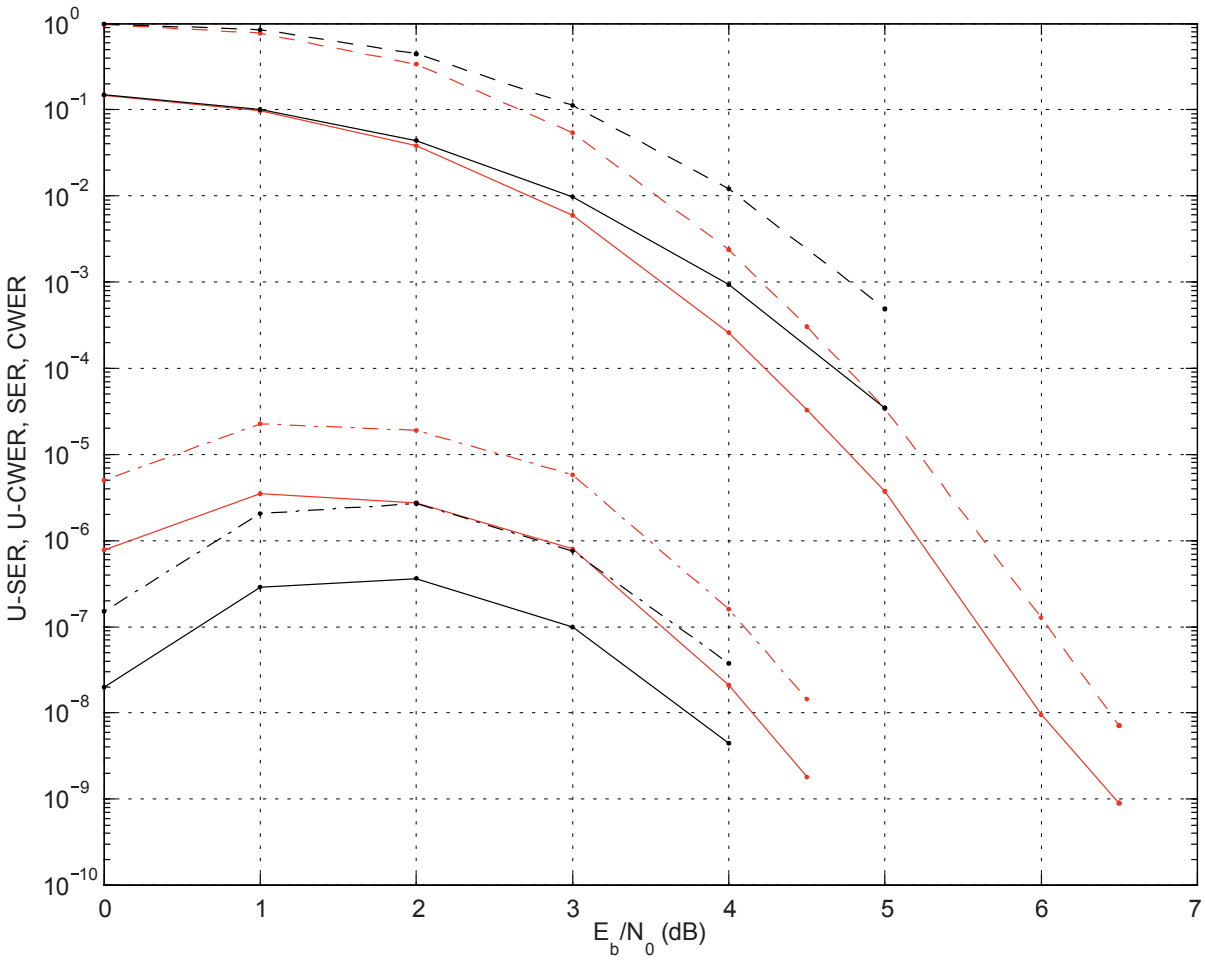
Performance for the three codes has been determined by software simulation, and is shown in figure 2-1. These curves were generated using the min* decoding algorithm (reference [C13]) with 100 iterations maximum. Results for the short (*n*=128, *k*=64) code are shown in red, those for the medium (*n*=256, *k*=128) code are in blue, and the long (*n*=512, *k*=256) code are in green. The uppermost fan of dashed curves show the CodeWord Error Rate (CWER), and the overlapping fan of solid curves give the Symbol Error Rate (SER) (very similar to Bit Error Rate). Also shown for the short code are Undetected CodeWord Error Rate (U-CWER) and undetected symbol error rate (U-SER). For the medium length code, initial results for U-CWER and U-SER are shown, but the data is not reliable. No undetected errors have been observed for the long (*n*=512, *k*=256) code.



Green=(512,256), Blue=(256,128), Red=(128,64);
Dashed=CWER, Solid=SER

Figure 2-1: Total and Undetected Error Rates of Short Block Length LDPC Codes

The undetected error rate can be improved, at a cost in total error rate, by modifying the decoder. One method is to decrease the maximum number of iterations performed. For the short (n=128, k=64) code, two sets of results are shown in figure 2-2. The red curves are identical to those shown in figure 2-1; the black curves are for a decoder that performs a maximum of 10 iterations instead of 100. In the measurable region, this lowers the undetected error rates by about an order of magnitude. Other, better, methods for building incomplete decoders are described in references [C1] and [C2].



Red=100 Iterations, Black=10 Iterations

Figure 2-2: Total and Undetected Error Rates of the (128,64) Code with Two Decoders

The minimum distance of the (128,64) code is d_{min}=14, for the (256,128) code, d_{min}=24, and for the (512,256) code, d_{min} remains unknown.

2.5 CODE DESIGN

Based on experience at JPL, other NASA centers, and of the research community at large, a protograph-plus-circulant construction (reference [C3]) was selected. This construction permits fast, simple encoders, and is amenable to fast, well-structured decoders. The protograph design is also amenable to analysis during code design and characterization. For long block length codes, research shows that good designs typically use about 3.5 edges per variable node; for short codes, this number must be increased to eliminate small trapping sets (reference [C4]) and low-weight codewords. A wide variety of protographs were studied, and that shown in figure 2-3 yields excellent performance for short block lengths.

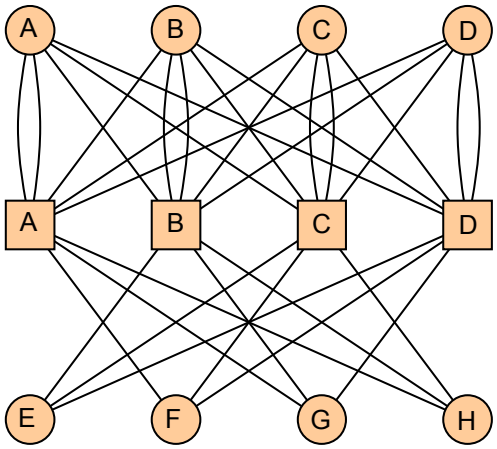


Figure 2-3: Protograph for Short Block Length LDPC Codes

In this figure, circles represent variable nodes, and squares represent check nodes, or constraint equations. To construct a full-size LDPC code, one replicates the protograph M times, in this case yielding $8M$ variable nodes (one per transmitted channel symbol), and $4M$ constraints on those channel symbols. Each edge in the protograph becomes a bundle of M edges; these are cut, cyclically permuted, and reconnected. Cyclic shifts must be chosen for each edge in the protograph; this is done using a variant of Progressive Edge Growth (PEG) (reference [C5]), a greedy algorithm that intends to maximize loop length among other metrics. It should be noted that while the protograph has parallel edges, they do not remain so when the full graph is constructed.

This process was repeated three times to generate a family of codes with three block lengths: (n=128, k=64), (n=256, k=128), and (n=512, k=128), where k is the number of information bits encoded, and n is the number of transmitted code symbols. The resulting circulant choices are listed in tables 2-4, 2-5, and 2-6. The full parity check matrix, $H_{128\times64}$, for the (n=128, k=64) code is shown in figure 2-4. Here, dots are used to indicate 1s in the parity check matrix, and lines are added to clarify the block structure.

