

**Report Concerning Space Data System Standards**

**TC SYNCHRONIZATION AND  
CHANNEL CODING—  
SUMMARY OF CONCEPT  
AND RATIONALE**

**INFORMATIONAL REPORT**

**CCSDS 230.1-G-1**

**GREEN BOOK**

**June 2006**

## AUTHORITY

Issue:	Informational Report, Issue 1
Date:	June 2006
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and reflects the consensus of technical panel experts from CCSDS Member Agencies. The procedure for review and authorization of CCSDS Reports is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*.

This document is published and maintained by:

CCSDS Secretariat  
Office of Space Communication (Code M-3)  
National Aeronautics and Space Administration  
Washington, DC 20546, USA

## FOREWORD

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Report is therefore subject to CCSDS document management and change control procedures which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. 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 addressed to the CCSDS Secretariat at the 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.
- British National Space Centre (BNSC)/United Kingdom.
- Canadian Space Agency (CSA)/Canada.
- Centre National d’Etudes Spatiales (CNES)/France.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (Roskosmos)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSP0)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- Centro Tecnico Aeroespacial (CTA)/Brazil.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish Space Research Institute (DSRI)/Denmark.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- 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.
- MIKOMTEK: CSIR (CSIR)/Republic of South Africa.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic & Atmospheric Administration (NOAA)/USA.
- National Space Organization (NSPO)/Taipei.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

## DOCUMENT CONTROL

<b>Document</b>	<b>Title and Issue</b>	<b>Date</b>	<b>Status</b>
CCSDS 230.1-G-1	TC Synchronization and Channel Coding—Summary of Concept and Rationale, Informational Report, Issue 1	June 2006	Current issue

## CONTENTS

<u>Section</u>	<u>Page</u>
<b>1 INTRODUCTION</b> .....	<b>1-1</b>
1.1 PURPOSE.....	1-1
1.2 SCOPE.....	1-1
1.3 ORGANIZATION.....	1-1
1.4 REFERENCES.....	1-2
<b>2 OVERVIEW</b> .....	<b>2-1</b>
2.1 CONTEXT.....	2-1
2.2 PURPOSE.....	2-2
2.3 PROCEDURES AT THE SENDING END.....	2-3
2.4 PROCEDURES AT THE RECEIVING END.....	2-4
<b>3 BCH ENCODING</b> .....	<b>3-1</b>
3.1 INTRODUCTION.....	3-1
3.2 FORMAL INTERFACE TO DATA LINK PROTOCOL SUBLAYER.....	3-1
3.3 FORMAT OF A BCH CODEBLOCK.....	3-2
3.4 BREAKING THE FRAMES INTO FIXED-LENGTH PIECES.....	3-3
3.5 THE BCH CODE.....	3-4
3.6 BCH ENCODING.....	3-5
<b>4 CLTU GENERATION</b> .....	<b>4-1</b>
4.1 INTRODUCTION.....	4-1
4.2 LENGTH OF A CLTU.....	4-1
4.3 FORMAT OF A CLTU.....	4-1
4.4 START SEQUENCE.....	4-2
4.5 TAIL SEQUENCE.....	4-3
<b>5 PHYSICAL LAYER OPERATIONS PROCEDURES</b> .....	<b>5-1</b>
5.1 GENERAL.....	5-1
5.2 ELEMENTS.....	5-1
5.3 PLOP-1.....	5-3
5.4 PLOP-2.....	5-4
5.5 SELECTION OF PLOP-1 OR PLOP-2.....	5-5
5.6 FORMAL INTERFACE TO SUBLAYER ABOVE.....	5-6

**CONTENTS (continued)**

<u>Section</u>	<u>Page</u>
<b>6 CLTU RECEPTION PROCEDURES .....</b>	<b>6-1</b>
6.1 INTRODUCTION .....	6-1
6.2 CLTU RECEPTION LOGIC.....	6-1
6.3 INACTIVE STATE.....	6-2
6.4 SEARCH STATE .....	6-3
6.5 DECODE STATE.....	6-4
6.6 INTERFACE TO DATA LINK PROTOCOL SUBLAYER .....	6-11
6.7 RELATIONSHIP TO THE PLOPS.....	6-13
<b>7 PSEUDO-RANDOMIZATION .....</b>	<b>7-1</b>
7.1 OVERVIEW .....	7-1
7.2 RELATIONSHIP TO OTHER PROCEDURES .....	7-1
7.3 RANDOMIZATION PROCEDURE.....	7-2
7.4 BACKGROUND .....	7-4
<b>8 PERFORMANCE DATA.....</b>	<b>8-1</b>
8.1 INTRODUCTION .....	8-1
8.2 PERFORMANCE CRITERIA .....	8-1
8.3 PERFORMANCE COMPONENTS.....	8-1
8.4 FACTORS AFFECTING FRAME REJECTION RATE .....	8-2
8.5 FACTORS AFFECTING FRAME UNDETECTED ERROR RATE.....	8-18
<b>ANNEX A GLOSSARY .....</b>	<b>A-1</b>
<b>ANNEX B ACRONYMS AND ABBREVIATIONS.....</b>	<b>B-1</b>
<b>ANNEX C THEORETICAL BACKGROUND OF THE RANDOMIZATION SEQUENCE .....</b>	<b>C-1</b>
<b>ANNEX D THEORETICAL BACKGROUND OF THE CLTU TAIL SEQUENCE .....</b>	<b>D-1</b>
<b>ANNEX E PERFORMANCE OF OBSOLETE FEATURES .....</b>	<b>E-1</b>
<b>ANNEX F PRACTICAL EXAMPLES OF FRAMES AND CLTUS .....</b>	<b>F-1</b>

Figure

2-1 Related Standards and Layers.....	2-1
2-2 Procedures at the Sending End .....	2-3
2-3 Procedures at the Receiving End .....	2-4
3-1 Format of a BCH Codeblock .....	3-2
3-2 Example of Breaking ‘Frames’ into BCH Codeblocks .....	3-4

**CONTENTS (continued)**

<u>Figure</u>	<u>Page</u>
4-1 Format of a CLTU .....	4-1
4-2 Example of BCH Codeblocks in a CLTU .....	4-2
4-3 Start Sequence .....	4-2
5-1 PLOP-1 Carrier Modulation Modes .....	5-4
5-2 PLOP-2 Carrier Modulation Modes .....	5-5
6-1 State Diagram for CLTU Reception Logic.....	6-2
6-2 Inverse of the Start Sequence Bit Pattern .....	6-3
6-3 An Implementation for the Error Correction Mode Decoder .....	6-7
7-1 Logic Diagram for the Bit Transition Generator .....	7-3
8-1 Frame Rejection Probability in TED Mode, $P_{FX}$ or $P_{F2X}$ , for the Last or Only Frame in a CLTU (PLOP-1 or PLOP-2).....	8-13
8-2 Frame Rejection Probability in SEC Mode, $P_{FY}$ , for the Last or Only Frame in an Independent CLTU (PLOP-1) .....	8-14
8-3 Frame Rejection Probability in SEC Mode, $P_{F2Y}$ , for the Last or Only Frame in a CLTU in a Sequence of CLTUs (PLOP-2).....	8-17
8-4 Probability of Undetected Error in a Frame in TED Mode .....	8-24
8-5 Probability of Undetected Error in a Frame in SEC Mode.....	8-25

Table

3-1 Bandwidth Overhead for Different Codeblock Lengths.....	3-3
5-1 Carrier Modulation Modes .....	5-2
6-1 State Table for CLTU Reception Logic.....	6-2
6-2 Codeblock Decisions for Error Correction Mode Decoder .....	6-9
6-3 Decoding Strategy Modified for Filler Bit Augmentation .....	6-10
8-1 Probability of Not Recognizing the Start Sequence .....	8-4
8-2 Probability $P_{CX}$ of Codeblock Rejection in TED Mode .....	8-6
8-3 Meaning of Decoding Values .....	8-7
8-4 Decoding Cases in SEC Mode.....	8-7
8-5 Probability $P_{CY}$ of Codeblock Rejection in SEC Mode.....	8-8
8-6 Parity and Syndrome When Tail Sequence Has Errors .....	8-10
8-7 Probability of Missing the Tail Sequence.....	8-11
8-8 Frame Rejection Probabilities, $P_{FX}$ and $P_{FY}$ , for the Last or Only Frame in an Independent CLTU (PLOP-1) .....	8-12
8-9 Frame Rejection Probabilities, $P_{F2X}$ and $P_{F2Y}$ , for the Last or Only Frame in a CLTU in a Sequence of CLTUs (PLOP-2).....	8-16
8-10 Sources of Undetected Errors .....	8-18
8-11 Probability of $n$ Errors Occurring in a Codeblock.....	8-19
8-12 Error Detection Performance for a Codeblock in SEC and TED Modes .....	8-19
8-13 Probability of Undetected Error in a Frame, TED Mode, No CRC .....	8-20



**CONTENTS (continued)**

<u>Table</u>	<u>Page</u>
8-14 Probability of Undetected Error in a Frame, SEC Mode, No CRC .....	8-21
8-15 Probability of Undetected Error in a Frame, TED Mode, with CRC .....	8-22
8-16 Probability of Undetected Error in a Frame, SEC Mode, with CRC.....	8-23
C-1 Run Distribution for Sequences of Period $2^m-1$ .....	C-2
C-2 Results for the 16 Randomizer Polynomials .....	C-4
E-1 Probability $P_{CX}$ of Codeblock Rejection in TED Mode, Frame Length 20 Octets.....	E-1
E-2 Probability $P_{CX}$ of Codeblock Rejection in TED Mode, Frame Length 50 Octets.....	E-2
E-3 Probability $P_{CX}$ of Codeblock Rejection in TED Mode, Frame Length 120 Octets.....	E-2
E-4 Probability $P_{CX}$ of Codeblock Rejection in TED Mode, Frame Length 256 Octets.....	E-2
E-5 Probability $P_{CY}$ of Codeblock Rejection in SEC Mode, Frame Length 20 Octets .....	E-3
E-6 Probability $P_{CY}$ of Codeblock Rejection in SEC Mode, Frame Length 50 Octets .....	E-3
E-7 Probability $P_{CY}$ of Codeblock Rejection in SEC Mode, Frame Length 120 Octets .....	E-4
E-8 Probability $P_{CY}$ of Codeblock Rejection in SEC Mode, Frame Length 256 Octets .....	E-4
E-9 Probabilities $P_{TX}$ and $P_{TY}$ of Missing a Tail Sequence with the Earlier Pattern.....	E-5
E-10 Probabilities $P_{TXD}$ and $P_{TYD}$ of Missing Both Tail Sequences (Earlier Pattern) When a Double Tail Sequence Is Used .....	E-6
E-11 Probability $P_{TYF}$ of Missing a Tail Sequence (Earlier Pattern) When Filler Bit Augmentation Is Used in SEC Mode.....	E-6
E-12 Probabilities of Missing a Tail Sequence (Current Pattern) for Different Modes and Lengths.....	E-7

## **1 INTRODUCTION**

### **1.1 PURPOSE**

This Informational Report contains the concept and supporting rationale for the TC Synchronization and Channel Coding Recommended Standard developed by the Consultative Committee for Space Data Systems (CCSDS). It has been prepared to serve two major purposes:

- to provide an introduction and overview for the concept upon which the detailed CCSDS TC Synchronization and Channel Coding Recommended Standard (reference [2]) are based;
- to describe and explain the procedures employed and to supply the supporting rationale, including performance data and historical background information on their selection.

This document is a CCSDS Informational Report and is therefore not to be taken as a CCSDS Recommended Standard. The actual Recommended Standard is in reference [2].

### **1.2 SCOPE**

The procedures and data formats developed for the TC Synchronization and Channel Coding are designed for space communications links. Typically, the links are used to carry telecommand data from ground elements to spacecraft, and the ‘TC’ in the name originates from this use. However, the TC Synchronization and Channel Coding can also be applied to links between spacecraft and to links carrying non-telecommand data.

The procedures and data formats of the TC Synchronization and Channel Coding are designed with reliability and efficiency as primary considerations. The results reflect the consensus of experts from many space agencies.

This document provides supporting and descriptive material only: it is not part of the Recommended Standard. In the event of any conflict between the TC Synchronization and Channel Coding Recommended Standard (reference [2]) and the material presented in this report, the Recommended Standard (reference [2]) shall prevail.

### **1.3 ORGANIZATION**

An overview of TC Synchronization and Channel Coding is presented in section 2. Sections 3 through 7 describe the channel coding elements, and section 8 presents performance data. Annex A is a glossary, and annex B is a list of acronyms and abbreviations. Annexes C and D provide theoretical background. Annex E gives performance data for obsolete features. Annex F shows examples of data structures before and after the channel coding process.

## 1.4 REFERENCES

The following documents are referenced in this Report. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Report are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Reports and Recommended Standards.

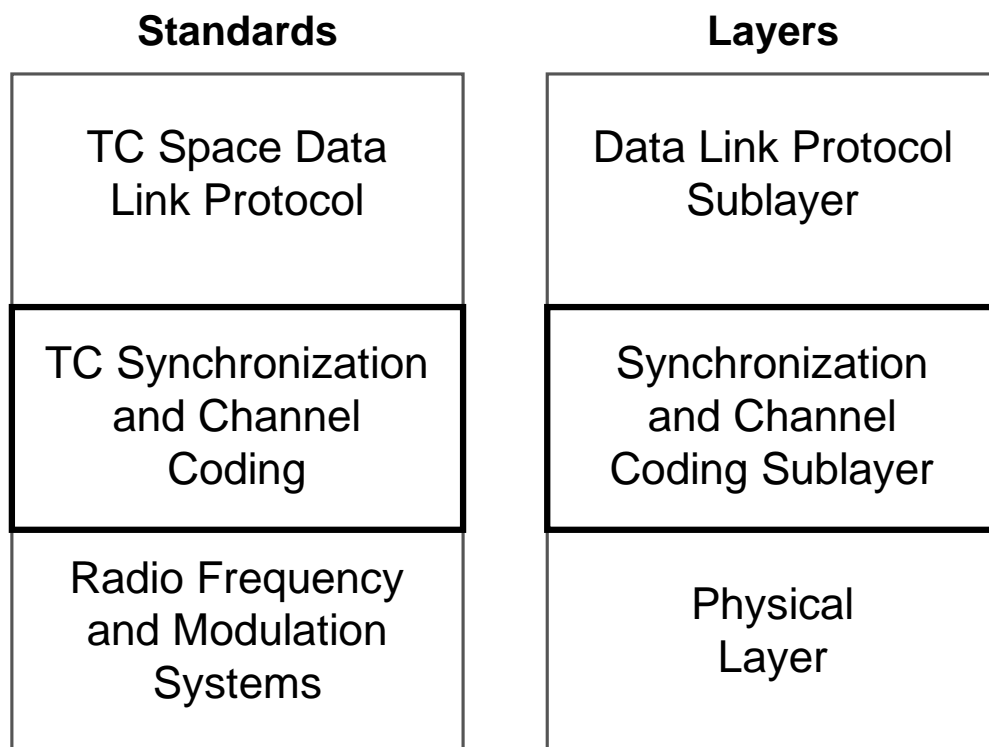
- [1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-9. Yellow Book. Issue 9. Washington, D.C.: CCSDS, November 2003.
- [2] *TC Synchronization and Channel Coding*. Recommendation for Space Data System Standards, CCSDS 231.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [3] *Overview of Space Link Protocols*. Report Concerning Space Data System Standards, CCSDS 130.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, June 2001.
- [4] *TC Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 232.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [5] *TM Synchronization and Channel Coding*. Recommendation for Space Data System Standards, CCSDS 131.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [6] *TM Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 132.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [7] *AOS Space Data Link Protocol*. Recommendation for Space Data System Standards, CCSDS 732.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.
- [8] *Radio Frequency and Modulation Systems—Part 1: Earth Stations and Spacecraft*. Recommendation for Space Data System Standards, CCSDS 401.0-B-15. Blue Book. Issue 15. Washington, D.C.: CCSDS, September 2005.
- [9] *Space Link Extension—Forward CLTU Service Specification*. Recommendation for Space Data System Standards, CCSDS 912.1-B-2. Blue Book. Issue 2. Washington, D.C.: CCSDS, December 2004.
- [10] R. Klootsema, et al. *Design and Development of a Very High Speed Reed-Solomon Encoder/Decoder Chip Set: Architectural Design Report*. ESA/ESOC Project no. 13.501. CSEM Technical Report 496.I. Neuchâtel, Switzerland: CSEM, 1992. [Not publicly available.]
- [11] J.C. Morakis. *Analysis of a Proposed TCM Coding System*. [Not publicly available.]
- [12] Robert J. McEliece. *Finite Field for Scientists and Engineers*. Kluwer International Series in Engineering and Computer Science. Norwell, MA, USA: Kluwer, 1987.