Table 2-4: Circulant Selections for (128,64) LDPC Code

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**
A	0,7	2	14	6		0	13	0
B	6	0,15	0	1	0		0	7
C	4	1	0,15	14	11	0		3
D	0	1	9	0,13	14	1	0	

Table 2-5: Circulant Selections for (256,128) LDPC Code

	A	B	C	D	E	F	G	H
A	0,31	15	25	0		20	12	0
B	28	0,30	29	24	0		1	20
C	8	0	0,28	1	29	0		21
D	18	30	0	0,30	25	26	0	

Table 2-6: Circulant Selections for (128,64) LDPC Code

	A	B	C	D	E	F	G	H
A	0,63	30	50	25		43	62	0
B	56	0,61	50	23	0		37	26
C	16	0	0,55	27	56	0		43
D	35	56	62	0,11	58	3	0	

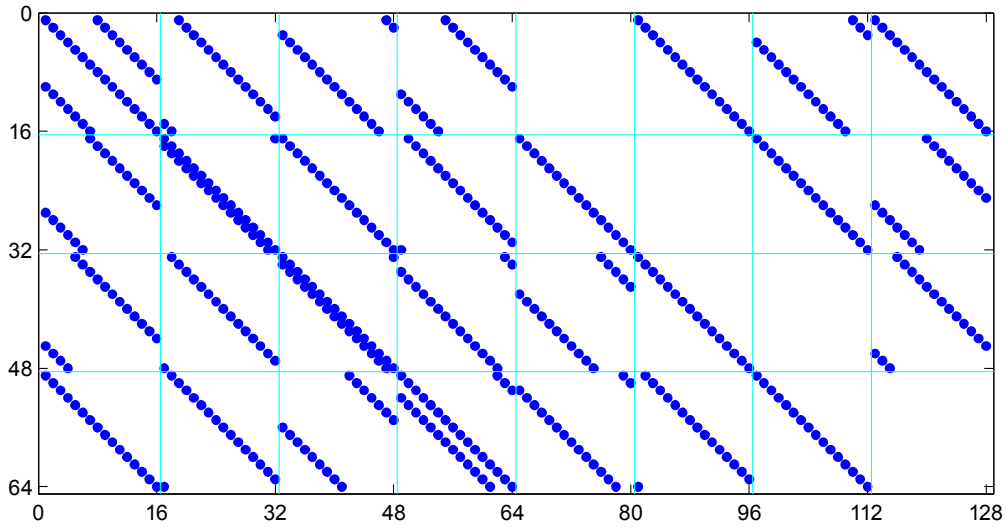


Figure 2-4: Parity Check Matrix for (128,64) LDPC Code

Some of the circulants selected are free parameters, because variable nodes and check nodes can be relabeled without changing the performance of the code. This freedom was used to put identity circulants on the main diagonal of the left half of the parity check matrix, and on the first sub-diagonal of the right half of *H* (considered as a block-matrix). This may simplify memory addressing in a hardware decoder implementation. Moreover, check equations may be reordered at will, and with the last row of circulants moved to the top, the matrix has the structure suggested by Richardson and Urbanke in reference [C14], and this may simplify encoding in some cases.

3 NON-BINARY LDPC CODES

3.1 INTRODUCTION

The vast majority of LDPC coding research has studied binary codes, though some recent results (references [C6]-[C8]) show that short non-binary LDPC codes, defined over the finite field GF(256), can offer performance improvements of 1.0 to 1.3 dB over corresponding binary LDPC codes. However, decoders for these are far more complex than decoders for the binary codes. Two designs of GF(256) LDPC codes are described in 3.2 and 3.3. LDPC codes over the smaller finite field GF(16) can offer much of the performance gain of the GF(256) codes, potentially with much less decoding complexity. Initial results for a set of GF(16) codes are given in 3.4.

Decoding of non-binary LDPC codes remains a field of research. Edge messages are probability vectors over each of the field elements. Over GF(q), each edge message consists of q probabilities, or rather q–1 degrees of freedom because the probabilities must sum to unity. Computation at the graph nodes is generally done with finite-field fast Fourier transforms, with complexity on the order of q log(q). Suboptimal decoding algorithms can reduce this complexity considerably, but with some cost in performance. In general, the complexity of practical decoding algorithms is not well understood, and this currently makes it difficult to select among different coding options.

3.2 LDPC CODES OVER GF(256) BY LIVA, PAOLINI, MATUZ, SCALISE, CHIANI

Liva, Paolini, Matuz, Scalise, and Chiani provided a structured non-binary IRA LDPC design tailored to the NGU requirements (references [C6]–[C7]). The design is based on a protograph for a rate-1/2 cycle code, depicted by a base matrix with form

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The graph has been obtained by expanding the protograph targeting a large girth. In particular, whenever possible, the resulting graph has been selected in the category of cages, i.e., graphs with a minimum number of vertices for a given girth (reference [C9]).

The choice of the coefficients in the final parity-check matrix has been performed with the aim of maximizing the minimum distance of the binary image of each check equation, following the approach proposed in reference [C10].

For the (128,64) case, the parity-check matrix has been obtained by lifting the graph with a lifting factor 4, resulting in a 16 × 8 parity-check matrix over GF(256), depicted below.

$$\mathbf{H} = \begin{bmatrix}
\alpha^{182} & 0 & 0 & 0 & \alpha^{8} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{0} \\
0 & \alpha^{89} & 0 & 0 & 0 & \alpha^{0} & 0 & 0 & 0 & \alpha^{9} & 0 & 0 & \alpha^{81} & 0 & 0 & 0 \\
0 & 0 & \alpha^{0} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^{182} & 0 & 0 & \alpha^{8} & 0 & 0 \\
0 & 0 & 0 & \alpha^{8} & 0 & 0 & 0 & \alpha^{182} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & \alpha^{0} & 0 \\
0 & 0 & \alpha^{88} & 0 & 0 & 0 & 0 & \alpha^{80} & \alpha^{0} & 0 & 0 & 0 & \alpha^{8} & 0 & 0 & 0 \\
0 & 0 & 0 & \alpha^{169} & \alpha^{0} & 0 & 0 & 0 & 0 & \alpha^{127} & 0 & 0 & 0 & \alpha^{40} & 0 & 0 \\
\alpha^{169} & 0 & 0 & 0 & 0 & \alpha^{128} & 0 & 0 & 0 & 0 & \alpha^{40} & 0 & 0 & 0 & \alpha^{0} & 0 \\
0 & \alpha^{8} & 0 & 0 & 0 & 0 & \alpha^{80} & 0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & 0 & \alpha^{0}
\end{bmatrix}$$

where α is the primitive element of the Galois field, i.e., the root of the polynomial $p(x)=1+x^2+x^3+x^4+x^8$. The performance of the code is depicted in figure 3-1.

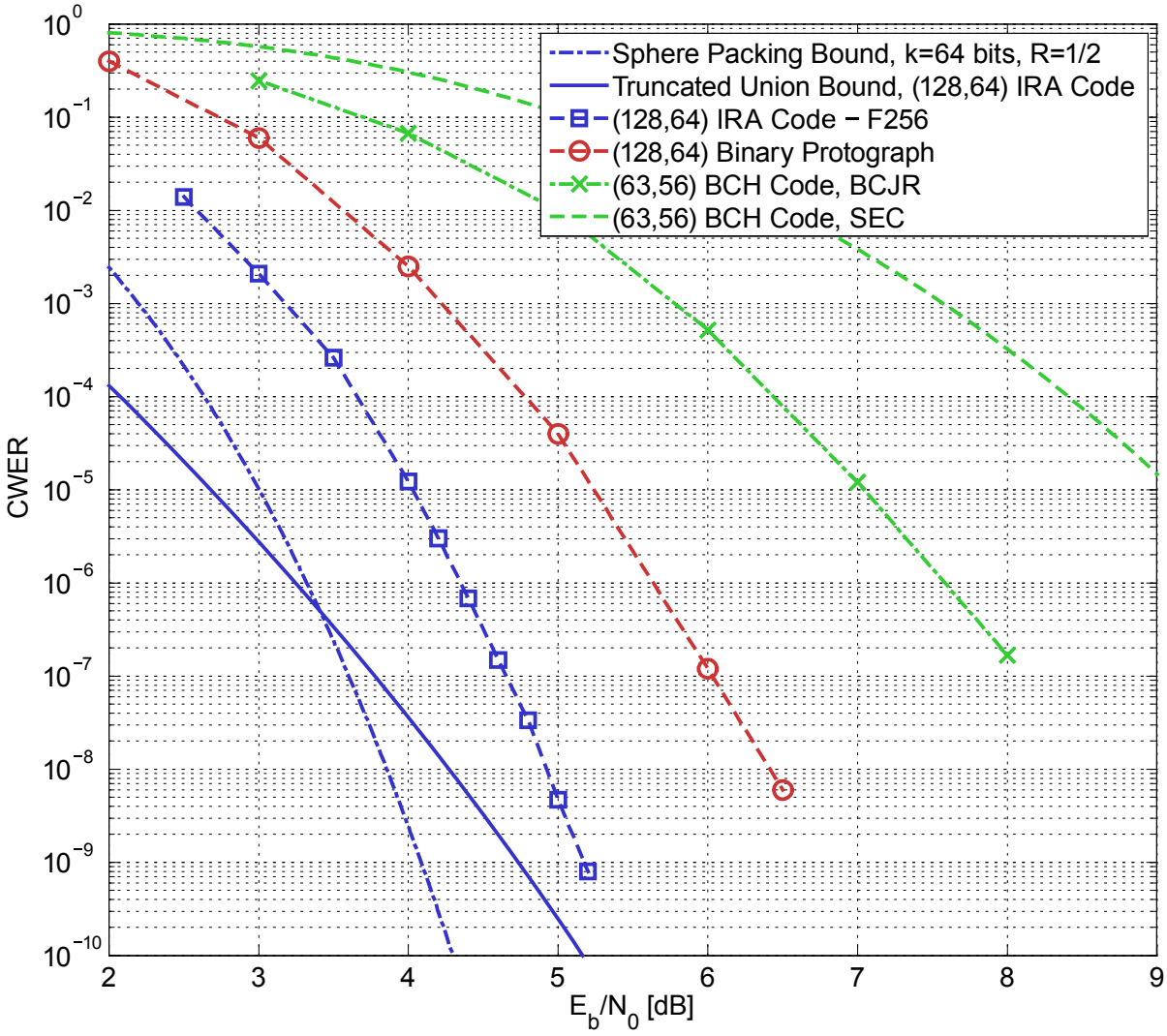


Figure 3-1: Codeword Error Rate Performance of the (128,64) Code over GF(256)

The performance of the code in terms of CWER has been simulated over the binary-input Additive White Gaussian Noise (AWGN) channel. The performance is provided in CWER versus E_b/N_o, E_b being the energy per information bit and N_o the one-sided noise power spectral density. The performance of the non-binary IRA code is compared to that of the (128, 64) binary protograph LDPC code under consideration within this document and to the sphere packing bound. The non-binary LDPC code provides a performance within 1 dB from the Sphere Packing Bound down to CWER $\simeq 4 \cdot 10^{-9}$, gaining almost 1.5 dB over the binary LDPC code. For sake of completeness, the performance of the (63, 56) BCH code in SEC mode is reported, as well as under soft-decision Bahl-Cocke-Jelinek-Raviv (BCJR) decoding (although the latter decoding algorithm is not considered by the CCSDS Recommended Standard).

3.3 LDPC CODES OVER GF(256) BY DIVSALAR AND DOLECEK

Divsalar and Dolecek have pursued protograph constructions for non-binary LDPC codes, much as described in 2.5, but now with multiplicative coefficients or 'scale factors' on each of the graph edges (references [C11] and [C12]). The coefficients may be introduced in a couple of different ways, but in each case, the code construction technique begins with the design of a protograph with unlabeled edges. As with binary codes, protograph performance in the waterfall region can be predicted with EXIT chart analysis, but here, their decoding threshold is also a function of the field size. Five small protographs are shown in figure 3-2, and their decoding thresholds are shown in figure 3-3. For reference, the capacity of the Binary-Input Additive White Gaussian Noise (BIAWGN) channel is 1/2 bit per channel use at E_b/N_o=0.187 dB. The figure shows that with the proper choice of field size, these protographs have thresholds within a small fraction of a dB of capacity. EXIT chart analysis assumes infinite block size, and does not predict error floor behavior, so protograph selection depends on experimentation and code simulation in addition to analysis.

Finite field coefficients can be added to a protograph in a couple of ways. The most flexible is to expand the protograph into a full Tanner graph using the standard copy-and-permute procedure (using cyclic permutations), and then to assign field elements to each edge in the graph. This procedure is shown schematically in figure 3-4, and is called the Unconstrained Non-Binary Protograph Based (U-NBPB) method of code construction. Alternatively, finite field elements may be assigned to the edges of the protograph, and then copied with the edges as the protograph is expanded into a full Tanner graph, as shown in figure 3-5. This approach, called the Constrained Non-Binary Protograph Based (C-NBPB) method, has less code design freedom, but simplifies the coefficient selection process, and may simplify LDPC decoding.

These non-binary protographs are expanded with circulants, as are the binary codes. Circulant selection is done to minimize the number of small loops. Due both to the small final code sizes and large finite field size, the protograph expansion factor is small, only 4, 8, and 16 for the cases considered here. This limits the choices of circulants, and the girths of the graphs that can be achieved.

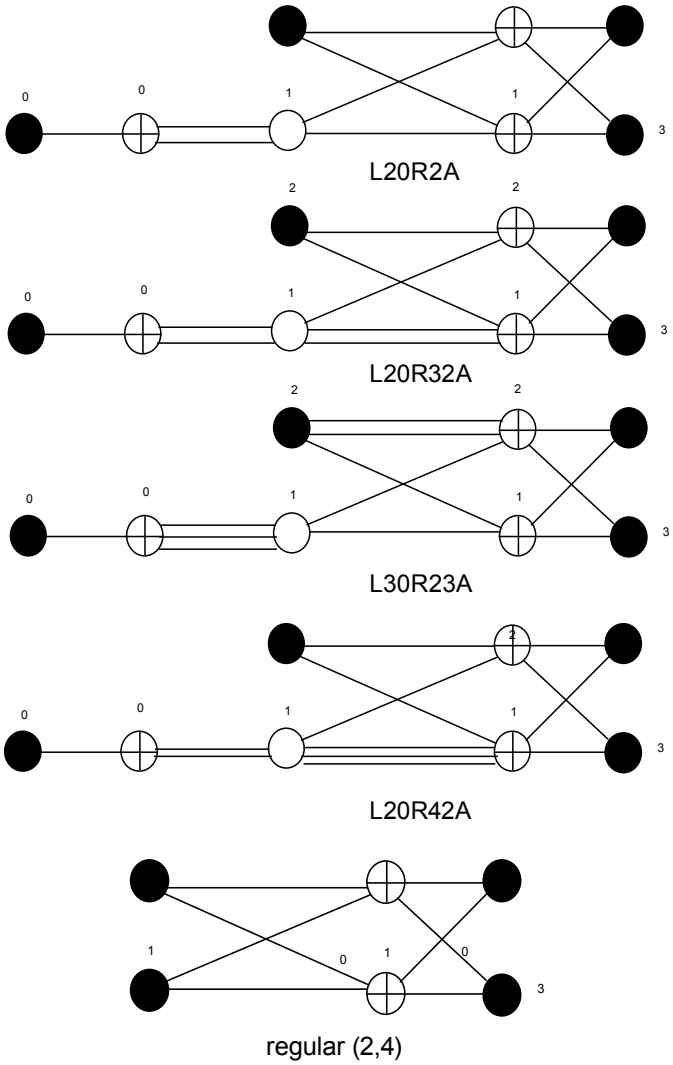


Figure 3-2: Five Protographs for Non-Binary LDPC Codes

With the field GF(256), the regular (2,4) protograph was selected, as shown in figure 3-2. As an unlabeled protograph matrix, this is written,

$$H_p = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

To construct a (128,64) binary code, this protograph is expanded by a factor of 4, by using circulants of size 4×4. If

$$\sigma = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

then circulants can be denoted as powers of σ. The circulants selected for the (128,64) code are

$$H_{(128,64)} = \begin{bmatrix} \sigma^0 & \sigma^1 & \sigma^3 & \sigma^2 \\ \sigma^0 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix}$$

Similarly,

$$H_{(256,128)} = \begin{bmatrix} \sigma^0 & \sigma^3 & \sigma^6 & \sigma^4 \\ \sigma^1 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix} \text{ and } H_{(512,256)} = \begin{bmatrix} \sigma^0 & \sigma^7 & \sigma^{12} & \sigma^8 \\ \sigma^2 & \sigma^0 & \sigma^0 & \sigma^0 \end{bmatrix}$$