## 2 OVERVIEW

### 2.1 CONTEXT

TC Synchronization and Channel Coding forms part of the CCSDS Space Link Protocols. The report in reference [3] provides an overview of the Space Link Protocols and shows the relationship of the TC Synchronization and Channel Coding to the other elements. It also shows how the Space Link Protocols relate to the equivalent layers of the OSI model.

The TC Synchronization and Channel Coding Recommended Standard (reference [2]) defines the behavior of the Synchronization and Channel Coding Sublayer. The interfaces between the sublayer and its immediate neighbors are shown in figure 2-1.



**Figure 2-1: Related Standards and Layers**

The neighbor above is the Data Link Protocol Sublayer, which in this case is defined by the TC Space Data Link Protocol (reference [4]). The Data Link Protocol Sublayer generates variable-length TC Transfer Frames and passes them to the Synchronization and Channel Coding Sublayer.

NOTE – TC Synchronization and Channel Coding is used with the variable-length frames of the TC Space Data Link Protocol. TM Synchronization and Channel Coding (reference [5]) is used with the fixed-length frames of the TM Space Data Link Protocol (reference [6]) and the AOS Space Data Link Protocol (reference [7]).

The neighbor below is the Physical Layer. The TC Synchronization and Channel Coding Recommended Standard (reference [2]) includes procedures, called Physical Layer Operations Procedures (PLOPs), for controlling the behavior of the Physical Layer. The Radio Frequency and Modulation Systems (reference [8]) provides recommendations for RF modulation and the properties of the generated RF signal.

## 2.2 PURPOSE

The purpose of the Synchronization and Channel Coding Sublayer and the associated Physical Layer Operations Procedures at the sending end is to:

- encode the data units received from the sublayer above to provide a forward error detection and correction capability, which gives a high degree of protection against errors or corruptions that occur during transmission through the space link;
- encapsulate the data units so that the start and end can be detected by the receiving end;
- enable the receiving end to resolve the data ambiguity (sense of ‘1’ and ‘0’) in the received symbol stream;
- control the transmission of an acquisition bit pattern, which enables the receiver to acquire bit synchronization;
- ensure there are sufficient bit transitions in the transmitted bit stream so that the receiver can maintain bit synchronization during the reception of a data unit;
- control the optional transmission of an idle bit pattern, which enables the receiver to maintain bit synchronization between data units.

The purpose of the Synchronization and Channel Coding Sublayer at the receiving end is to:

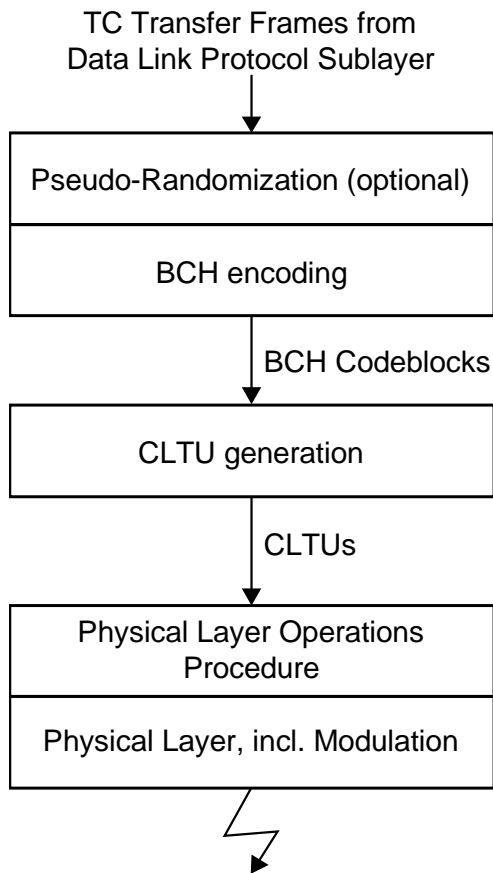
- detect the start and end of each data unit in the received symbol stream;
- resolve, if necessary, the data ambiguity (sense of ‘1’ and ‘0’) in the received symbol stream;
- decode each data unit and optionally correct the errors detected.

One instance of the Synchronization and Channel Coding Sublayer processes the data stream for a single Physical Channel. A Physical Channel is a stream of bits transferred over a space link in a single direction.

NOTE – Layers above the Synchronization and Channel Coding Sublayer may be using timers that include the time for a data unit to pass through the Synchronization and Channel Coding Sublayer. For example, the Sequence-Controlled Service Type of the TC Space Data Link Protocol has such timers. It can therefore be helpful if queuing and buffering mechanisms in the Synchronization and Channel Coding Sublayer are designed to limit unpredictable delays.

### 2.3 PROCEDURES AT THE SENDING END

TC Synchronization and Channel Coding defines procedures for the sending end, which are applied in the order shown in figure 2-2.



**Figure 2-2: Procedures at the Sending End**

**Pseudo-randomization** improves bit transition density, in order to help maintain bit synchronization of the RF equipment at the receiving end. Randomization is used unless it can be shown that a sufficient bit transition density will be generated by other means. Randomization is discussed in section 7.

NOTE – For brevity, the word ‘random’ is used in place of ‘pseudo-random’ in this report.

**BCH coding** provides protection against errors which occur during RF transmission. At the sending end, the data is encoded with a BCH (Bose-Chaudhuri-Hocquenghem) block code; the encoding procedure is discussed in section 3. The encoding generates a set of BCH Codeblocks.

NOTE – In recommendations published in September 2003, the name ‘BCH Codeblock’ replaced the earlier name ‘TC Codeblock’.

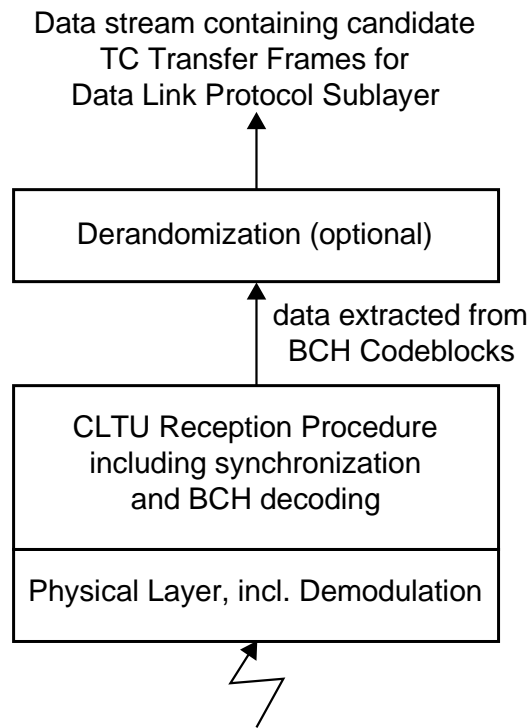
**A CLTU is generated** from each set of BCH Codeblocks. The Communications Link Transmission Unit (CLTU) consists of a Start Sequence, a set of BCH Codeblocks, and a Tail Sequence. The Start Sequence provides frame synchronization for the TC Transfer Frames. CLTU generation is discussed in section 4.

NOTE – In recommendations published in September 2003, the name ‘Communications Link Transmission Unit’ replaced the earlier name ‘Command Link Transmission Unit’.

**The Physical Layer Operations Procedures (PLOPs)** define how CLTUs and additional data structures, the Acquisition Sequence and the Idle Sequence, are used with different states of channel modulation. The Recommended Standard (reference [2]) includes two procedures, called PLOP-1 and PLOP-2, which are discussed in section 5.

## 2.4 PROCEDURES AT THE RECEIVING END

TC Synchronization and Channel Coding defines procedures for the receiving end, which are applied in the order shown in figure 2-3.



**Figure 2-3: Procedures at the Receiving End**

**The CLTU Reception Procedure** processes a stream of bits. It searches the input stream, looking for the Start Sequence, which identifies the start of a CLTU. It then decodes the following BCH Codeblocks, in the decoding mode set for the Physical Channel, and delivers data extracted from the successfully decoded codeblocks. Two decoding modes are available:

an error detecting mode and an error correcting mode. The procedure continues processing the codeblocks until it detects a decoding failure. The CLTU Reception Procedure is discussed in section 6.

**If Pseudo-randomization** is in operation on the Physical Channel, the data extracted from the decoded BCH codeblocks is derandomized before it is passed to the Data Link Protocol Sublayer. Randomization is discussed in section 7.



### 3 BCH ENCODING

#### 3.1 INTRODUCTION

At the sending end, the Data Link Protocol Sublayer generates variable-length TC Transfer Frames and passes them to the Synchronization and Channel Coding Sublayer. The format of a TC Transfer Frame is defined in reference [4]. The Synchronization and Channel Coding Sublayer encodes the Frames with a BCH (Bose-Chaudhuri-Hocquenghem) block code and generates a set of BCH Codeblocks.

The BCH encoding is the subject of this section. It provides a forward error detection and correction capability which gives a high degree of protection against errors or corruptions that occur during transmission through the space link.

If randomization is in use for the Physical Channel, the Frames are randomized before the BCH encoding. Section 7 discusses randomization at the sending and receiving ends.

#### 3.2 FORMAL INTERFACE TO DATA LINK PROTOCOL SUBLAYER

The Recommended Standard (reference [2]) includes a formal definition of the service interface between the Data Link Protocol Sublayer and the Synchronization and Channel Coding Sublayer at the sending end. The Data Link Protocol Sublayer uses the following service primitive:

ChannelAccess.request (Frames)

The Frames parameter consists of one or more TC Transfer Frames to be transferred to the receiving end through the Physical Channel.

From the point of view of the Synchronization and Channel Coding Sublayer, the content of the Frames parameter is a single block of data. The Synchronization and Channel Coding Sublayer does not use the length fields or other values in the TC Transfer Frames and it does not need to know the positions of any Frame boundaries within the block of data.

For a single ChannelAccess.request, the Synchronization and Channel Coding Sublayer generates a set of BCH Codeblocks, and that set of BCH Codeblocks is placed in a single CLTU. One of the managed parameters for the Physical Channel is the maximum length of a CLTU; this parameter places a constraint on the maximum length of the Frames parameter for the service request. The length of the CLTU can be calculated from the total length of the Frames in the Frames parameter as follows:

$$\text{Length of the CLTU} = 10 + ((\text{Total length of the Frames} + 6) / 7) * 8$$

where the lengths are expressed in octets, and the division represented by the '/' character is integer division, which truncates any fractional part of the result. The result of the division corresponds to the number of BCH Codeblocks. So, for example, if the total length of the Frames is 1189 octets, then:

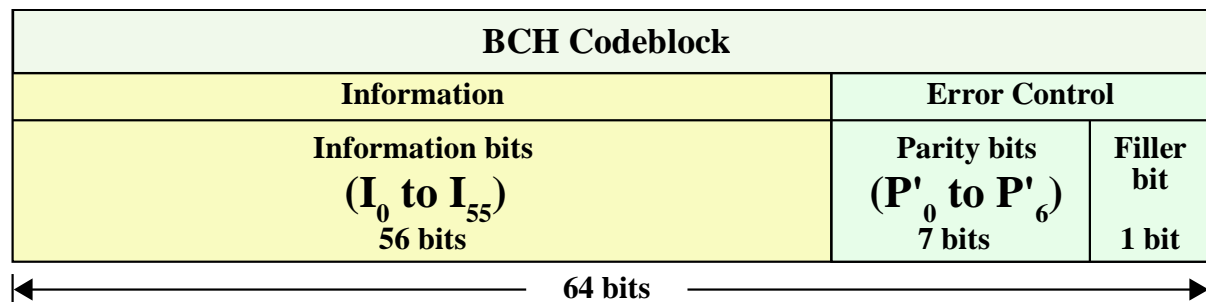
- $1189 + 6 = 1195$ ;
- integer division of 1195 by 7 gives 170 (the result of the division is 170.71 and the fractional 0.71 is truncated), so there will be 170 BCH Codeblocks;
- $170 * 8 = 1360$ ; and
- addition of 10 gives a CLTU length of 1370 octets.

Also, some of the standards derived from the Recommended Standard (reference [2]) place a limit of one TC Transfer Frame per CLTU. A CLTU carrying a single maximum-length TC Transfer Frame (1024 octets) has 147 BCH Codeblocks and a length of 1186 octets.

### 3.3 FORMAT OF A BCH CODEBLOCK

#### 3.3.1 THE 64-BIT BCH CODEBLOCK

The format of a BCH Codeblock is shown in figure 3-1.



**Figure 3-1: Format of a BCH Codeblock**

**Information bits:** Subsection 3.4 below discusses how the variable-length input from the Frames parameter is handled to provide 56 Information bits for each BCH Codeblock.

**Parity bits:** The Information bits are used as input to the encoding process described in 3.6 below. The encoding process delivers seven parity bits,  $P_0$  to  $P_6$ . It is the complements of these bits,  $P'_0$  to  $P'_6$ , which are placed in the Error Control field of the BCH Codeblock.

**Filler bit:** The Filler Bit is always zero.

When a BCH Codeblock is transmitted as part of a CLTU, the first bit of the codeblock to be transmitted is  $I_0$ .

### 3.3.2 OBSOLETE CODEBLOCK LENGTHS

The Recommended Standard (reference [2]) includes only the BCH Codeblock with a length of 64 bits. Earlier CCSDS recommendations for Telecommand Channel Coding included further codeblock length options of 40 bits, 48 bits and 56 bits. For the other codeblock lengths, the Error Control field remained at eight bits and the Information field was shortened, so the codeblock carried less data. The selected codeblock length option was fixed for the mission.

The further length options were eliminated in favor of a single standard length of 64 bits, in recommendations published in June 2000.

The 64-bit codeblock is the most efficient of the options. Table 3-1 shows the bandwidth overhead of the BCH coding for the 64-bit codeblock and for the obsolete shorter codeblock lengths.

**Table 3-1: Bandwidth Overhead for Different Codeblock Lengths**

<b>Codeblock length</b>	<b>Data length</b>	<b>Bandwidth overhead</b>
64 bits	56 bits	14%
56 bits (obsolete)	48 bits	17%
48 bits (obsolete)	40 bits	20%
40 bits (obsolete)	32 bits	25%

### 3.4 BREAKING THE FRAMES INTO FIXED-LENGTH PIECES

The data from the TC Transfer Frames are placed, seven octets (56 bits) at a time, into a set of BCH Codeblocks.