In the U-NBPB code construction method, these matrices are written out, and then GF(256) field elements are selected for each nonzero entry. For the (128,64) case, the result is,

$$H = \begin{bmatrix} \alpha^0 & 0 & 0 & 0 & 0 & \alpha^{89} & 0 & 0 & 0 & 0 & 0 & \alpha^{81} & 0 & 0 & \alpha^9 & 0 \\ 0 & \alpha^8 & 0 & 0 & 0 & 0 & \alpha^0 & 0 & \alpha^{182} & 0 & 0 & 0 & 0 & 0 & 0 & \alpha^{173} \\ 0 & 0 & \alpha^{173} & 0 & 0 & 0 & 0 & \alpha^8 & 0 & \alpha^0 & 0 & 0 & \alpha^{183} & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha^8 & \alpha^0 & 0 & 0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & \alpha^{80} & 0 & 0 \\ \alpha^{183} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^8 & 0 & 0 & 0 & \alpha^0 & 0 & 0 & 0 \\ 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{88} & 0 & 0 & 0 & \alpha^{80} & 0 & 0 & 0 & \alpha^8 & 0 & 0 \\ 0 & 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{167} & 0 & 0 & 0 & \alpha^{127} & 0 & 0 & 0 & \alpha^{40} & 0 \\ 0 & 0 & 0 & \alpha^0 & 0 & 0 & 0 & \alpha^{182} & 0 & 0 & 0 & \alpha^{173} & 0 & 0 & 0 & \alpha^8 \end{bmatrix}$$

where α is a primitive element of GF(256). Finally, this is converted into a binary code by expansion of the field elements. By using the primitive polynomial $p(x) = 1 + x^2 + x^3 + x^4 + x^8$, α may also be represented as the 8×8 matrix,

$$\alpha = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Substituting powers of this matrix into H gives the full binary parity check matrix. Performance curves for three U-NBPB codes are shown in figure 3-6, with performances of the corresponding binary codes included for comparison.

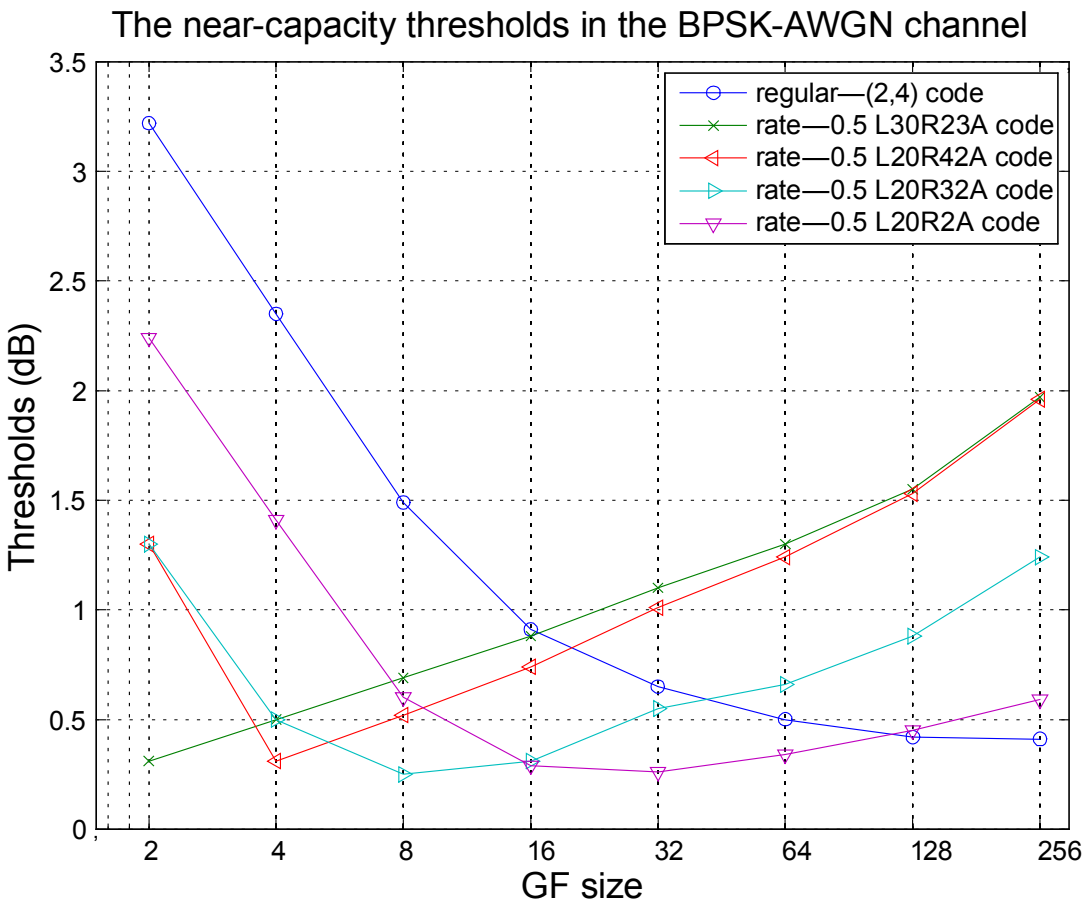


Figure 3-3: Thresholds of Five Protographs with the BIAWGN Channel

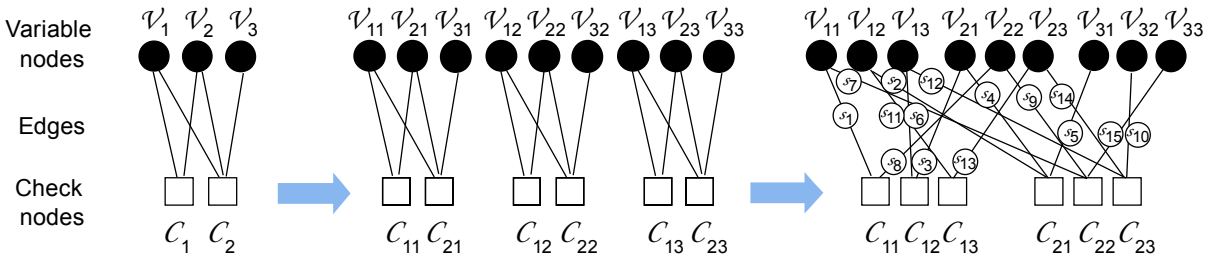


Figure 3-4: Unconstrained Non-Binary Protograph-Based Code Construction

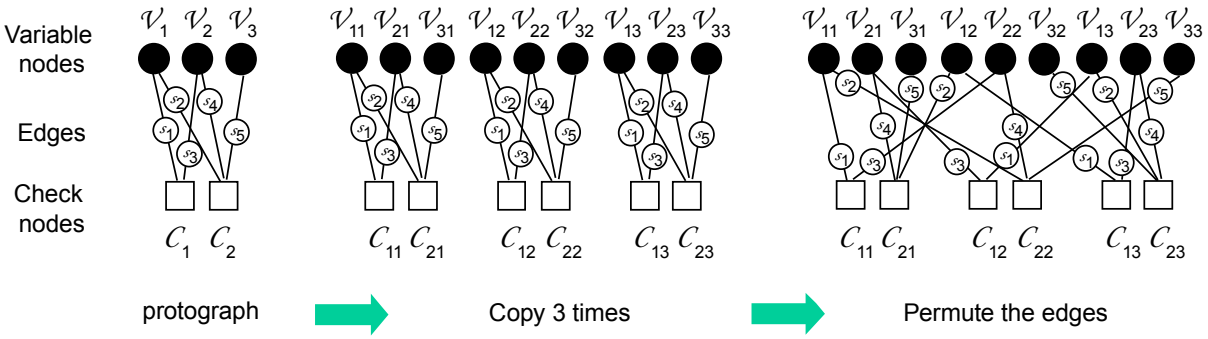


Figure 3-5: Constrained Non-Binary Protograph-Based Code Construction

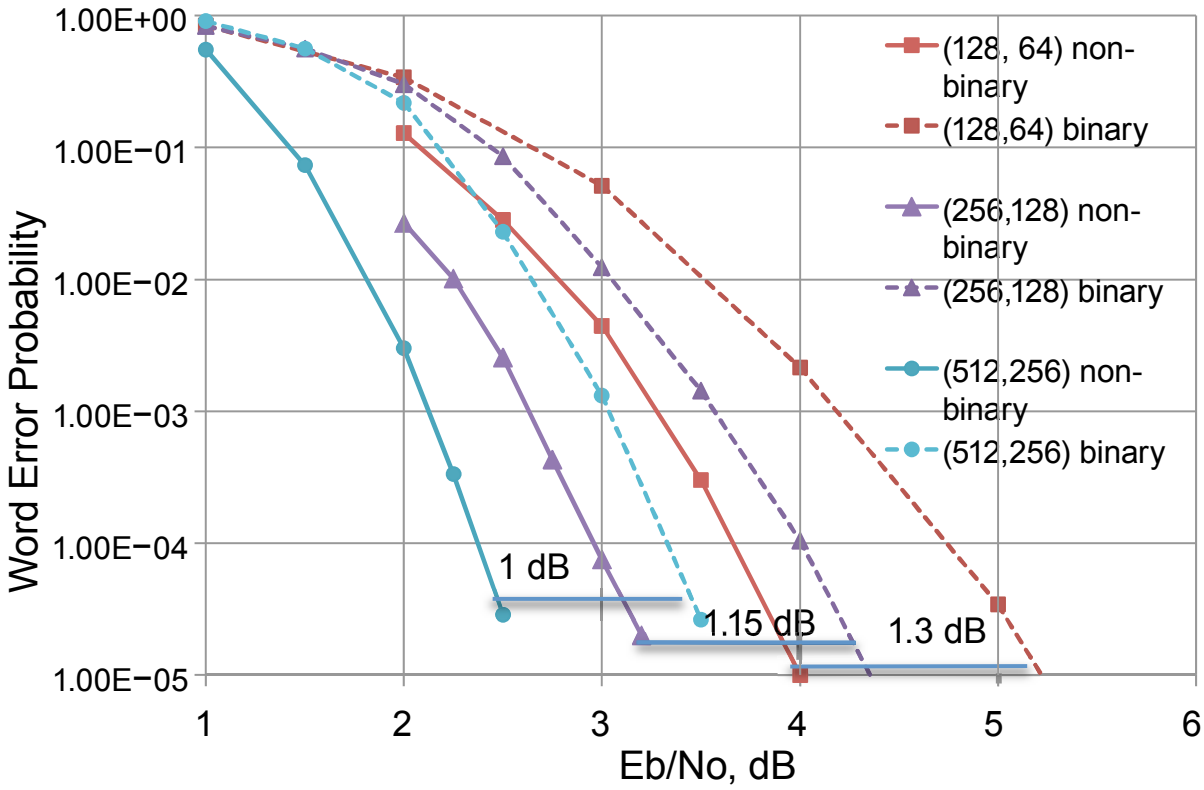


Figure 3-6: Performance of U-NBPB Codes over GF(256)

Constrained non-binary protograph based LDPC codes are constructed by substituting the elements of H_p with powers of α. Those chosen here are,

$$H = \begin{bmatrix} \alpha^{0} & \alpha^{80} & \alpha^{88} & \alpha^{8} \\ \alpha^{89} & \alpha^{9} & \alpha^{0} & \alpha^{81} \end{bmatrix}$$

This matrix is then copy-and-permuted using the same circulants as in the U-NBPB case. This provides a complete description of three GF(256) LDPC codes, and their performances are given in figure 3-7. It should be noted that these are marginally inferior to those in figure 3-6, because of the extra constraints on the code design.

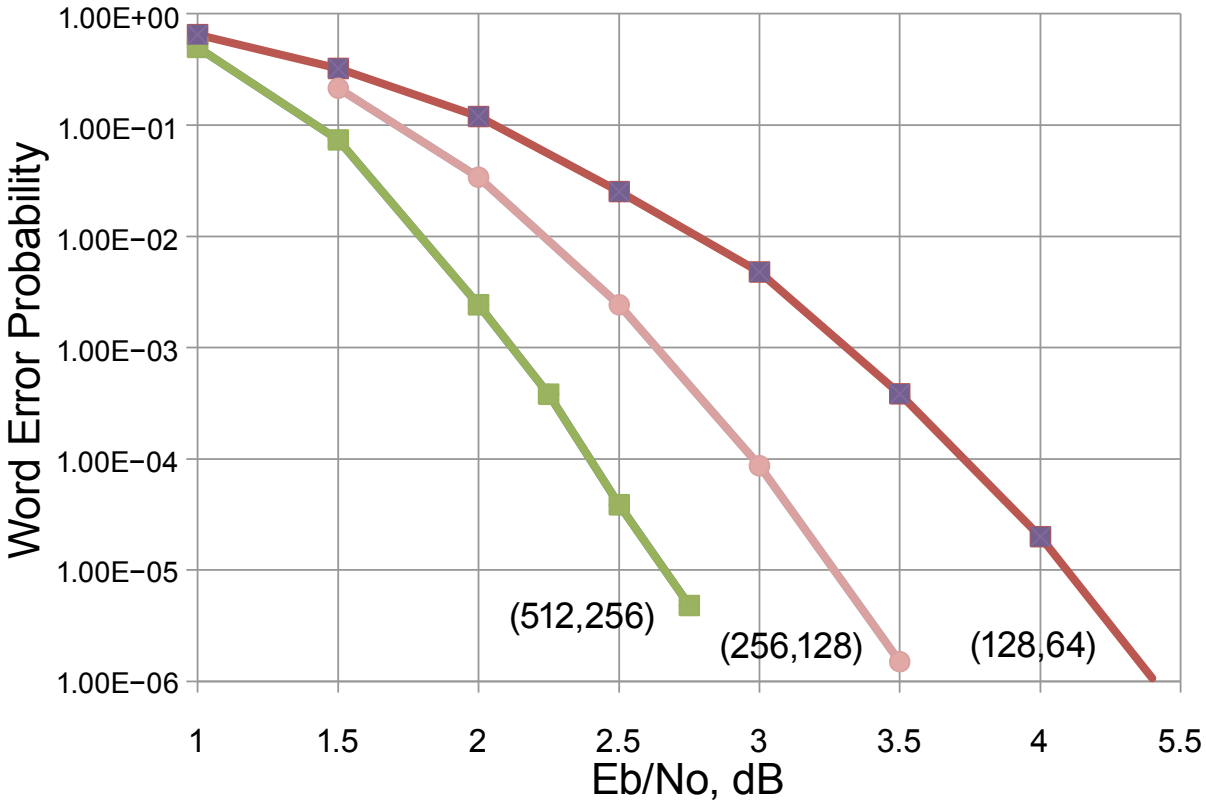


Figure 3-7: Performance of C-NBPB Codes over GF(256)

3.4 LDPC CODES OVER GF(16) BY DIVSALAR AND DOLECEK

In a search for lower complexity codes, Divsalar and Dolecek have performed initial studies of LDPC codes over GF(16) (references [C11] and [C12]). The same U-NBPB construction method described for GF(256) is used again here. Performance curves are shown for codes with binary sizes (128,64), (256,128), and (512,256) in figure 3-8, with both the binary and the GF(256) LDPC codes for comparison.