Note that if randomization is in use for the Physical Channel, then the Frame data are randomized before being placed in the BCH Codeblocks. Section 7 discusses randomization.

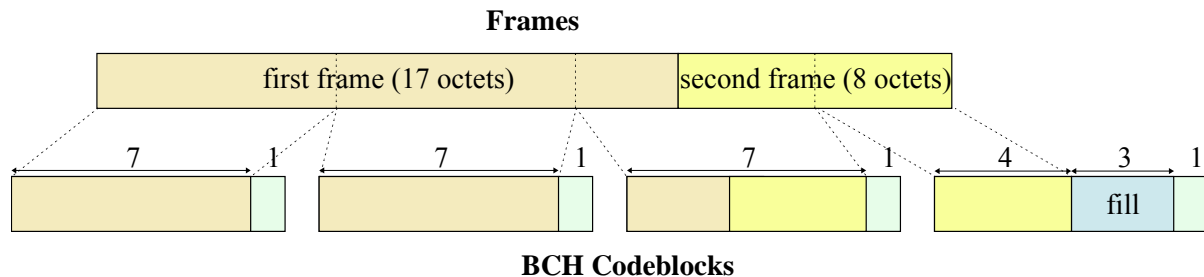
The first seven octets from the start of ‘Frames’ are placed in the Information field of the first codeblock and the next seven octets in the next codeblock and so on. Frame boundaries are ignored during this process. As mentioned in 3.2 above, the Synchronization and Channel Coding Sublayer does not need to know the positions of any Frame boundaries within the ‘Frames’ data.

At the end of ‘Frames’, there may not be seven octets left for the last BCH Codeblock. In this case, the available octets are placed in the first octets of the Information field and the rest of the Information field is filled with a fill pattern:

- the fill pattern consists of alternating ‘0’ and ‘1’ bits, starting with a ‘0’, so an octet of fill bits has the hexadecimal value 55;

- only the last BCH Codeblock of a CLTU can have fill bits;
- the last BCH Codeblock can have up to six octets of fill bits;
- if randomization is in use for the Physical Channel, then randomization of the fill bits is optional.

Figure 3-2 shows a simple example, where a Frames parameter with a total length of 25 octets results in four BCH Codeblocks. In the example, the last BCH Codeblock has three octets of fill bits.



**Figure 3-2: Example of Breaking ‘Frames’ into BCH Codeblocks**

### 3.5 THE BCH CODE

#### 3.5.1 GENERAL

The BCH code specified in the Recommended Standard (reference [2]) is a systematic (63,56) block code. A 56-bit information word is encoded into a 63-bit codeword. Because the code is systematic, the information word is unchanged by the encoding, which generates seven check bits.

The code is an expurgated Hamming code, derived from a basic (63,57) Hamming code.

#### 3.5.2 BASIC (63,57) HAMMING CODE

The basic Hamming codes have a codeword length of  $2^n - 1$  bits, with  $n$  check bits. The codes have minimum distance of three, that is, each codeword differs from every other codeword in at least three bit positions. A basic Hamming code can either correct a single error or detect two errors, but not both simultaneously.

When  $n=6$ , the basic Hamming code has six check bits, a codeword length of 63 bits and an information word of 57 bits. Therefore the code has  $2^{57}$  codewords corresponding to the  $2^{57}$  possible input information words.

The generator polynomial,  $g_1(x)$ , for the basic (63,57) Hamming code is:

$$g_1(x) = x^6 + x + 1$$

The same polynomial can be applied to the received 63-bit codeblock to give the six-bit syndrome value. Considering the received 63-bit codeblock as a polynomial whose coefficients are the bit values, the syndrome is the remainder when it is divided (with polynomial division and arithmetic in the binary field) by the generator polynomial.

If the syndrome is zero, then the received codeblock is one of the codewords, and is assumed to be error-free. If a single error has occurred, the syndrome value indicates the position of the error.

### 3.5.3 EXPURGATED HAMMING CODE (63,56)

The basic (63,57) Hamming code can be improved by limiting it to the even-parity codewords. The resulting code has only  $2^{56}$  codewords so it has an information word of 56 bits and 7 check bits. The code now includes a parity check, and the generator polynomial,  $g(x)$ , for this expurgated Hamming code (63,56) is:

$$g(x) = g_1(x)(x + 1)$$

$$g(x) = (x^6 + x + 1)(x + 1)$$

$$g(x) = x^7 + x^6 + x^2 + 1$$

The code has a minimum distance of four. If it is used in error detecting mode, it can detect up to three errors and it can also detect any odd number of errors. If it is used in error-correcting mode, it can detect up to two errors and correct a single error.

## 3.6 BCH ENCODING

The Recommended Standard (reference [2]) includes a schematic diagram for the implementation of the BCH encoding using a linear feedback shift register.

## 4 CLTU GENERATION

### 4.1 INTRODUCTION

At the sending end, the set of BCH Codeblocks resulting from the BCH encoding are placed in a CLTU.

The CLTU includes fixed data patterns at the start and end. The Start Sequence provides the synchronization pattern and delimits the beginning of the first BCH Codeblock. The Tail Sequence marks the end of the CLTU.

### 4.2 LENGTH OF A CLTU

For a single ChannelAccess.request from the Data Link Protocol Sublayer, the Synchronization and Channel Coding Sublayer generates a set of BCH Codeblocks, and that set of BCH Codeblocks is placed in a single CLTU.

If  $N$  is the number of BCH Codeblocks in the CLTU, then:

$$\text{Length of the CLTU} = 2 + (N + 1) * 8$$

where the lengths are expressed in octets. The minimum length of a CLTU is therefore the case when  $N=1$ , giving a CLTU of 18 octets.

The Recommended Standard (reference [2]) does not specify an upper limit for the length of a CLTU. However, it specifies a managed parameter for the maximum length of a CLTU for a Physical Channel. Also, the Space Link Extension Forward CLTU Service (reference [9]) has a management parameter which sets the maximum length of a CLTU for a service instance.

Section 8 contains a detailed analysis of the performance of the link based on the CLTU length.

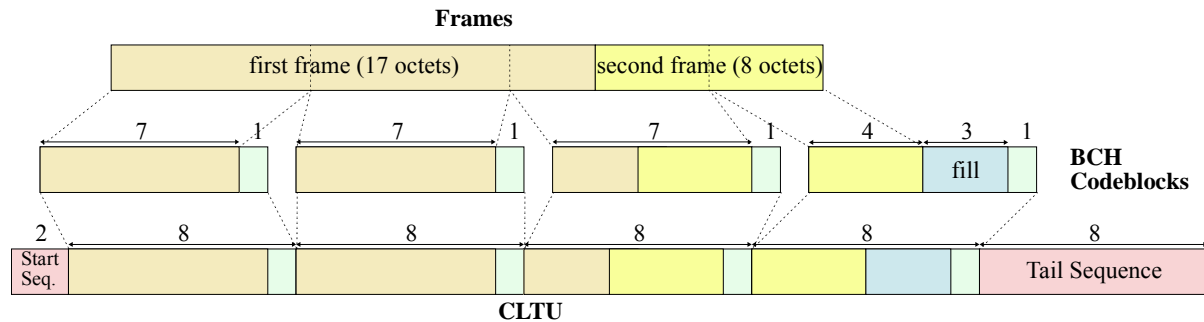
### 4.3 FORMAT OF A CLTU

Figure 4-1 shows the format of a CLTU.

<b>Communications Link Transmission Unit (CLTU)</b>		
<b>START SEQUENCE</b>	<b><math>N</math> BCH Codeblocks</b>	<b>TAIL SEQUENCE</b>
<b>16 bits</b>	<b><math>N * 64</math> bits</b>	<b>64 bits</b>

**Figure 4-1: Format of a CLTU**

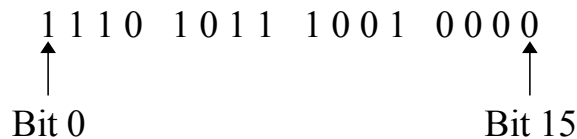
Figure 4-2 extends the example from figure 3-2 by showing the resulting CLTU.



**Figure 4-2: Example of BCH Codeblocks in a CLTU**

#### 4.4 START SEQUENCE

The Start Sequence is a 16-bit (two-octet) field containing the synchronization pattern shown in figure 4-3. The pattern has the hexadecimal value EB90.



**Figure 4-3: Start Sequence**

When the CLTU is transmitted, bit 0 of the Start Sequence is the first bit to be transmitted. The Start Sequence is immediately followed by the first BCH Codeblock of the CLTU.

The Start Sequence provides a synchronization pattern with low autocorrelation side lobes, thereby reducing the probability that the receiving end will mistakenly identify the pattern. It is clearly distinguishable from the Idle Sequence or Acquisition Sequence. The Idle Sequence and the Acquisition Sequence are discussed in section 5.

## 4.5 TAIL SEQUENCE

### 4.5.1 TAIL SEQUENCE FORMAT

The Tail Sequence is a 64-bit (eight-octet) field containing the following bit pattern:

```
11000101 11000101 11000101 11000101 11000101 11000101 11000101 01111001
```

The first seven octets of the Tail Sequence each contain the bit pattern 11000101 (hexadecimal C5) and the last octet has the bit pattern 01111001 (hexadecimal 79). So the Tail Sequence is hexadecimal C5 C5 C5 C5 C5 C5 C5 79.

The Tail Sequence immediately follows the last BCH Codeblock of the CLTU.

The Tail Sequence field has the same length as a BCH Codeblock, but its contents differ from a valid or correctable BCH Codeblock. As discussed in section 6, when the decoding process at the receiving end encounters an invalid or noncorrectable BCH Codeblock, it stops processing the CLTU and starts looking for the Start Sequence of the next CLTU.

In recommendations published in November 1995, the current Tail Sequence format replaced the earlier pattern described in 4.5.3 below. Annex D contains the theoretical background behind the choice of the current Tail Sequence pattern.

### 4.5.2 EARLIER TAIL SEQUENCE LENGTHS

The Tail Sequence field has the same length as a BCH Codeblock. This was also the case when one of the obsolete codeblock length options was used.

As discussed in 3.3.2, earlier CCSDS recommendations for Telecommand Channel Coding included further codeblock length options of 40 bits, 48 bits, and 56 bits. For a Tail Sequence with one of these other lengths, the last octet contained hexadecimal 79 and the other octets contained hexadecimal C5.

The shorter Tail Sequence lengths were eliminated with the associated BCH Codeblock length options, in recommendations published in June 2000.

### 4.5.3 EARLIER TAIL SEQUENCE PATTERN

Earlier CCSDS recommendations for Telecommand Channel Coding defined a different bit pattern for the Tail Sequence. The pattern consisted of alternating '0' and '1' bits, starting with a '0' bit, so that each octet in the Tail Sequence contained hexadecimal 55.

In recommendations published in November 1995, the old Tail Sequence pattern was replaced by the current pattern described in 4.5.1 above. The current Tail Sequence provides a more distinctive pattern, which is more reliably detected as the end of a CLTU (see performance discussion in section 8 and annex E).



## 5 PHYSICAL LAYER OPERATIONS PROCEDURES

### 5.1 GENERAL

The Physical Layer interfaces directly with the transmission medium, typically a radio path through space. As such it activates and deactivates the physical connection between the transmitter and the receiving spacecraft.

As the Physical Layer is mainly concerned with providing the physical connection between the transmitter and the receiving spacecraft, the major specification of this layer is found in Reference [8]. However, a small part is specified by the TC Synchronization and Channel Coding Recommended Standard (reference [2]).

The Physical Layer modulates the CLTUs received from the Synchronization and Channel Coding Sublayer onto the RF carrier. Reference [2] defines two Physical Layer Operations Procedures (PLOPs):

**PLOP-1.** The Physical Channel is deactivated between each CLTU, so (at least) bit synchronization is lost. In this case the Physical Channel is reactivated with each CLTU.

**PLOP-2.** The Physical Channel is not deactivated after each CLTU, and an idle sequence ensures that the bit synchronization is maintained at the receiving end.

### 5.2 ELEMENTS

#### 5.2.1 COMMUNICATIONS SESSION

The PLOPs specify the sequence of operations performed during a communications session. A communications session is defined as a continuous period of time during which the signal path is established for the communications channel.

A session begins when the Physical Layer establishes the physical connection from the transmitter to the receiver, by the provision of the RF carrier. The session ends with the removal of the carrier.

#### 5.2.2 ACQUISITION SEQUENCE

The purpose of the Acquisition Sequence is to enable the receiving end to acquire bit synchronization. It is a bit pattern with a high transition density, consisting of alternating '1' and '0' bits, starting with either a '1' or a '0'.

The length of the Acquisition Sequence is selected for a mission, according to the communications link performance properties of the mission. The preferred minimum length is 16 octets (128 bits). The length does not have to be an integral number of octets.

### 5.2.3 IDLE SEQUENCE

The purpose of the Idle Sequence is to enable the receiving end to maintain bit synchronization when no CLTUs are being transmitted. It is a bit pattern with a high transition density, consisting of alternating ‘1’ and ‘0’ bits, starting with either a ‘1’ or a ‘0’.

An Idle Sequence can have any length and does not have to be an integral number of octets.

NOTE – Earlier CCSDS recommendations for Telecommand Channel Coding included the requirement that the bit pattern of an Idle Sequence should start with a ‘0’. This requirement was related to the earlier Tail Sequence pattern (see 4.5.3), which also consists of alternating ‘1’ and ‘0’, starting with a ‘0’. With the improved performance of the current Tail Sequence pattern (see 4.5.1), the requirement was removed.

### 5.2.4 CARRIER MODULATION MODES

#### 5.2.4.1 General

The Carrier Modulation Modes (CMMs) consist of different states of data modulation upon the RF carrier which creates the physical telecommand channel. Table 5-1 shows the four basic CMMs which are combined to perform the PLOPs.

**Table 5-1: Carrier Modulation Modes**

<b>Mode</b>	<b>Description</b>
CMM-1	Unmodulated carrier only
CMM-2	Carrier Modulated with Acquisition Sequence
CMM-3	Carrier Modulated with CLTUs
CMM-4	Carrier Modulated with Idle Sequence

#### 5.2.4.2 CMM-1

CMM-1 is concerned with the establishment and maintenance of the radio-frequency part of the physical connection. The unmodulated carrier mode is defined as the state in which no telecommand modulation (i.e., data modulation) is present.

In establishing and maintaining the connection, the sending end uses information about the status of the receiving spacecraft transponders. This information is carried by the No RF Available Flag in the Communications Link Control Word (CLCW) as specified in reference [4].

### **5.2.4.3 CMM-2**

CMM-2 is concerned with the establishment of the modulation part of the physical connection. The sending end transmits an Acquisition Sequence with the length selected for the mission, to enable the receiving end to acquire bit synchronization.

The sending end may use information about the quality of the received bit stream, carried by the No Bit Lock Flag in the CLCW as specified in reference [4]. The use of the No Bit Lock Flag is optional and mission-specific. If used, the flag provides the same service throughout CMM-3 and CMM-4.

### **5.2.4.4 CMM-3**

CMM-3 is concerned with the transmission of one CLTU on the physical connection.