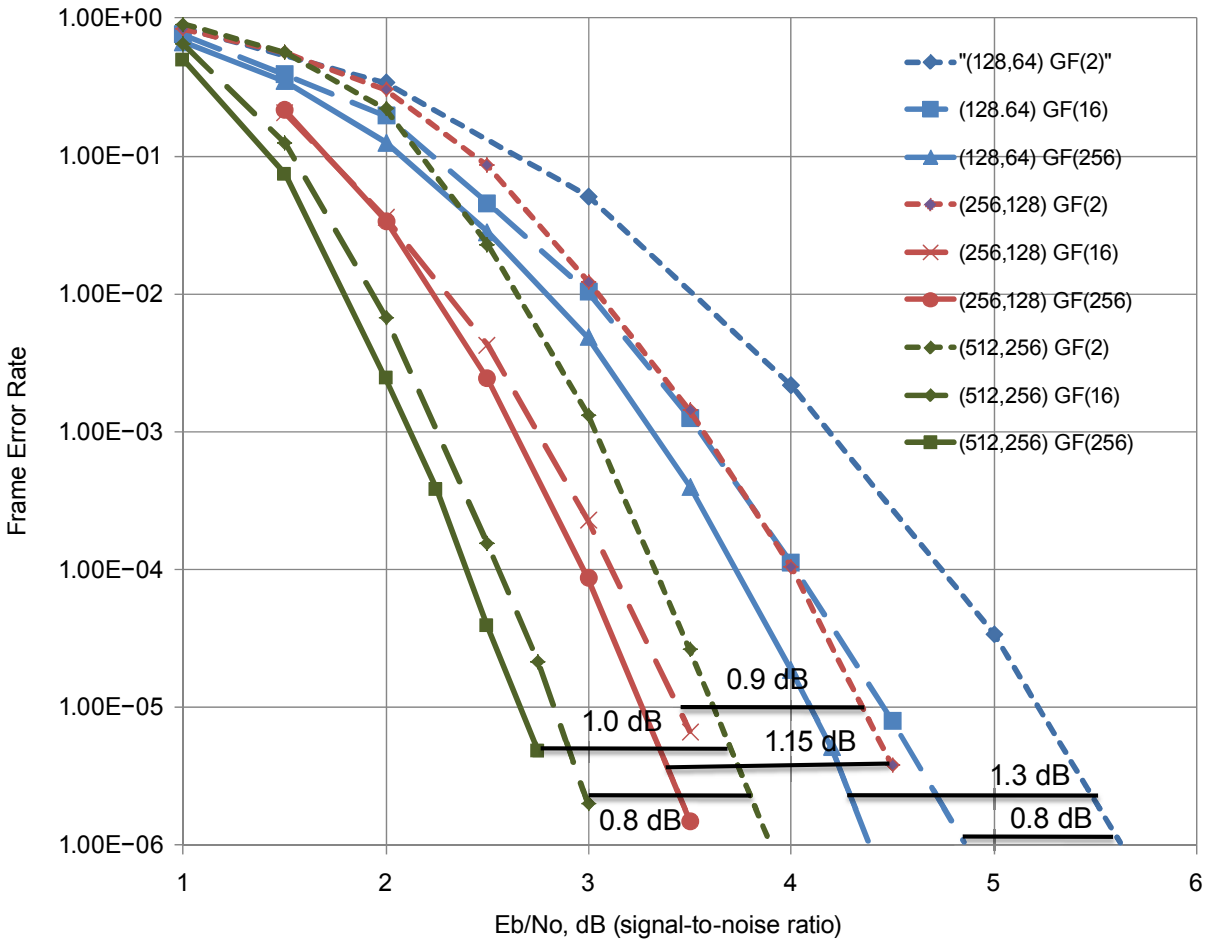


Figure 3-8: Performance of U-NBPB LDPC Codes over GF(16)

The GF(16) codes achieve most of the performance benefits of the GF(256) codes, especially at the longer block lengths. Supposing that decoder complexity is reasonably approximated as $q \log_2(q)$, then decoders for these codes can be expected to be about 1/32 times as complex as those for the larger field. However, the number of required iterations is not yet well characterized, and as with all LDPC codes, there are a wealth of suboptimal decoding algorithms to explore.

ANNEX A

BINARY LDPC CODE IMPLEMENTATION

Encoders and decoders for these three binary LDPC codes have been implemented on a Field Programmable Gate Array (FPGA). As with most FPGA problems, the speed of an LDPC encoder or decoder will scale essentially linearly with FPGA area. The designs reported here were intended to be small and not especially fast. Each is small enough that it would use only a few percent of any FPGA that might be used in a spacecraft design (such as the Xilinx radiation-tolerant XQR2V1000), and will support any data rate that is anticipated for these codes. The encoders produce one code symbol per clock cycle; the decoders use 26 clock cycles per iteration per information bit. Speed and area results are given in table A-1 and table A-2 for implementations on an FPGA in the Xilinx Virtex-2 family.

Table A-1: Implementation Results for LDPC Encoders on Xilinx Virtex-2 FPGAs

Encoder	Look-Up Tables	Max Clock Rate	symbols/sec	bits/sec
(n=128, k=64)	89	200 MHz	200 Ms/s	100 Mb/s
(n=256, k=128)	155	230 MHz	200 Ms/s	100 Mb/s
(n=512, k=256)	155	230 MHz	200 Ms/s	100 Mb/s

Table A-2: Implementation Results for LDPC Decoders on Xilinx Virtex-2 FPGAs

Decoder	Look-up Tables	Max Clock Rate	Iterations at WER=10^{-6}	bits/sec
(n=128, k=64)	521	70 MHz	2.3	1170 kb/s
(n=256, k=128)	521	70 MHz	3.1	868 kb/s
(n=512, k=256)	558	70 MHz	4.1	656 kb/s

ANNEX B

INTERACTION OF SHORT UPLINK CODES WITH OTHER TC PROTOCOL LAYERS

B1 INTERACTION WITH OTHER ELEMENTS WITHIN THE CODING SUBLAYER

The TC standard was developed with the expectation that the transfer frames from the data link protocol sublayer would fill one or a few BCH codewords, and modestly larger data volumes were accommodated with the addition of PLOP-2 to the protocol. This is the same application profile for which the short block length LDPC codes are designed (low-data-rate command and emergency communications), and so the same design remains appropriate. For next-generation uplink applications that involve larger data volumes and higher speeds, a different set of longer block length LDPC codes is proposed, and surrounding protocol should also be modified to be similar to the CCSDS Telemetry standards. In this document, attention is restricted to the short block length codes.

The coding and synchronization sublayer of the telecommand protocol consists of four operations, as shown in figure B-1, from the perspective of the transmitter. The new LDPC codes are proposed as a direct substitution for the current BCH code, with the single modification that the randomizer is mandatory with the LDPC codes. In the remainder of this annex, implications are more carefully considered.

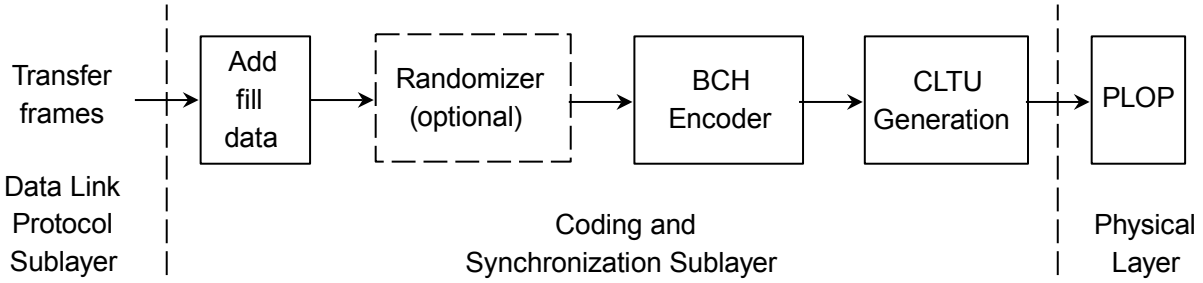


Figure B-1: Components of the Coding and Synchronization Sublayer

With either the BCH code or the LDPC codes, fill data is added if necessary to complete the last codeword, and it may be added either before or after randomization. It is suggested that the fill data is added first as shown in figure B-1 for a couple of minor reasons. First this is the better inverse of the receive side that always derandomizes before passing the fill data up to the TC data link protocol sublayer; second, this reduces the minor spectral effects that the fill data may introduce.

The randomizer is mandatory with the LDPC codes, because the codewords may be longer, and longer transition-free runs are possible in the unrandomized symbol stream. It should be noted that the parity symbols of the BCH codewords are inverted to prevent long runs. When LDPC codes are used, the randomizer is mandatory, and the parity symbols are not inverted.

CLTU generation (the addition of the start sequence and tail sequence) is performed with LDPC codewords exactly in the same way as with BCH codewords.[2] It should be noted that there are no synchronization markers between LDPC codewords, as there is in the TM standard. This is acceptable because resynchronization is accomplished at the end of each CLTU. Markerless codeword synchronization may be used if desired for marginally improved performance.

The following subsections address the various implications of the code design on the CCSDS physical layer and upper layers. The case of binary LDPC codes is explored, although the considerations have a broader scope and hold also in the case of the other coding families (e.g., non-binary LDPC codes) drawn in this Experimental Specification.