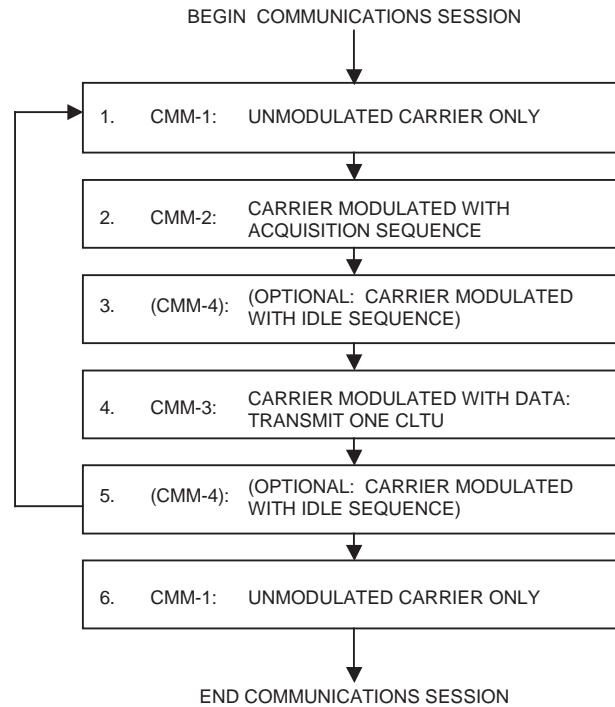
### **5.2.4.5 CMM-4**

CMM-4 is concerned with the maintenance of the modulation part of the physical connection. The Idle Sequence is transmitted to enable the receiving end to maintain bit synchronization.

The maximum length of the Idle Sequence is unconstrained and is dictated by the timing of the telecommand operations (e.g., when no CLTU is available).

## **5.3 PLOP-1**

The sequence of CMMs for PLOP-1 is shown in figure 5-1, reproduced from Recommended Standard (reference [2]).

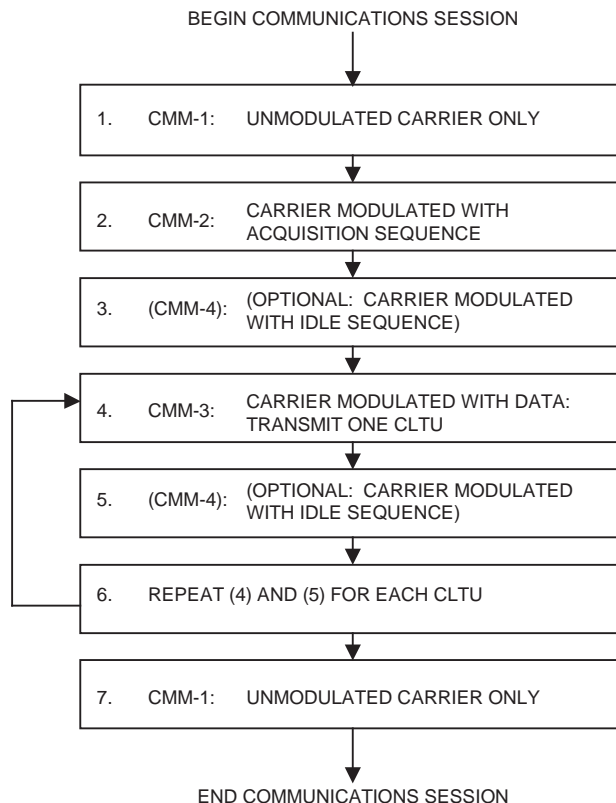


**Figure 5-1: PLOP-1 Carrier Modulation Modes**

In PLOP-1, the Physical Channel returns to CMM-1 (unmodulated carrier only) after each CLTU and, as a result, bit synchronization is lost at the receiving end. Therefore the Acquisition Sequence (CMM-2) is transmitted to enable the receiving end to regain bit synchronization, before transmitting the next CLTU (CMM-3). Optional Idle Sequences of any desired lengths can be transmitted (CMM-4) before and after a CLTU, to suit the timing conditions.

#### 5.4 PLOP-2

The sequence of CMMs for PLOP-2 is shown in figure 5-2, reproduced from Recommended Standard (reference [2]).



**Figure 5-2: PLOP-2 Carrier Modulation Modes**

In PLOP-2, CMM-1 is only used at the start and end of a communications session. The Acquisition Sequence (CMM-2) is transmitted only once, when it is used at the start of a session to enable the receiving end to acquire bit synchronization.

Successive CLTU transmissions (CMM-3) are separated only by an optional Idle Sequence (CMM-4) of any desired length. For example, if no CLTU is available for transmission, the Idle Sequence continues until a CLTU becomes available or the end of the session is reached. The Idle Sequence enables bit synchronization to be maintained at the receiving end.

The Idle Sequence between CLTUs is optional, and it can be omitted if the next CLTU is available. However, reference [2] recommends that an Idle Sequence of at least one octet should be transmitted between two CLTUs (see 6.7 below).

## 5.5 SELECTION OF PLOP-1 OR PLOP-2

With PLOP-1, the Acquisition Sequence is transmitted before every CLTU. The maximum throughput of the channel is therefore reduced compared to a similar channel operated with PLOP-2. The magnitude of the reduction depends on factors such as:

- the average CLTU length;

- the lengths of any optional Idle Sequences in PLOP-1 and in PLOP-2;
- the duration of the CMM-1 transmission between CLTUs in PLOP-1.

However, PLOP-1 can provide improved reliability, thereby reducing the need for retransmissions by the higher layers. With PLOP-2, there is a small risk that the receiving end fails to detect the boundary between successive CLTUs. With PLOP-1, this risk is avoided. The difference depends on the behavior of the CLTU reception procedures at the receiving end and is discussed in 6.7 below.

## **5.6 FORMAL INTERFACE TO SUBLAYER ABOVE**

The Recommended Standard (reference [2]) does not define the service provided by the PLOP and the Physical Layer to the Synchronization and Channel Coding Sublayer. Clearly, the PLOP accepts CLTUs from the Synchronization and Channel Coding Sublayer, but there is no formal definition of the service interface. The Recommended Standard makes no mention of any information passing from the PLOP and the Physical Layer to the Synchronization and Channel Coding Sublayer.

The Space Link Extension Forward CLTU Service (reference [9]) defines a service for transmitting CLTUs. The underlying function is the service provided by the PLOP as defined in reference [2]. The Forward CLTU Service includes operations for sending a CLTU and for reporting events in the performance of the service. Parameters include, for example, the minimum delay between transmitting one CLTU and the next. Implementers may find it helpful to consider the operations and parameters of the Forward CLTU Service when designing the interfaces presented by the PLOP and the Physical Layer.

## 6 CLTU RECEPTION PROCEDURES

### 6.1 INTRODUCTION

Sections 3, 4, and 5 discuss the procedures at the sending end of the space link. This section looks at the procedures for reception of a CLTU at the receiving end.

At the receiving end, the Physical Layer delivers a stream of channel bits to the Synchronization and Channel Coding Sublayer, together with information on the state of the physical communications channel. The Synchronization and Channel Coding Sublayer:

- scans the input stream of bits, looking for the Start Sequence of a CLTU;
- if necessary, uses the Start Sequence to resolve the data ambiguity (sense of ‘1’ and ‘0’) in the input stream of bits;
- decodes the following BCH Codeblocks until a decoding failure is detected;
- scans the input stream of bits, looking for the Start Sequence of the next CLTU, and so on.

For the data in the successfully decoded BCH Codeblocks in a CLTU, the Synchronization and Channel Coding Sublayer:

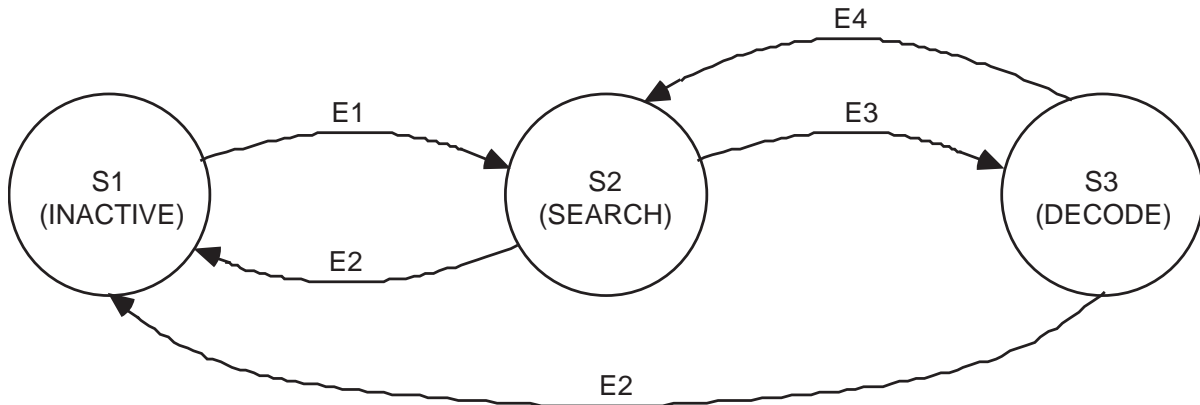
- derandomizes the data, if randomization is in use for the Physical Channel;
- delivers the data, which consists of candidate TC Transfer Frames, to the Data Link Protocol Sublayer.

The following subsections discuss the CLTU reception procedures in more detail. The derandomization process is described in section 7.

### 6.2 CLTU RECEPTION LOGIC

The definition of the CLTU reception logic in the Recommended Standard (reference [2]) uses the state diagram shown in figure 6-1. There are three states:

- S1, INACTIVE, when no bits are being received from the Physical Layer, typically because no bit modulation is detected or because bit lock is not achieved;
- S2, SEARCH, when the bits received from the Physical Layer are being searched for a CLTU Start Sequence;
- S3, DECODE, when BCH Codeblocks are being decoded.



**Figure 6-1: State Diagram for CLTU Reception Logic**

The CLTU reception logic can also be summarized by the state table in table 6-1.

**Table 6-1: State Table for CLTU Reception Logic**

	<b>INACTIVE State S1</b>	<b>SEARCH State S2</b>	<b>DECODE State S3</b>
<b>CHANNEL ACTIVATION Event E1</b>	Go to SEARCH	-	-
<b>CHANNEL DEACTIVATION Event E2</b>	-	Go to INACTIVE	End processing of current CLTU; Go to INACTIVE
<b>START SEQUENCE FOUND Event E3</b>	-	Go to DECODE	-
<b>CODEBLOCK REJECTION Event E4</b>	-	-	End processing of current CLTU; Go to SEARCH

### 6.3 INACTIVE STATE

The initial state for the CLTU reception procedure is the INACTIVE state (S1).

In this state, the Physical Layer is not sending any bits to the Synchronization and Channel Coding Sublayer. The Physical Layer does not have bit synchronization of an RF signal.

When the Physical Layer achieves bit synchronization, it signals CHANNEL ACTIVATION (Event E1) and starts sending the demodulated bits to the Synchronization and Channel Coding Sublayer. The CLTU reception procedure goes to SEARCH state (S2) and starts processing the received bits.





### 6.4.3 ERRORS IN THE START SEQUENCE

The Recommended Standard (reference [2]) specifies two options in the CLTU reception procedure for the handling of errors in the Start Sequence. Either no errors are accepted or a single error is accepted.

If no errors are accepted, then the received bit pattern is accepted as a valid Start Sequence only if it exactly matches the expected bit pattern. This option is used with the error-detecting mode for the BCH decoding.

If a single error is accepted, then the received bit pattern is accepted as a valid Start Sequence either if it exactly matches the expected bit pattern or if it differs by one bit from the expected bit pattern. This option is used with the error-correcting mode for the BCH decoding.

NOTE – If the option for accepting a single error applies, and the CLTU reception procedure is searching for the Start Sequence or its inverse, then a received bit pattern which differs by one bit from the Start Sequence inverse is also accepted.

## 6.5 DECODE STATE

### 6.5.1 GENERAL

In the DECODE state (S3) the CLTU reception procedure decodes the BCH Codeblocks in the received stream of bits. The DECODE state is entered from the SEARCH state when the Start Sequence is found. The first bit following the Start Sequence is the first bit of the first BCH Codeblock of the CLTU.

The CLTU reception procedure considers each 64-bit section of the incoming bit stream as a BCH Codeblock and decodes it. The BCH decoding has two modes of operation: error detection (see 6.5.2) and error correction (see 6.5.3).

NOTE – The algorithms for decoding described in 6.5.2 and 6.5.3 assume that the Physical Layer produces a hard decision output, so that the CLTU reception procedure receives a sequence of elements which are either ‘0’ or ‘1’. If the Physical Layer produces a soft decision output with more quantization levels, the CLTU reception procedure receives a sequence of elements with a larger range of possible values. In the soft decision case, the principle of searching for the codeword with the minimum distance from the received sequence still applies, but the discussion of algorithms is outside the scope of this report.

The procedure continues until it finds a codeblock that does not decode successfully. It rejects and discards the failed codeblock, and this is CODEBLOCK REJECTION (Event E4) which marks the end of a CLTU. The CLTU Tail Sequence is designed to trigger a CODEBLOCK REJECTION event. Following a CODEBLOCK REJECTION event, the CLTU reception procedure stops decoding and returns to SEARCH state.

The data contents of successfully decoded BCH Codeblocks are transferred to the Data Link Protocol Sublayer. If randomization is in use for the Physical Channel, the data are derandomized before transfer (see section 7). The CLTU reception procedure also delivers an indication of the start and end of the data extracted from the CLTU.

Occurrence of a CHANNEL DEACTIVATION (Event E2) also stops the decoding. The data contents of any successfully decoded BCH Codeblocks are transferred to the Data Link Protocol Sublayer as above. The CLTU reception procedure discards any partially received BCH Codeblock and returns to INACTIVE state.

## **6.5.2 ERROR DETECTION MODE**

### **6.5.2.1 Overview**

One of the modes of operation for BCH decoding is the error detection mode, which is also called Triple Error Detection (TED) because it is capable of detecting up to three errors in a BCH Codeblock.

In TED mode, no attempt is made to correct any errors. The decoding of a BCH Codeblock is considered successful only if no errors are found. If the decoding process detects any errors then the codeblock is rejected.

In TED mode, the value of the received Filler Bit at the end of a BCH Codeblock is ignored.

The Recommended Standard (reference [2]) states that, when TED mode is in use, no error is allowed in the Start Sequence (see 6.4.3).

Because the received codeblock can have more than three errors, there is a small risk that the TED mode decoding fails to detect errors. There is also a small risk that transmission errors can transform the Tail Sequence so that it appears to be an error-free codeblock. Section 8 includes information on the performance of the BCH code in TED mode, including the probabilities of failing to detect the Tail Sequence.

### **6.5.2.2 An Implementation for Error Detection Mode**

The Recommended Standard (reference [2]) includes a schematic diagram for the implementation of the BCH encoding using a linear feedback shift register. A similar method can be used by a BCH decoder in TED mode to calculate the seven parity bits.

A BCH decoder for TED mode could be implemented as follows:

- the first 56 bits from the received BCH Codeblock are used to calculate seven parity bits;
- the next seven bits from the received BCH Codeblock are inverted and compared with the calculated parity bits;

- if any difference is found between the received parity bits and the calculated parity bits, the BCH Codeblock is rejected.

The value of the Filler Bit, which is the last bit of the received BCH Codeblock, is not used by the decoder.

### **6.5.3 ERROR CORRECTION MODE**

#### **6.5.3.1 Overview**

One of the modes of operation for BCH decoding is the error correction mode, which has the full name Single Error Correction and Double Error Detection mode because it is capable of correcting a single error and detecting two errors in a BCH Codeblock. For convenience it is generally called Single Error Correction (SEC) mode.