B2 IMPACT ON THE RADIO FREQUENCY AND MODULATION LAYER

The proposed LDPC codes can operate at a lower Signal-to-Noise Ratio (SNR) than the current (63,56) BCH code for several reasons: the code rate is reduced to 1/2 from 0.89, they are typically decoded with a soft-decision decoder which saves about 2 dB as a rule of thumb, and all three LDPC codes have longer codeword lengths than the BCH code. These changes need not have any significant impact on the transmit side of an uplink communications link, but there are two notable impacts on the receive side.

a) It will be advantageous for the spacecraft's radio receiver to work at a lower symbol SNR than with the current BCH code. By analogy, a communications system is only as capable as its weakest link. The BCH code is currently a weak link, and by replacing it with the stronger LDPC code, the system becomes more capable. The next-weakest link may be the SNR at which the receiver can maintain symbol synchronization, and in this case, further performance gains may be won by improving the receiver's tracking loops. If the receiver cannot be improved though, of course the use of the modern codes can only be an improvement. If one aims to achieve a particular Word Error Rate (WER), the potential gains are shown in figure B-2, where the horizontal axis shows the symbol-SNR, rather than the more common bit-SNR. If the receiver can maintain synchronization at an SNR below that required by an LDPC code, then the full power of the code can be used; otherwise the benefits are smaller. Similarly, if one has a constraint on the Undetected Word Error Rate (UER), the potential gains are shown in figure B-3.

[2] Because the 16-symbol start sequence may not be detectable with an acceptably high probability, it may be advantageous to assure that the first LDPC codeword transmitted contains entirely idle data.

b) An LDPC code is generally decoded using 'soft symbols', rather than the binary 'hard symbols' typically used for a BCH code. This provides a performance improvement of about 2 dB, but depends on a receiver that can produce soft outputs. This modification is not mandatory, however, for a belief propagation decoder can also operate on the hard symbols if necessary. Conversely, BCH codes are typically decoded with an algebraic decoder that operates on binary inputs, but there are soft-decision BCH decoding algorithms that can provide a performance improvement at a substantial cost in complexity.

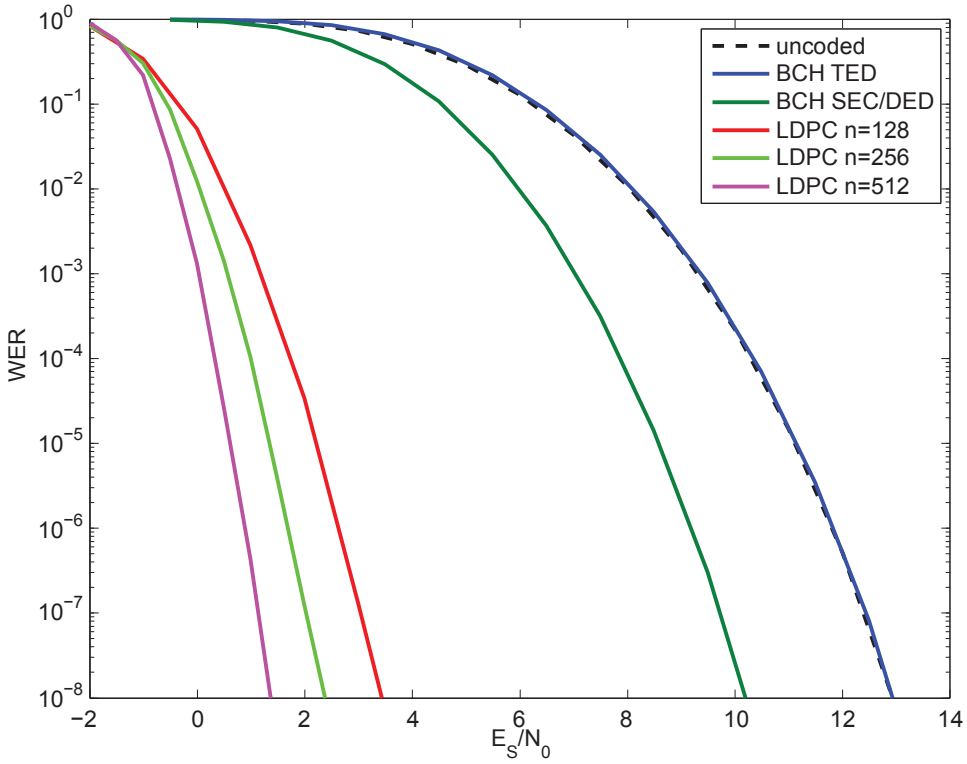


Figure B-2: Word Error Rates for Several Uplink Coding Schemes

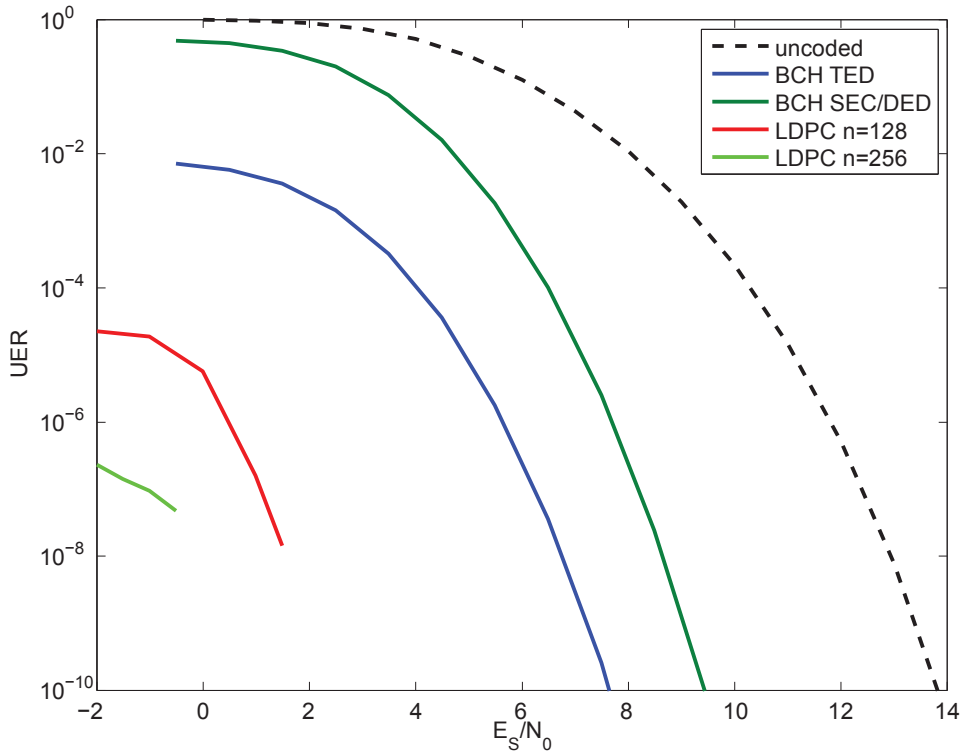


Figure B-3: Undetected Error Rates for Several Coding Schemes

Currently, CCSDS standards specify that telecommand uses one of two Physical Layer Operations Procedures (PLOPs). All missions whose planning began after September 2010 are to use PLOP-2, as shown in figure B-4; the older PLOP-1 differs only in that the loop returns to Carrier Modulation Mode 1 (CMM-1) instead of to CMM-3. No modification is required to use the PLOP-2 protocol and the short block length LDPC codes together. Any additional changes to the RFM protocol layer are at the discretion of the Next Generation Uplink working group.