In SEC mode, the decoding of a BCH Codeblock is considered successful if no errors are found or if a single error is found and corrected. If the decoding process detects uncorrectable errors, then the codeblock is rejected.

In SEC mode, the value of the received Filler Bit at the end of a BCH Codeblock is ignored, unless the obsolete Filler Bit Augmentation Algorithm described in 6.5.3.3 is in use.

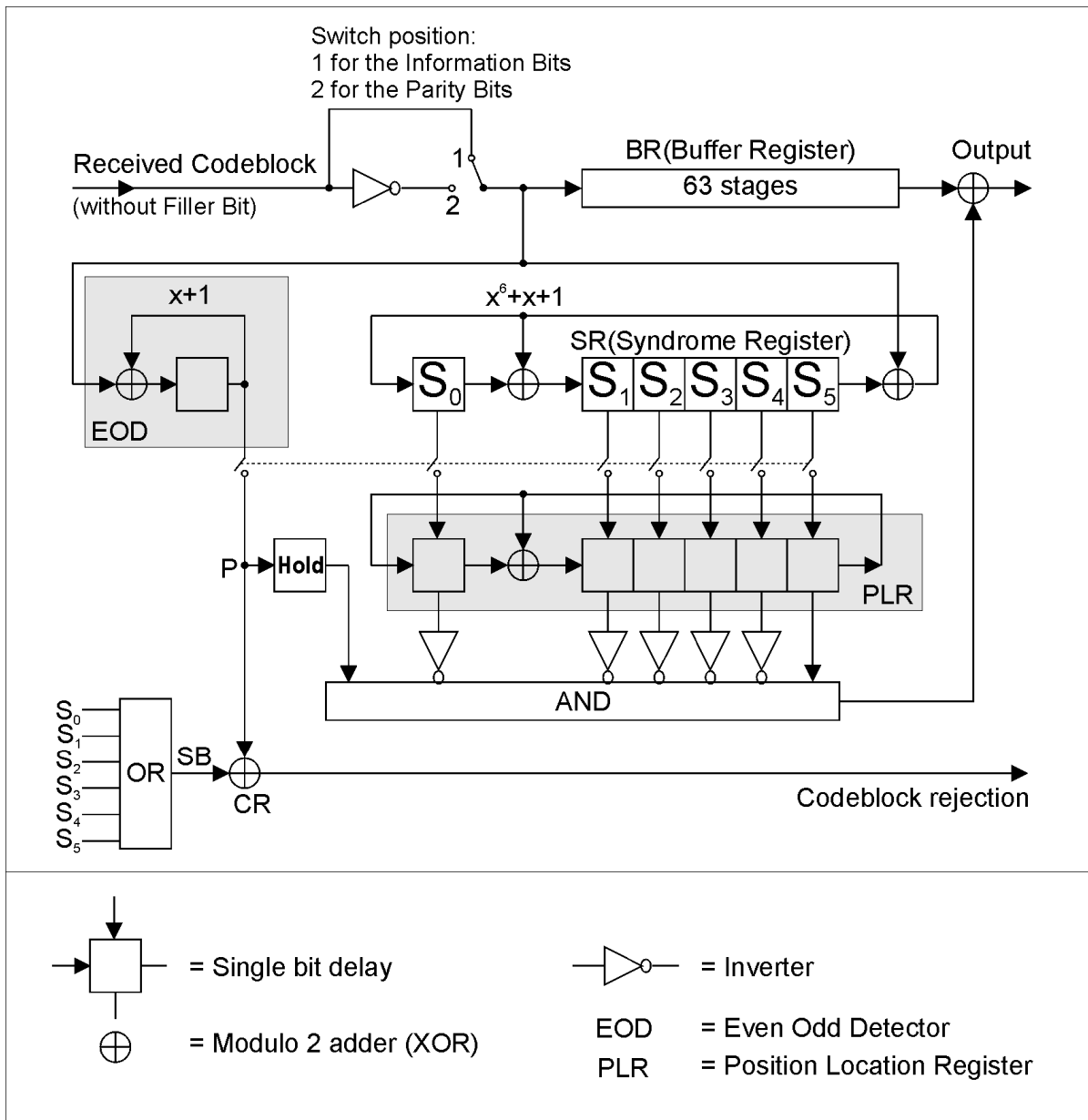
Because the received codeblock can have more than two errors, there is a small risk that the SEC mode decoding fails to detect errors or that it makes a faulty correction. There is also a risk that transmission errors can transform the Tail Sequence so that it appears to be an error-free or correctable codeblock. Section 8 includes information on the performance of the BCH code in SEC mode, including the probabilities of failing to detect the Tail Sequence.

#### **6.5.3.2 An Implementation for Error Correction Mode**

Figure 6-3 shows a schematic diagram for a decoder for SEC mode, using shift registers. It consists of the following devices:

- an inverter, which inverts the Parity Bits of the incoming BCH Codeblock;
- a 63-stage Buffer Register (BR);
- a six-stage upper Syndrome Register (SR), which calculates the syndrome;
- a six-stage lower Position Location Register (PLR);
- a seven-input AND gate for error correction at the appropriate location;
- a one-stage Even/Odd Detector (EOD) which calculates the parity;
- a Hold device, which applies the EOD output to the AND gate;
- a six-input OR gate, with inputs tied to the SR, which outputs SB (Syndrome Binary);

- an XOR gate (CR, Codeblock Rejection), which indicates an uncorrectable error, as described below;
- an error correction XOR gate at the output.



**Figure 6-3: An Implementation for the Error Correction Mode Decoder**

The decoder operates as follows.

The EOD (Even Odd Detector) and the SR (Syndrome Register) are all set to zeros before receiving a new BCH Codeblock. The first 63 bits of a BCH Codeblock are input to the decoder.

NOTE – The Filler Bit at the end of a BCH Codeblock is ignored.

The switch at the input is in position 1 during the first 56 bits of each BCH Codeblock and in position 2 (which uses the inverter) for the next seven bits. The resulting 63 bits are input to the BR, the SR, and the EOD.

After the 63rd bit is received:

- the BR is full;
- the SR contains the error syndrome ( $S = S_0, S_1, S_2, S_3, S_4, S_5$ );
- the EOD outputs a bit P, which is ‘1’ if the parity is odd and ‘0’ if it is even.

The set of seven switches below SR closes momentarily to:

- transfer the contents of the SR to the PLR;
- transfer the output of the EOD to the XOR gate CR and to the Hold device.

At the same time, the contents of the SR are input to the OR gate (the transfer path is not shown in the figure). The output of the OR gate is SB (Syndrome Binary), and:

- $SB = '0'$  if the error syndrome S ( $S = S_0, S_1, S_2, S_3, S_4, S_5$ ) is ‘all 0’;
- $SB = '1'$  if  $S > 0$ .

Then the SR and the EOD are all set to zeros to prepare for the next BCH Codeblock. The seven switches open before the 63 bits of the next BCH Codeblock are clocked in and the process repeats.

While the SR is solving for the syndromes of the next BCH Codeblock, the PLR is solving for the location of any single error in the current one. The Hold applies the output of the EOD to the AND gate for the next 63 shifts of the PLR. In the meantime, as the next BCH Codeblock is shifted into the buffer BR, the current one is shifted out and, if there is an error, it is corrected at the output XOR gate. This correction occurs when the contents of the PLR register are ‘000001’ and the Hold output is ‘1’.

The XOR gate CR determines whether the codeblock is rejected, as summarized in table 6-2.

**Table 6-2: Codeblock Decisions for Error Correction Mode Decoder**

EOD OUTPUT (P)	BINARY SYNDROME VALUE (SB)	DECISION
'0'	'0'	No Errors Detected The output of the AND gate is '0' so no correction is made. The output of the XOR gate CR is '0' so the codeblock is accepted.
'0'	'1'	CODEBLOCK REJECTION: Even Number of Errors Detected The output of the AND gate is '0' so no correction is made. The output of the XOR gate CR is '1' so the codeblock is rejected.
'1'	'0'	CODEBLOCK REJECTION: Odd Number of Errors Detected in an Apparently Correct Codeblock The output of the AND gate is '0' so no correction is made. The output of the XOR gate CR is '1' so the codeblock is rejected.
'1'	'1'	Correction of Single Error: Odd Number of Errors Detected in an Incorrect Codeblock The output of the AND gate is '1' at the position where the contents of PLR are '000001', so correcting the single error. The output of the XOR gate CR is '0' so the codeblock is accepted.

### 6.5.3.3 The Obsolete Filler Bit Augmentation Algorithm

With the earlier Tail Sequence pattern (see 4.5.3), there is a higher risk of missing (failing to recognize) the Tail Sequence. If a single bit of this Tail Sequence becomes altered in the channel, the received sequence may no longer be recognized in SEC mode as uncorrectable, and an improper correction may take place. Because the expected CODEBLOCK REJECTION does not occur, the Tail Sequence is missed. Missing a Tail Sequence has far-reaching effects, since it is possible that the CLTU reception procedure will not be in SEARCH state for the next CLTU and therefore the entire subsequent CLTU may be missed.

The Filler Bit Augmentation algorithm was designed to reduce the problem significantly and improve the reliability of identifying a Tail Sequence. The algorithm uses the Filler Bit to selectively inhibit the SEC mode. Each valid BCH Codeblock terminates on a Filler Bit which is always set to '0' but the Tail Sequence terminates in a '1'. This fact can be exploited at the receiving end to inhibit the error correction function of the SEC mode

whenever the received Filler Bit is a ‘1’. When the algorithm is used, the decision in the last row of table 6-2 is expanded as shown in table 6-3. This algorithm substantially improves the ability to identify a Tail Sequence with the earlier Tail Sequence pattern.

NOTE – The earlier Tail Sequence pattern consists of an alternating pattern of ‘ones’ and ‘zeros’ starting with a ‘zero’, and so ends with a ‘one’. The current Tail Sequence pattern (see 4.5.1) also ends with a ‘one’ and is therefore compatible with a receiving end which uses the Filler Bit Augmentation algorithm.

**Table 6-3: Decoding Strategy Modified for Filler Bit Augmentation**

<b>FILLER BIT VALUE</b>	<b>DECISION WHEN FILLER BIT AUGMENTATION ALGORITHM IN USE</b>
‘0’	Correction of Single Error: Odd Number of Errors Detected in an Incorrect Codeblock The output of the AND gate is ‘1’ at the position where the contents of PLR are ‘000001’, so correcting the single error. The output of the XOR gate is ‘0’ and the Filler Bit is ‘0’ so the codeblock is accepted.
‘1’	CODEBLOCK REJECTION: Odd Number of Errors Detected and Filler Bit is ‘1’ The output of the AND gate is ‘1’ at the position where the contents of PLR are ‘000001’, so correcting the single error. The output of the XOR gate is ‘0’ but the Filler Bit is ‘1’ so the codeblock is rejected.
The table replaces the DECISION box in the last row of table 6-2.	

While the algorithm substantially reduces the likelihood of missing a Tail Sequence, there is a penalty in the form of a slight increase in frame rejection rate. A BCH Codeblock containing one error which would otherwise be corrected and accepted may also have an error in its Filler Bit. If this algorithm is not used, the erroneous Filler Bit is not tested and the codeblock is corrected and accepted. Under this algorithm, such a codeblock is rejected, even though it is correctable.

The current Tail Sequence pattern (see 4.5.1) has an improved performance compared to the earlier pattern. The distance to the nearest codewords is increased by one. That is, it differs in more bits from the nearest codewords, so it takes more channel errors to corrupt it into a correctable codeword. Therefore, the Filler Bit Augmentation algorithm is not included in the current issue of the Recommended Standard (reference [2]).

Section 8 includes information on the performance of the current Tail Sequence pattern and annex E gives statistics on the performance of obsolete features including the Filler Bit Augmentation algorithm.



## 6.6 INTERFACE TO DATA LINK PROTOCOL SUBLAYER

### 6.6.1 DEFINITION OF THE INTERFACE

#### 6.6.1.1 Formal Definition

The Recommended Standard (reference [2]) includes a formal definition of the service interface between the Synchronization and Channel Coding Sublayer and the Data Link Protocol Sublayer at the receiving end. The Synchronization and Channel Coding Sublayer uses the following service primitive:

ChannelAccess.indication (Frames)

to deliver data to the Data Link Protocol Sublayer. One instance of the ChannelAccess.indication corresponds to the processing of one CLTU by the CLTU reception procedure.

The Synchronization and Channel Coding Sublayer does not need to know the positions of any Frame boundaries within the data it extracts from a CLTU, nor does it use the length fields or other values in the TC Transfer Frames.

#### 6.6.1.2 Alternative Description

In the formal definition described in Recommended Standard (reference [2]), the interface consists of a single interaction where the Synchronization and Channel Coding Sublayer delivers all the data extracted from a CLTU. However, this definition does not exclude implementations where the data from a CLTU is delivered piece-by-piece, provided the Synchronization and Channel Coding Sublayer also delivers indications of the start and end of the CLTU. In this respect, it may be helpful to consider the following alternative description of the interface.

The Synchronization and Channel Coding Sublayer provides a ‘Data Start’ signal to the Data Link Protocol Sublayer, indicating that data are being transferred. The Data Start signal is set to ‘true’ while the Channel Coding Sublayer is in the process of actively transferring data octets. The first octet transferred after Data Start goes ‘true’ corresponds to the first octet of the first TC Transfer Frame. The Data Start signal is set to ‘false’ (indicating ‘Data Stop’) when the Synchronization and Channel Coding Sublayer stops transferring octets because of a decoder failure or channel deactivation.

### 6.6.2 DELIVERED FRAMES

#### 6.6.2.1 Overview

The Frames parameter of the ChannelAccess.indication consists of one or more TC Transfer Frames, but:

- any fill bits added at the sending end to complete the last BCH Codeblock of the CLTU have not yet been removed;
- the last Frame may be incomplete;
- the Frames have not yet been validated;
- the Spacecraft Identifier fields of the Frames have not yet been checked to verify that the Frames are addressed to the receiving spacecraft.

The Data Link Protocol Sublayer therefore has to perform various procedures on the received data to check that it is handling valid TC Transfer Frames.

#### **6.6.2.2 Length Field**

For each incoming CLTU, the Data Link Protocol Sublayer receives a block of data that is frame-synchronized; i.e., it begins on a frame boundary. The Frame Delimiting and Fill Removal Procedure in the Data Link Protocol Sublayer can therefore locate the Frame Length field of the first Frame. The procedure locates the positions of any following Frames in the block, discards truncated Frames and removes any fill bits. See reference [4] for a definition of the Frame Delimiting and Fill Removal Procedure.

#### **6.6.2.3 Frame Error Control Field**

Reference [4] defines an optional Frame Error Control Field in a TC Transfer Frame. The presence or absence of the field is a managed parameter for a Physical Channel.

The purpose of the field is to detect errors in the Frame which have not been detected by the Synchronization and Channel Coding Sublayer. If the field is present, it is checked by the Frame Validation Check Procedure in the Data Link Protocol Sublayer.

The Frame Error Control Field adds significant extra protection against errors, and its use is particularly advisable when the BCH decoding uses SEC mode. For example, if a received BCH Codeblock has more than two errors, the BCH decoding in SEC mode may detect a single error and make an erroneous correction. See the performance discussion in section 8.