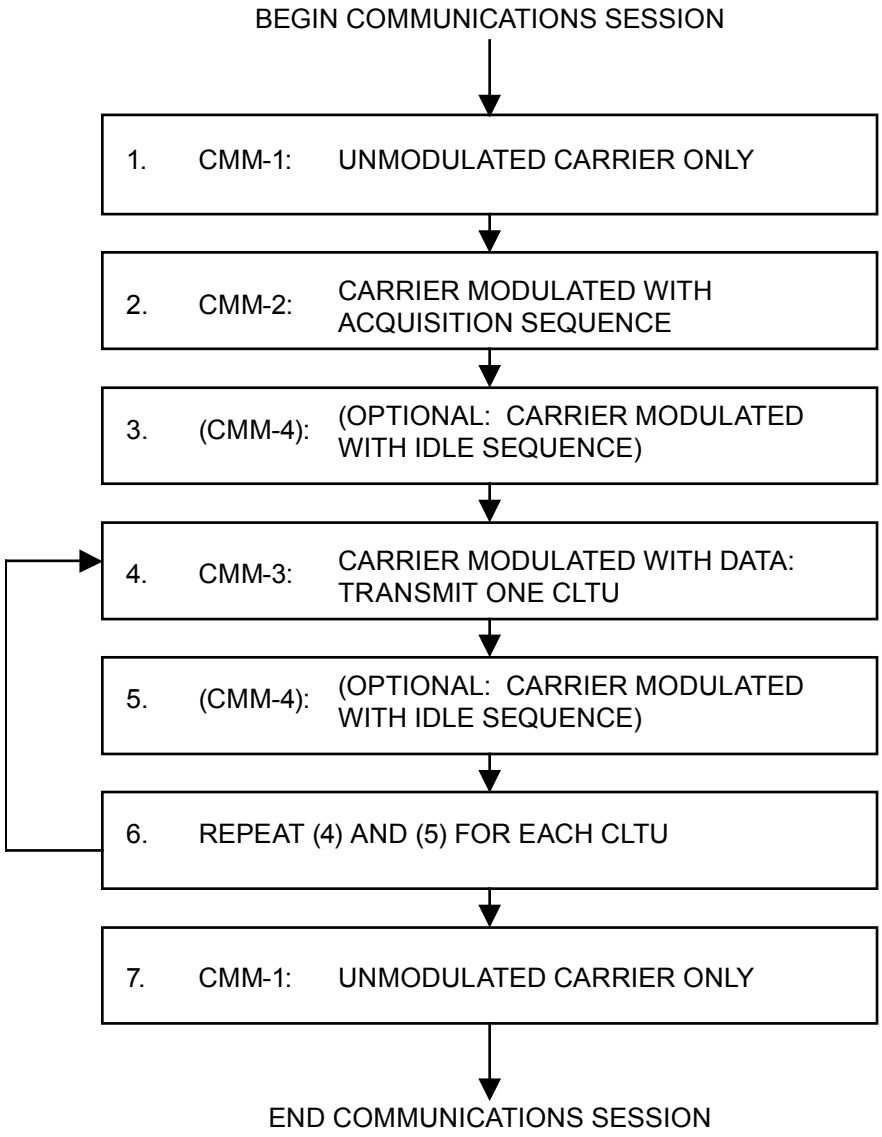


Figure B-4: PLOP-2 Telecommand Protocol[3]

[3] Reproduced from figure 6-2 of reference [2]

B3 INTERACTIONS WITH UPPER PROTOCOL LAYERS

The Coding and Synchronization sublayer exchanges variable length TC Transfer Frames with the TC Space Data Link Protocol (reference [2]). Because these are of variable length, the protocol sublayers are almost completely isolated from each other. Further to this, it must be noted that the service offered to the CCSDS Data Link sublayer remains fully compliant with the annex A of reference [1], thus implying that also the protocol interface defined between the two sublayer is not affected by the code replacement.

As to the specific functions implemented by the upper layer, a summary of the protocol operations occurring when the proposed LDPC codes are used is reported as follows:

- **All Frames Generation Function**. It is responsible for computing the Frame Error Control Field (FECF), more commonly known as a Cyclic Redundancy Check (CRC), and passing the resulting Transfer Frames to the coding sublayer. Its behavior, like those of the functions above it, remains unchanged.
- **All Frames Reception Function**. It is responsible for stripping fill data from TC Transfer Frames, and validating the resulting frames. Because the LDPC codewords can be up to 64 octets long, as many as 63 octets of fill may have to be stripped from the end of a TC Transfer Frame. As with the BCH coding scheme, there is the possibility that the fill may be mistaken for a Transfer Frame Header, but for the same reasons, it will fail the standard frame validation check procedure.

Any additional changes to the TC Space Data Link Protocol sublayer are at the discretion of the Space Link Protocol (SLP) working group, and are independent of the error correcting codes. Perhaps the most compelling is the opportunity to modify the go-back-*n* retransmission scheme to improve its efficiency and storage demands.

ANNEX C

INFORMATIVE REFERENCES

[C1] S. Dolinar, et al. “Bounded Angle Iterative Decoding of LDPC Codes.” In *Proceedings of MILCOM 2008 (16–19 Nov. 2008, San Diego, CA)*, 1–6. New York: IEEE, 2008.

[C2] S. Dolinar, et al. “Bounds on Error Probability of Block Codes with Bounded-Angle Maximum-Likelihood Incomplete Decoding.” In *Proceedings of ISITA 2008 (7–10 Dec. 2008, Auckland, New Zealand)*, 1–6. New York: IEEE, 2008.

[C3] J. Thorpe. “Low-Density Parity-Check (LDPC) Codes Constructed from Protographs.” *IPN Progress Report* 42-154 (August 15, 2003).

[C4] T. Richardson. “Error Floors of LDPC Codes.” In *Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing (1–3 Oct. 2003, Monticello, Illinois)*, 1425–1435.

[C5] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold. “Irregular Progressive Edge-Growth (PEG) Tanner Graphs.” In *Proceedings of 2002 IEEE International Symposium on Information Theory*, 480. New York: IEEE, 2002.

[C6] G. Liva, et al. “Turbo Codes Based on Time-Variant Memory-1 Convolutional Codes over F_q.” In *2011 IEEE International Conference on Communications (ICC) (5–9 June 2011, Kyoto, Japan)*, 1–6. New York: IEEE, 2011.

[C7] G. Liva, et al. “Short Turbo Codes over High Order Fields.” *IEEE Transactions on Communications* 61, no. 6 (June 2013): 2201–2211.

[C8] T. De Cola, et al. “Reliability Options for Data Communications in the Future Deep-Space Missions.” *Proceedings of the IEEE* 99, no. 11 (November 2013): 2056–2074.

[C9] A. Venkiah, D. Declercq, and C. Poulliat. “Design of Cages with a Randomized Progressive Edge-Growth Algorithm.” *IEEE Communications Letters* 12, no. 4 (April 2008): 301–303.

[C10] C. Poulliat, M. Fossorier, and D. Declercq. “Design of Regular $(2,d_c)$-LDPC Codes over GF(q) Using Their Binary Images.” *IEEE Transactions on Communications* 56, no. 10 (October 2008): 1626–1635.

[C11] D. Divsalar and L. Dolecek. “Graph Cover Ensembles of Non-Binary Protograph LDPC Codes.” In *Proceedings of 2012 IEEE International Symposium on Information Theory Proceedings (ISIT) (1–6 July 2012, Cambridge, Massachusetts)*, 2526–2530. New York: IEEE, 2012.

[C12] L. Dolecek, et al. "Non-Binary Protograph-Based LDPC Codes: Enumerators, Analysis, and Designs." *IEEE Transactions on Information Theory* 60, no. 7 (July 2014): 3913–3941.

[C13] K.S. Andrews, et al. "The Development of Turbo and LDPC Codes for Deep-Space Applications." *Proceedings of the IEEE* 95, no. 11 (Nov. 2007): 2142–2156.

[C14] T.J. Richardson and R.L. Urbanke. "Efficient Encoding of Low-Density Parity-Check Codes." *IEEE Transactions on Information Theory* 47, no. 2 (Feb 2001): 638–656.

ANNEX D

ABBREVIATIONS

ARQ	automatic repeat queuing
AWGN	additive white gaussian noise
BCH	Bose-Chaudhuri-Hocquenghem
BCJR	Bahl-Cocke-Jelinek-Raviv
BIAWGN	binary-input additive white gaussian noise
CLTU	communications link transmission unit
C-NBPB	constrained non-binary protograph based
CRC	cyclic redundancy check
CWER	codeword error rate
DSN	Deep Space Network
FECF	frame error control field
FPGA	field programmable gate array
LDPC	low density parity check
MSB	most significant bit
MSPA	multiple spacecraft per antenna
NGU	Next Generation Uplink
PLOPs	physical layer operations procedures
RFM	radio frequency and modulation
SER	symbol error rate
SLP	space link protocol
SNR	signal-to-noise ratio
TC	telecommand
U-CWER	undetected codeword error rate
UER	undetected word error rate
U-NBPB	unconstrained non-binary protograph based
WER	word error rate