#### **6.6.2.4 Spacecraft Identifier**

Spacecraft which have identical RF characteristics may both detect an RF signal carrying TC data and start the decoding and frame reconstruction activities. When the Frame reaches the Data Link Protocol Sublayer, the Frame Validation Check Procedure checks the Spacecraft Identifier field to determine if the TC Transfer Frame is meant for the receiver. If it is not, the Frame is ignored.

If the Frame contains errors, there is a risk that the Spacecraft Identifier field is corrupt. Therefore, the undetected error rate for the Spacecraft Identifier field must be good enough to protect the spacecraft against accepting foreign data.

## **6.7 RELATIONSHIP TO THE PLOPS**

### **6.7.1 PLOP-1**

When the sending end is operating PLOP-1, there is a period of CMM-1 (unmodulated carrier only) between CLTUs. At the receiving end, the CMM-1 causes a loss of bit synchronization in the Physical Layer, which signals CHANNEL DEACTIVATION (Event E2). Therefore, the CLTU reception procedure is forced back to the INACTIVE state (S1). So even if the CLTU reception procedure fails to detect the CLTU Tail Sequence, it is forced out of DECODE state (S3).

During the following Acquisition Sequence (CMM-2), the Physical Layer achieves bit synchronization and signals CHANNEL ACTIVATION (Event E1), sending the CLTU reception procedure to SEARCH state (S2), where it is ready to detect the Start Sequence of the next CLTU.

### **6.7.2 PLOP-2**

#### **6.7.2.1 Failing to Detect a Tail Sequence**

When the sending end is operating PLOP-2, a CLTU is followed by an optional amount of Idle Sequence (CMM-4) and then the next CLTU. The Physical Layer at the receiving end maintains bit synchronization and does not signal an event to the CLTU reception procedure. In this case, the successful operation of the procedure depends on the Tail Sequence causing a CODEBLOCK REJECTION (Event E4), to put the procedure to SEARCH state (S2) ready for the next CLTU.

There is a small risk that the Tail Sequence will not cause a CODEBLOCK REJECTION. If this happens, the CLTU reception procedure stays in DECODE state (S3) and tries to decode the eight octets following the Tail Sequence as a BCH Codeblock. It can therefore miss the beginning of the next CLTU. Section 8 includes performance data on the risk of failing to detect a Tail Sequence.

#### **6.7.2.2 Synchronization Lockout**

The Recommended Standard (reference [2]) also mentions another small risk with operating PLOP-2, which can be avoided by inserting a minimum Idle Sequence of one octet between CLTUs. The risk is synchronization lockout, where a failure during the reception of one CLTU can cause the loss of the following CLTU.

Synchronization lockout is caused by a combination of two events. The first event is the reception procedure staying in or entering SEARCH state (S2) following an error, such as:

- the Start Sequence of a CLTU is not detected;
- a faulty BCH codeblock (not the Tail Sequence) fails to decode.

The second event is the detection of a chance Start Sequence pattern over the last bits of the last BCH Codeblock of the CLTU and the first bits of its Tail Sequence, causing an erroneous change to DECODE state (S3). The CLTU reception procedure attempts to process the next 64 bits as a BCH Codeblock. The 64 bits are the remainder of the Tail Sequence and the following bits. If there is no Idle Sequence present, the following bits include the beginning of the Start Sequence of the following CLTU, causing the following CLTU to be lost.

NOTE – The missed Start Sequence on the following CLTU is caused by the reception procedure being in the wrong state, rather than by any properties of the received bits in the Start Sequence.

The risk of synchronization lockout is higher in systems which accept a Start Sequence with a single error. It is also increased in systems where the CLTU reception procedure is searching for the Start Sequence or its inverse.

The chance Start Sequence which triggers the second event is most likely to include only a few bits of the Tail Sequence. For example, the inverse Start Sequence ends with four '1's and the Tail Sequence starts with two '1's, so a Start Sequence might occur by chance (without corruption) across the last 14 bits of the last BCH codeblock and the first two bits of the Tail Sequence. With some corruption, five or six bits of the Tail Sequence could be in the chance Start Sequence. The eight-bit inserted Idle Sequence should be sufficient to provide protection, unless bit errors are so frequent that CLTU reception is unlikely to be working anyway.

## **7 PSEUDO-RANDOMIZATION**

### **7.1 OVERVIEW**

The purpose of randomization is to enable the receiving end to maintain bit synchronization with the received signal.

NOTE – For brevity, the word ‘random’ is used in place of ‘pseudo-random’ in this report.

To ensure proper receiver operation, the data stream must have a minimum bit transition density. That is, it must be sufficiently random. The randomizer specified in reference [2] is the preferred method to ensure sufficient randomness for all combinations of CCSDS-recommended modulation and coding schemes.

Reference [2] specifies that the randomizer is used unless the system designer verifies that the system will operate properly without it.

The presence or absence of randomization is fixed for a Physical Channel and is a managed parameter.

NOTE – It may be helpful to have a way to temporarily disable randomization during early pre-launch testing (e.g., board-level checkout, spacecraft integration, environmental testing), so that integration and testing personnel can read the binary data that they are familiar with. This can make data validation and troubleshooting easier.

The randomization procedure consists of exclusive-ORing a random bit pattern with the data. When the procedure is repeated, the result is the original data. Therefore, the derandomization procedure at the receiving end is the same as the randomization procedure at the sending end.

### **7.2 RELATIONSHIP TO OTHER PROCEDURES**

#### **7.2.1 AT THE SENDING END**

At the sending end, the TC Transfer Frames received from the Data Link Protocol Sublayer are randomized before the BCH encoding.

As described in 3.4, the Frame data are placed, seven octets (56 bits) at a time, into a set of BCH Codeblocks. If randomization is in use for the Physical Channel, then the data octets from the Frames are randomized before being placed in the BCH Codeblocks.

At the end of the Frames, there may not be exactly seven octets left, so in this case the Information field of the last BCH Codeblock in a CLTU is filled with a pattern of fill bits. If randomization is in use for the Physical Channel, then the randomization of the fill bits is optional. Whichever option is chosen, the 56 bits in the Information field of a BCH Codeblock must be the same as the 56 bits used to generate the parity bits for the codeblock.

The Error Control Field of a BCH Codeblock is not randomized.

Randomization is applied during the generation of BCH Codeblocks and does not affect the generation of CLTUs or the PLOPs, so the following fields are not randomized:

- the Start Sequence of a CLTU;
- the Tail Sequence of a CLTU;
- the Acquisition Sequence transmitted by a PLOP;
- the Idle Sequence transmitted by a PLOP.

### 7.2.2 AT THE RECEIVING END

At the receiving end, derandomization takes place after the BCH decoding. The CLTU reception procedure derandomizes the data extracted from successfully decoded BCH Codeblocks, before passing the data to the Data Link Protocol Sublayer.

The derandomization is applied to the 56-bit Information field of a BCH Codeblock. Therefore, any fill bits in the Information field of the last BCH Codeblock of a CLTU are derandomized, whether they were randomized or not. One of the tasks of the Data Link Protocol Sublayer is to discard any fill bits following the end of the last Frame, as described in 6.6.2.

NOTE – An octet of fill bits has the hexadecimal value 55, as described in 3.4. In a system which uses randomization but which does not randomize the fill bits, the values of the fill bits are changed by the derandomization process. In this case, the octets of fill bits which are delivered to the Data Link Protocol Sublayer do not have the hexadecimal value 55. As a CLTU can have up to six octets of fill bits and the minimum length of a Frame is six octets, there is a small risk that the Data Link Protocol Sublayer could mistake the fill bits for a Frame. The risk only applies in systems which use multiple Frames per CLTU. The validation by the Data Link Protocol Sublayer of the Frame Length field and other fields is likely to detect that the fill bits are not a valid Frame.

## 7.3 RANDOMIZATION PROCEDURE

### 7.3.1 THE RANDOM SEQUENCE

The random sequence is generated by the Bit Transition Generator (BTG), which uses a linear feedback shift register with eight bits. The bits of the shift register are all initialized to '1'.

Figure 7-1, which is reproduced from reference [2], shows the logic diagram for the BTG. The logic can be represented by the polynomial:

$$h(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$$

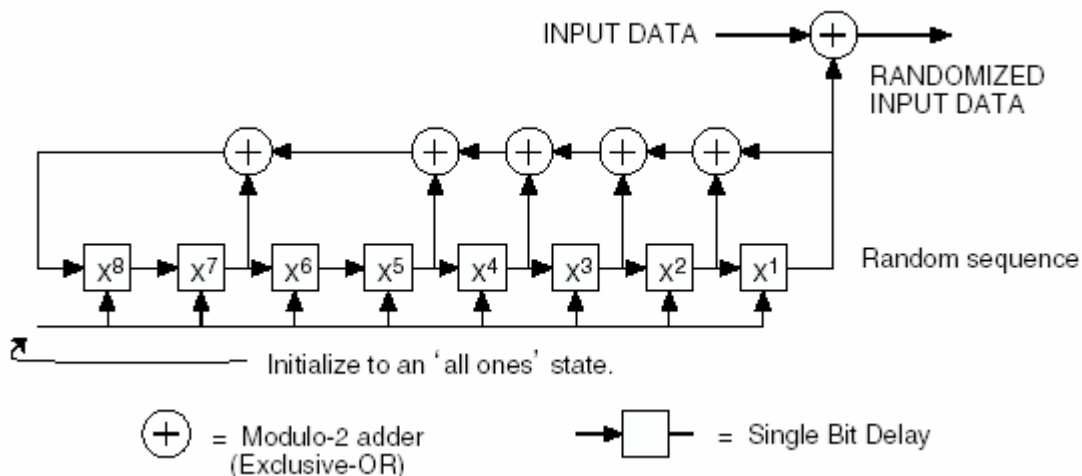
The first 40 bits of the generated sequence are:

1111 1111 0011 1001 1001 1110 0101 1010 0110 1000

The generated sequence repeats after 255 bits.

NOTE – The maximum repetition period of a sequence generated by an  $m$ -bit linear feedback shift register is  $2^m - 1$ . In this case  $m = 8$ .

The random sequence is restarted for each CLTU.



**Figure 7-1: Logic Diagram for the Bit Transition Generator**

### 7.3.2 APPLYING THE RANDOM SEQUENCE AT THE SENDING END

The random sequence is restarted for each CLTU. Therefore, at the sending end, the shift registers of the BTG are all reinitialized to ‘1’ at the start of processing for each new ChannelAccess.request from the Data Link Protocol Sublayer.

The first bit of the first TC Transfer Frame in the Frames parameter of the request is exclusive-ORed with the first bit of the random sequence. Then the second bit of the first TC Transfer Frame is exclusive-ORed with the second bit of the random sequence. And so on until the end of the Frames and (optionally) the fill bits is reached.

The boundaries between Frames are ignored. The BTG is not reinitialized until the start of processing for the next ChannelAccess.request.

### **7.3.3 APPLYING THE RANDOM SEQUENCE AT THE RECEIVING END**

The random sequence is restarted for each CLTU. Therefore, at the receiving end, the shift registers of the BTG are all reinitialized to '1' whenever the CLTU reception procedure (see section 6) finds a CLTU Start Sequence (Event E3, START SEQUENCE FOUND).

The 56 Information bits from the first decoded BCH Codeblock are exclusive-ORed with the first 56 bits of the random sequence. Then the 56 Information bits from the next decoded BCH Codeblock are exclusive-ORed with the next 56 bits of the random sequence. And so on until the end of the last successfully decoded BCH Codeblock in the CLTU. The processing for the CLTU ends with a CODEBLOCK REJECTION (Event E4) or a CHANNEL DEACTIVATION (Event E2).

## **7.4 BACKGROUND**

### **7.4.1 ADDING RANDOMIZATION TO THE RECOMMENDATIONS**

Early CCSDS recommendations for Telecommand Channel Coding did not include randomization. Telecommand data tended to consist of short, discrete commands with frequent transitions. Then there was a shift towards larger volumes of telecommand data, including lengthy transfers of large areas of computer memory, parts of which have few transitions. This led to problems in systems that rely on frequent bit transitions in the data to maintain bit synchronization.

Randomization was added to the CCSDS recommendations for Telecommand Channel Coding published in November 1995.

### **7.4.2 RELATIONSHIP TO TM RANDOMIZATION**

The randomization in reference [2] is not the same as the randomization for TM Synchronization and Channel Coding in reference [5]. The basic principle is the same but the pseudo-random bit pattern and its underlying generator polynomial are different.

The TM randomizer was found to have some minor interactive properties with the Reed-Solomon code used in TM for forward error correction. The randomizer chosen for telecommand avoids this interaction. Annex C includes the theoretical background to the choice of the generator polynomial for the randomizer specified in reference [2].

There is also a difference in the position of the randomization relative to the other procedures. In TC (reference [2]) the randomization is before the BCH encoding but in TM (reference [5]) the randomization is after the Reed-Solomon or Turbo encoding and before any Convolutional encoding.



## **8 PERFORMANCE DATA**

### **8.1 INTRODUCTION**

The objective of the TC Synchronization and Channel Coding Sublayer is to provide a reliable service for the delivery of TC Transfer Frames.

This section defines the performance criteria for a reliable service. It provides performance data on the coding options in the Recommended Standard (reference [2]) to enable the system engineer to select the coding parameters and other features so that the performance criteria are satisfied. Because the criteria are measured in frames, the features considered here extend into the Data Link Protocol Sublayer.

Annex F provides performance data for obsolete synchronization and coding features not included in the current issue of the Recommended Standard (reference [2]).

NOTE – Throughout this section, a binary symmetric channel with additive white Gaussian noise is assumed. Performance statistics are shown for three different channel bit error rates:  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$ . In equations, the channel bit error rate is represented by the letter  $p$ .

### **8.2 PERFORMANCE CRITERIA**

#### **8.2.1 GENERAL**

The unit for measuring the performance is the TC Transfer Frame. A reliable service for the delivery of TC Transfer Frames is considered to be achieved when the following two performance criteria are simultaneously met.

#### **8.2.2 TC TRANSFER FRAME REJECTION RATE**

A maximum of one TC Transfer Frame is deleted (rejected) for every  $10^3$  frames transmitted. This operating point is defined to be the Command Threshold.

#### **8.2.3 TC TRANSFER FRAME UNDETECTED ERROR RATE**

A maximum of one TC Transfer Frame for every  $10^9$  frames transmitted is erroneously accepted (that is, contains one or more undetected bit errors).

### **8.3 PERFORMANCE COMPONENTS**

To support the performance criteria, the Recommended Standards (references [2] and [4]) specify procedures for handling the TC Transfer Frame and related structures:

- PLOP-1 or PLOP-2 and the Acquisition Sequence to enable the receiving end to acquire bit synchronization;
- optional randomization to ensure that the receiving end can maintain bit synchronization;
- placing a Start Sequence at the start of each CLTU so that the CLTU reception procedure can delimit the beginning of the first BCH Codeblock (and therefore the beginning of the first frame) in the CLTU;
- BCH encoding/decoding of the codeblocks to provide error detection and/or correction;
- placing a Tail Sequence at the end of each CLTU so that the CLTU reception procedure can detect the end of the CLTU and so start searching for the Start Sequence of the next CLTU;
- Frame Delimiting and Fill Removal Procedure (in the Data Link Protocol Sublayer at the receiving end) to delimit the end of the first TC Transfer Frame in the data extracted from the CLTU and delimit the beginning and end of each subsequent frame in the data;
- Frame Validation Check Procedure (in the Data Link Protocol Sublayer at the receiving end) to check the mandatory fields in the header of each TC Transfer Frame, and to check the optional Frame Error Control Field which provides additional protection against undetected errors.

## **8.4 FACTORS AFFECTING FRAME REJECTION RATE**

### **8.4.1 GENERAL**

The following factors affect the frame rejection rate, where a frame is lost or deleted because of errors on the channel.

### **8.4.2 BIT SYNCHRONIZATION FACTOR**

As described in section 5, the PLOP in the Physical Layer at the sending end transmits CMM-1 (unmodulated carrier mode) at the start of a communications session. The transponder in the Physical Layer at the receiving end locks onto the signal.

Then the PLOP transmits the Acquisition Sequence (CMM-2) with the length selected for the mission. The Acquisition Sequence, consisting of alternating ‘1’ and ‘0’ bits, has maximum transition density, to provide the fastest lock-up time for the on-board bit synchronizer.

The preferred minimum length of 128 bits was chosen to provide 0.9999 probability of acquisition of bit synchronization, based on experience with a number of hardware implementations operating at the command threshold level. This length may be modified as needed to suit different hardware characteristics or channel bit error rates.

Because the probability of achieving bit synchronization is primarily hardware dependent, it is not considered further in the performance analysis in this section. The remainder of this section assumes that bit synchronization is achieved whenever the Acquisition Sequence is transmitted. As described in section 6, whenever bit synchronization is achieved, the Physical Layer signals CHANNEL ACTIVATION (Event E1), and the CLTU reception procedure goes to SEARCH state (S2).

### **8.4.3 CLTU START SEQUENCE FACTORS**

#### **8.4.3.1 General**

There are two CLTU Start Sequence factors:

- the probability that the CLTU reception procedure misses a Start Sequence because it is not in SEARCH state (S2) when the Start Sequence is received;
- the probability that the Start Sequence is not recognized by the CLTU reception procedure in SEARCH state (S2).

#### **8.4.3.2 CLTU Reception Procedure Not in SEARCH State**

If the Start Sequence follows an Acquisition Sequence (plus an optional length of Idle Sequence), then the SEARCH state of the CLTU reception procedure depends on successfully achieving bit synchronization. As described in 8.4.2, this performance analysis assumes that bit synchronization is achieved.

If the sending end is operating the PLOP-1 procedure, then the Start Sequence always follows an Acquisition Sequence, so this performance analysis assumes that the CLTU reception procedure is in SEARCH state when receiving the Start Sequence.

If the sending end is operating the PLOP-2 procedure, then the Start Sequence follows the Tail Sequence of the previous CLTU (with an optional length of Idle Sequence between the Tail Sequence and the Start Sequence). The synchronization lockout described in 6.7.2.2 can cause a Start Sequence to be missed, but the probability is not considered here. The probability that the CLTU reception procedure is not in SEARCH state when receiving the Start Sequence therefore depends on the probability of failing to recognize the Tail Sequence—see 8.4.5.

NOTE – For PLOP-1 and PLOP-2 there is the remote possibility that the proper Start Sequence is missed because of a false (premature) start when the Start Sequence is erroneously recognized in the Acquisition Sequence / Idle Sequence pattern. The Start Sequence has a distance of six bits from the Acquisition Sequence / Idle Sequence, so it would require six changed bits to convert a section of the Acquisition Sequence / Idle Sequence into a Start Sequence. If the option to accept the Start Sequence with one error is in use, then a change of five bits could be enough to cause a premature start.

**8.4.3.3 Start Sequence Not Recognized**

The Start Sequence is a 16-bit synchronization pattern. The CLTU reception procedure in SEARCH state searches the received stream of bits, looking for the Start Sequence pattern.

As described in 6.4.3, there are two options in the CLTU reception procedure for the handling of errors in the Start Sequence. Either:

- X no errors are accepted, or
- Y a single error is accepted.

The options are indicated by subscripts X and Y in the following discussion.

If no errors are accepted, then the received bit pattern is accepted as a valid Start Sequence only if it exactly matches the expected bit pattern. In this case, the probability  $P_{SX}$  of failing to recognize the Start Sequence is therefore the probability that one or more bit errors fall on any of the 16 bits of the Start Sequence:

$$P_{SX} = 1 - (1-p)^{16} \tag{1}$$

If a single error is accepted, then the received bit pattern is accepted as a valid Start Sequence either if it exactly matches the expected bit pattern or if it differs by one bit from the expected bit pattern. In this case, the probability  $P_{SY}$  of failing to recognize the Start Sequence is therefore the probability that two or more bit errors appear in the 16 bits of the Start Sequence:

$$P_{SY} = 1 - [(1-p)^{16} + 16p(1-p)^{15}] \tag{2}$$

Table 8-1 shows the values of the probabilities  $P_{SX}$  and  $P_{SY}$  for the three channel bit error rates.

**Table 8-1: Probability of Not Recognizing the Start Sequence**

Case	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
$P_{SX}$ no errors accepted	$1.60 \times 10^{-3}$	$1.60 \times 10^{-4}$	$1.60 \times 10^{-5}$
$P_{SY}$ single error accepted	$1.20 \times 10^{-6}$	$1.20 \times 10^{-8}$	$1.20 \times 10^{-10}$

## 8.4.4 BCH CODEBLOCK FACTOR

### 8.4.4.1 Decoding the BCH Codeblocks

The BCH Codeblocks of a CLTU are decoded one by one, as long as each codeblock is successfully decoded. If a BCH Codeblock causes a codeblock rejection, the remaining codeblocks of the CLTU are discarded.

The probability of a codeblock rejection in a CLTU increases with the number of BCH Codeblocks in the CLTU.

As described in 6.5, there are two options in the CLTU reception procedure for the decoding of BCH codeblocks. Either

- X Triple Error Detection (TED) mode is used, or
- Y Single Error Correction (SEC) mode is used.

When TED mode is in use, then no error is allowed in the Start Sequence, as in option X in section 8.4.3.3 above. Similarly, when SEC mode is used, a single error is accepted in the Start Sequence. The options are indicated by subscripts X and Y in the following discussion.

### 8.4.4.2 Codeblock Rejection in TED Mode

In TED mode, the probability  $P_{CX}$  of one of the BCH Codeblocks causing a codeblock rejection is:

$$P_{CX} = 1 - [(1 - p)^{63}]^N \quad (3)$$

where N is the number of codeblocks in the CLTU. Table 8-2 shows the values of the probability  $P_{CX}$  for the three channel bit error rates.

**Table 8-2: Probability  $P_{CX}$  of Codeblock Rejection in TED Mode**

Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
1	$6.28 \times 10^{-3}$	$6.30 \times 10^{-4}$	$6.30 \times 10^{-5}$
2	$1.25 \times 10^{-2}$	$1.26 \times 10^{-3}$	$1.26 \times 10^{-4}$
4	$2.49 \times 10^{-2}$	$2.52 \times 10^{-3}$	$2.52 \times 10^{-4}$
6	$3.71 \times 10^{-2}$	$3.77 \times 10^{-3}$	$3.78 \times 10^{-4}$
9	$5.51 \times 10^{-2}$	$5.65 \times 10^{-3}$	$5.67 \times 10^{-4}$
12	$7.28 \times 10^{-2}$	$7.53 \times 10^{-3}$	$7.56 \times 10^{-4}$
16	$9.59 \times 10^{-2}$	$1.00 \times 10^{-2}$	$1.01 \times 10^{-3}$
20	$1.18 \times 10^{-1}$	$1.25 \times 10^{-2}$	$1.26 \times 10^{-3}$
24	$1.40 \times 10^{-1}$	$1.50 \times 10^{-2}$	$1.51 \times 10^{-3}$
28	$1.62 \times 10^{-1}$	$1.75 \times 10^{-2}$	$1.76 \times 10^{-3}$
32	$1.83 \times 10^{-1}$	$2.00 \times 10^{-2}$	$2.01 \times 10^{-3}$
36	$2.03 \times 10^{-1}$	$2.24 \times 10^{-2}$	$2.27 \times 10^{-3}$
37	$2.08 \times 10^{-1}$	$2.30 \times 10^{-2}$	$2.33 \times 10^{-3}$
74	$3.73 \times 10^{-1}$	$4.56 \times 10^{-2}$	$4.65 \times 10^{-3}$
110	$5.00 \times 10^{-1}$	$6.70 \times 10^{-2}$	$6.91 \times 10^{-3}$
147	$6.04 \times 10^{-1}$	$8.85 \times 10^{-2}$	$9.22 \times 10^{-3}$
293	$8.42 \times 10^{-1}$	$1.69 \times 10^{-1}$	$1.83 \times 10^{-2}$
439	$9.37 \times 10^{-1}$	$2.42 \times 10^{-1}$	$2.73 \times 10^{-2}$
586	$9.75 \times 10^{-1}$	$3.09 \times 10^{-1}$	$3.62 \times 10^{-2}$

#### 8.4.4.3 Codeblock Rejection in SEC Mode

Two of the values used in the decoding process are the error parity check, PAR, and the syndrome, SYND, shown in table 8-3. In SEC mode, the values are used to determine the actions of the decoder, and they lead to six possible cases, as shown in table 8-4.

**Table 8-3: Meaning of Decoding Values**

<b>Error parity check</b>	PAR='0'	the number of errors is even, 0, 2, 4, 6, ...
	PAR='1'	the number of errors is odd, 1, 3, 5, 7, ...
<b>Syndrome</b>	SYND=0	the received codeword is a valid codeword, though not necessarily the one that was transmitted
	SYND>0	the received codeword is not valid, so errors are present

**Table 8-4: Decoding Cases in SEC Mode**

PAR	SYND	Action	Case	
'0'	0	accept codeblock	A1	the received codeblock has no errors
			A2	the received codeblock has an even number of errors (at least 4)
'0'	>0	codeblock rejection	B	the received codeblock has an even number of errors (at least 2)
'1'	0	codeblock rejection	C	the received codeblock has an odd number of errors (at least 3)
'1'	>0	correct single error and accept codeblock	D1	the corrected codeblock has no errors
			D2	the corrected codeblock has an even number of errors

In cases A1 and D1, the SEC decoding is successful and delivers an error-free codeblock. In cases A2 and D2, the SEC decoding delivers a codeblock containing undetected errors: these cases are considered further in 8.5 below. Cases B and C lead to codeblock rejection and are considered here.

The probabilities  $P_B$  and  $P_C$  for cases B and C are given by:

$$P_B = P_2 + P_4 (1-R_4) + P_6 (1-R_6) + \dots$$

$$P_C = P_3 (1-R_3) + P_5 (1-R_5) + \dots$$

where  $P_i$  is the probability of  $i$  errors and  $R_i$  is the portion of these errors that are not detected. Adding the two equations gives:

$$P_B + P_C = P_2 + P_3 (1-R_3) + P_4 (1-R_4) + P_5 (1-R_5) + P_6 (1-R_6) + \dots$$

To three significant figures, this gives similar results to the general expression for the probability of two or more errors in a codeblock. So the probability  $P_{CY}$  of one of the BCH Codeblocks causing a codeblock rejection can be written as:

$$P_{CY} = 1 - [(1 - p)^{63} + 63p(1 - p)^{62}]^N \quad (4)$$

where N is the number of codeblocks in the CLTU. Table 8-5 shows the values of the probability  $P_{CY}$  for the three channel bit error rates.

**Table 8-5: Probability  $P_{CY}$  of Codeblock Rejection in SEC Mode**

Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
1	$1.95 \times 10^{-5}$	$1.95 \times 10^{-7}$	$1.95 \times 10^{-9}$
2	$3.89 \times 10^{-5}$	$3.90 \times 10^{-7}$	$3.91 \times 10^{-9}$
4	$7.78 \times 10^{-5}$	$7.81 \times 10^{-7}$	$7.81 \times 10^{-9}$
6	$1.17 \times 10^{-4}$	$1.17 \times 10^{-6}$	$1.17 \times 10^{-8}$
9	$1.75 \times 10^{-4}$	$1.76 \times 10^{-6}$	$1.76 \times 10^{-8}$
12	$2.33 \times 10^{-4}$	$2.34 \times 10^{-6}$	$2.34 \times 10^{-8}$
16	$3.11 \times 10^{-4}$	$3.12 \times 10^{-6}$	$3.12 \times 10^{-8}$
20	$3.89 \times 10^{-4}$	$3.90 \times 10^{-6}$	$3.91 \times 10^{-8}$
24	$4.67 \times 10^{-4}$	$4.69 \times 10^{-6}$	$4.69 \times 10^{-8}$
28	$5.44 \times 10^{-4}$	$5.47 \times 10^{-6}$	$5.47 \times 10^{-8}$
32	$6.22 \times 10^{-4}$	$6.25 \times 10^{-6}$	$6.25 \times 10^{-8}$
36	$7.00 \times 10^{-4}$	$7.03 \times 10^{-6}$	$7.03 \times 10^{-8}$
37	$7.19 \times 10^{-4}$	$7.22 \times 10^{-6}$	$7.23 \times 10^{-8}$
74	$1.44 \times 10^{-3}$	$1.44 \times 10^{-5}$	$1.45 \times 10^{-7}$
110	$2.14 \times 10^{-3}$	$2.15 \times 10^{-5}$	$2.15 \times 10^{-7}$
147	$2.86 \times 10^{-3}$	$2.87 \times 10^{-5}$	$2.87 \times 10^{-7}$
293	$5.68 \times 10^{-3}$	$5.72 \times 10^{-5}$	$5.72 \times 10^{-7}$
439	$8.50 \times 10^{-3}$	$8.57 \times 10^{-5}$	$8.57 \times 10^{-7}$
586	$1.13 \times 10^{-2}$	$1.14 \times 10^{-4}$	$1.14 \times 10^{-6}$



### 8.4.5 TAIL SEQUENCE FACTOR

There is a risk of missing (failing to recognize) the Tail Sequence. If errors in the channel alter the Tail Sequence, the received sequence may be mistakenly accepted as a valid or correctable codeblock. Because the expected CODEBLOCK REJECTION does not occur, the Tail Sequence is missed. In this case, it is possible that the CLTU reception procedure will not be in SEARCH state for the next CLTU and therefore the subsequent CLTU may be missed.

NOTE – The description here of the probability of missing a Tail Sequence applies to the Tail Sequence pattern defined in the current issue of the Recommended Standard (reference [2]). See annex E for the probability for the earlier Tail Sequence pattern.

The two options in the CLTU reception procedure for the decoding of BCH codeblocks have different probabilities of missing a Tail Sequence. Either

- X Triple Error Detection (TED) mode is used, or
- Y Single Error Correction (SEC) mode is used.

Two of the values in decoding the BCH code are the error parity check, PAR, and the syndrome, SYND, shown in table 8-3. When decoding in TED mode, the codeblock is accepted only if PAR='0' and SYND=0. When decoding in SEC mode:

- the codeblock is accepted if PAR='0' and SYND=0; or
- one error is corrected and the codeblock is accepted if PAR='1' and SYND>0.

The PAR and SYND values for the Tail Sequence pattern with different numbers of errors,  $e = 0, 1, 2$  or  $3$ , are shown in table 8-6. If a value,  $t$ , in the table can lead to the mistaken acceptance of the Tail Sequence as a codeblock, then its contribution to the probability of a missed Tail Sequence is given by the expression:

$$t p^e (1 - p)^{63-e} \quad (5)$$

Numbers of errors greater than three are not shown, because they do not affect the three significant figures for the probabilities in table 8-7.

**Table 8-6: Parity and Syndrome When Tail Sequence Has Errors**

	Number of errors, $e$			
	$e = 0$	$e = 1$	$e = 2$	$e = 3$
PAR='0', SYND=0	-	-	-	651
PAR='0', SYND>0	-	63	-	39060
PAR='1', SYND=0	1	-	-	-
PAR='1', SYND>0	-	-	1953	-
Total combinations	1	63	1953	39711

When there are no errors, the Tail Sequence gives PAR='1' and SYND=0, so it is rejected in TED and SEC modes. With all 63 possible cases of a single error, the Tail Sequence gives PAR='0' and SYND>0 and is rejected in TED and SEC modes.

There are 1953 combinations of two errors, all of which give PAR='1' and SYND>0. These are rejected in TED mode but are erroneously corrected and accepted in SEC mode, causing a missed Tail Sequence.

Most of the 39711 combinations of three errors cause the Tail Sequence to be rejected in TED and SEC modes, but there are 651 combinations which give PAR='0' and SYND=0 and are accepted in both modes, causing a missed Tail Sequence.

Using expression 5 above, the probability  $P_{TX}$  of missing the Tail Sequence in TED mode is given by:

$$P_{TX} = 651 p^3 (1 - p)^{60} \quad (6)$$

The probability  $P_{TY}$  of missing the Tail Sequence in SEC mode is given by:

$$P_{TY} = 1953 p^2 (1 - p)^{61} + 651 p^3 (1 - p)^{60} \quad (7)$$

Table 8-7 shows the values of the probabilities  $P_{TX}$  and  $P_{TY}$  for the three channel bit error rates.

**Table 8-7: Probability of Missing the Tail Sequence**

Case	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
$P_{TX}$ error detection mode	$6.47 \times 10^{-10}$	$6.51 \times 10^{-13}$	$6.51 \times 10^{-16}$
$P_{TY}$ error correction mode	$1.94 \times 10^{-5}$	$1.95 \times 10^{-7}$	$1.95 \times 10^{-9}$

## 8.4.6 FRAMES AND CLTUs

### 8.4.6.1 General

The probabilities discussed in subsections 8.4.2 to 8.4.5 affect CLTU rejection. However, in the performance criteria in 8.2, the TC Transfer Frame is the accounting entity.

If there is only one TC Transfer Frame in a CLTU, the probability of frame rejection is the same as the probability of rejection of the CLTU.

If there are multiple TC Transfer Frames in a CLTU, the acceptance of a frame depends on the successful acceptance of the codeblocks which precede it in the CLTU. If the frame ends in codeblock  $M$ , then the probability of frame rejection is the same as the probability of rejection of a CLTU with  $M$  codeblocks. For the last frame in a CLTU, the probability of frame rejection is the same as the probability of rejection of the CLTU. As this is the worst case for the frames in the CLTU, it appears prudent to base performance comparisons on this value.

CLTUs may be sent either as independent entities (PLOP-1, where modulation is dropped between CLTUs) or as a sequence (PLOP-2, where bit synchronization is maintained on the channel). The choice of PLOP-1 or PLOP-2 contributes to the CLTU rejection probability.

### 8.4.6.2 Independent CLTUs (PLOP-1)

In calculating the rejection probability with independent CLTUs, two events are considered:

- the CLTU Start Sequence is not recognized;
- the CLTU Start Sequence is recognized but one of the codeblocks is rejected.

The probability of CLTU rejection depends on which options are in use in the CLTU reception procedure:

- X no errors accepted in the Start Sequence and codeblocks decoded in TED mode, or
- Y a single error accepted in the Start Sequence and codeblocks decoded in SEC mode.

For a frame which is the last or only frame in a CLTU, the corresponding frame rejection probabilities,  $P_{FX}$  and  $P_{FY}$ , are given by:

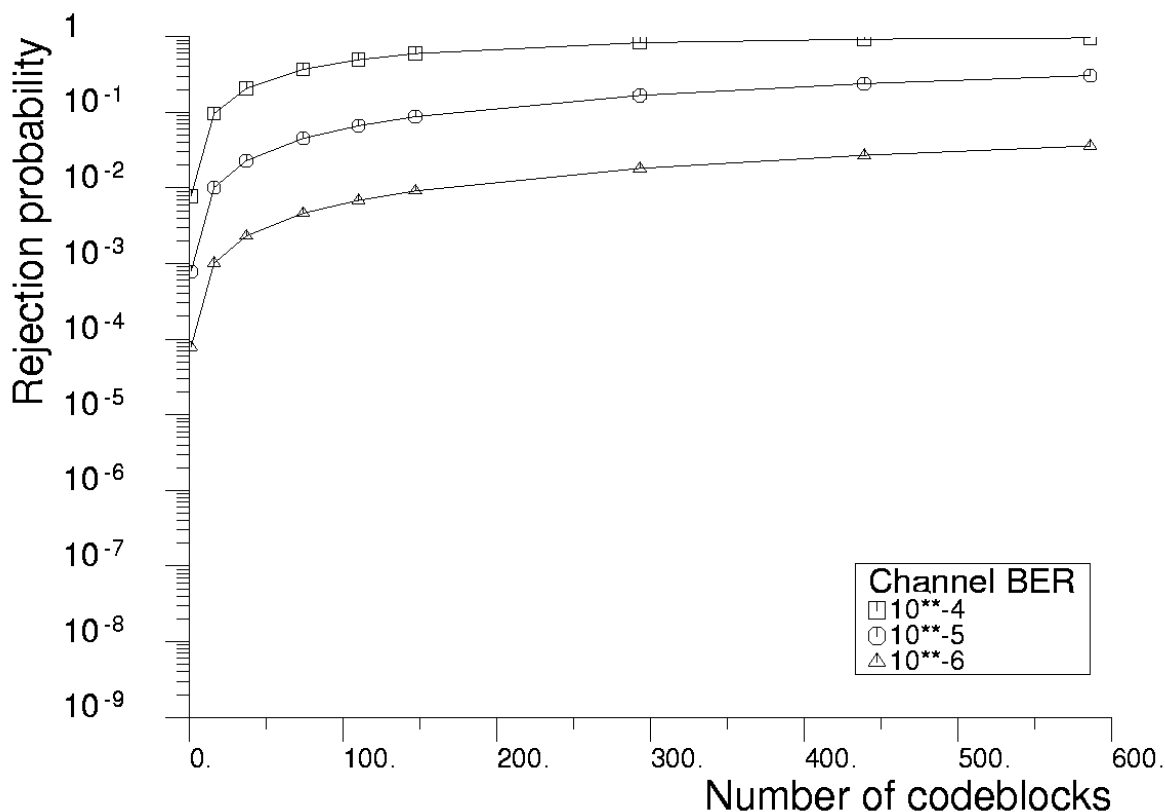
$$P_{FX} = P_{SX} + (1 - P_{SX}) P_{CX}$$

$$P_{FY} = P_{SY} + (1 - P_{SY}) P_{CY}$$

Table 8-8 shows the values of the probabilities  $P_{FX}$  and  $P_{FY}$  for the three channel bit error rates. The table includes values for long CLTUs carrying multiple maximum-length TC Transfer Frames. For example, a CLTU with 293 codeblocks can carry up to 2051 octets, which accommodates two 1024-octet frames.

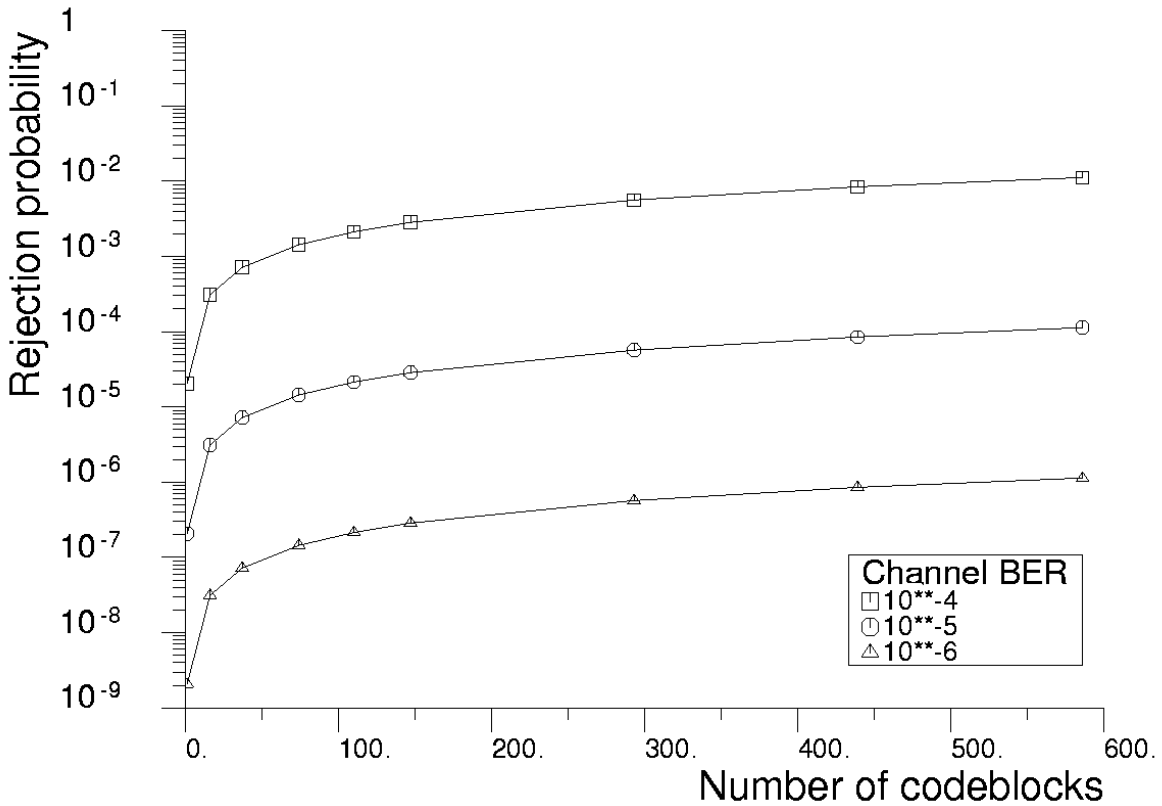
**Table 8-8: Frame Rejection Probabilities,  $P_{FX}$  and  $P_{FY}$ , for the Last or Only Frame in an Independent CLTU (PLOP-1)**

Number of codeblocks in the CLTU	Total frame lengths up to (octets)	$P_{FX}$ $P_{FY}$	Channel Bit Error Rate		
			$10^{-4}$	$10^{-5}$	$10^{-6}$
1	7	$P_{FX}$ (TED)	$7.87 \times 10^{-3}$	$7.90 \times 10^{-4}$	$7.90 \times 10^{-5}$
		$P_{FY}$ (SEC)	$2.06 \times 10^{-5}$	$2.07 \times 10^{-7}$	$2.07 \times 10^{-9}$
16	112	$P_{FX}$ (TED)	$9.73 \times 10^{-2}$	$1.02 \times 10^{-2}$	$1.02 \times 10^{-3}$
		$P_{FY}$ (SEC)	$3.12 \times 10^{-4}$	$3.14 \times 10^{-6}$	$3.14 \times 10^{-8}$
37	259	$P_{FX}$ (TED)	$2.09 \times 10^{-1}$	$2.32 \times 10^{-2}$	$2.34 \times 10^{-3}$
		$P_{FY}$ (SEC)	$7.21 \times 10^{-4}$	$7.24 \times 10^{-6}$	$7.24 \times 10^{-8}$
74	518	$P_{FX}$ (TED)	$3.74 \times 10^{-1}$	$4.57 \times 10^{-2}$	$4.67 \times 10^{-3}$
		$P_{FY}$ (SEC)	$1.44 \times 10^{-3}$	$1.45 \times 10^{-5}$	$1.45 \times 10^{-7}$
110	770	$P_{FX}$ (TED)	$5.01 \times 10^{-1}$	$6.71 \times 10^{-2}$	$6.92 \times 10^{-3}$
		$P_{FY}$ (SEC)	$2.14 \times 10^{-3}$	$2.15 \times 10^{-5}$	$2.15 \times 10^{-7}$
147	1029	$P_{FX}$ (TED)	$6.05 \times 10^{-1}$	$8.86 \times 10^{-2}$	$9.23 \times 10^{-3}$
		$P_{FY}$ (SEC)	$2.86 \times 10^{-3}$	$2.87 \times 10^{-5}$	$2.87 \times 10^{-7}$
293	2051	$P_{FX}$ (TED)	$8.42 \times 10^{-1}$	$1.69 \times 10^{-1}$	$1.83 \times 10^{-2}$
		$P_{FY}$ (SEC)	$5.68 \times 10^{-3}$	$5.72 \times 10^{-5}$	$5.72 \times 10^{-7}$
439	3073	$P_{FX}$ (TED)	$9.37 \times 10^{-1}$	$2.42 \times 10^{-1}$	$2.73 \times 10^{-2}$
		$P_{FY}$ (SEC)	$8.50 \times 10^{-3}$	$8.57 \times 10^{-5}$	$8.57 \times 10^{-7}$
586	4102	$P_{FX}$ (TED)	$9.75 \times 10^{-1}$	$3.09 \times 10^{-1}$	$3.63 \times 10^{-2}$
		$P_{FY}$ (SEC)	$1.13 \times 10^{-2}$	$1.14 \times 10^{-4}$	$1.14 \times 10^{-6}$



**Figure 8-1: Frame Rejection Probability in TED Mode,  $P_{FX}$  or  $P_{F2X}$ , for the Last or Only Frame in a CLTU (PLOP-1 or PLOP-2)**

Figure 8-1 shows the probability  $P_{FX}$  of frame rejection in TED mode. The values apply for the last or only frame in an independent CLTU (PLOP-1, table 8-8). Within the accuracy of the tables and figures, the values in TED mode are the same as the probability  $P_{F2X}$  when the CLTU is in a sequence of CLTUs (PLOP-2, table 8-9).



**Figure 8-2: Frame Rejection Probability in SEC Mode,  $P_{FY}$ , for the Last or Only Frame in an Independent CLTU (PLOP-1)**

Figure 8-2 shows the probability  $P_{FY}$  of frame rejection in SEC mode. The values apply for the last or only frame in an independent CLTU (PLOP-1, table 8-8). See figure 8-3 below for the probability in SEC mode when the CLTU is in a sequence of CLTUs (PLOP-2).

### 8.4.6.3 Sequence of CLTUs (PLOP-2)

If the sending end is operating the PLOP-2 procedure, then the Start Sequence follows the Tail Sequence of the previous CLTU (with an optional length of Idle Sequence between the Tail Sequence and the Start Sequence). Failure of the CLTU reception procedure to recognize the Tail Sequence of a CLTU can cause the following CLTU to be missed.

In calculating the rejection probability with a sequence of CLTUs, three events are considered:

- the CLTU reception procedure is not in SEARCH state when the Start Sequence arrives, because the previous Tail Sequence was missed;
- the CLTU Start Sequence is not recognized;
- the CLTU Start Sequence is recognized but one of the codeblocks is rejected.

The probability of CLTU rejection depends on which options are in use in the CLTU reception procedure:

- X no errors accepted in the Start Sequence and codeblocks decoded in TED mode, or
- Y a single error accepted in the Start Sequence and codeblocks decoded in SEC mode.

For a frame which is the last or only frame in a CLTU, the corresponding frame rejection probabilities,  $P_{F2X}$  and  $P_{F2Y}$ , are given by:

$$P_{F2X} = P_{TX} + (1 - P_{TX}) ( P_{SX} + (1 - P_{SX}) P_{CX} )$$

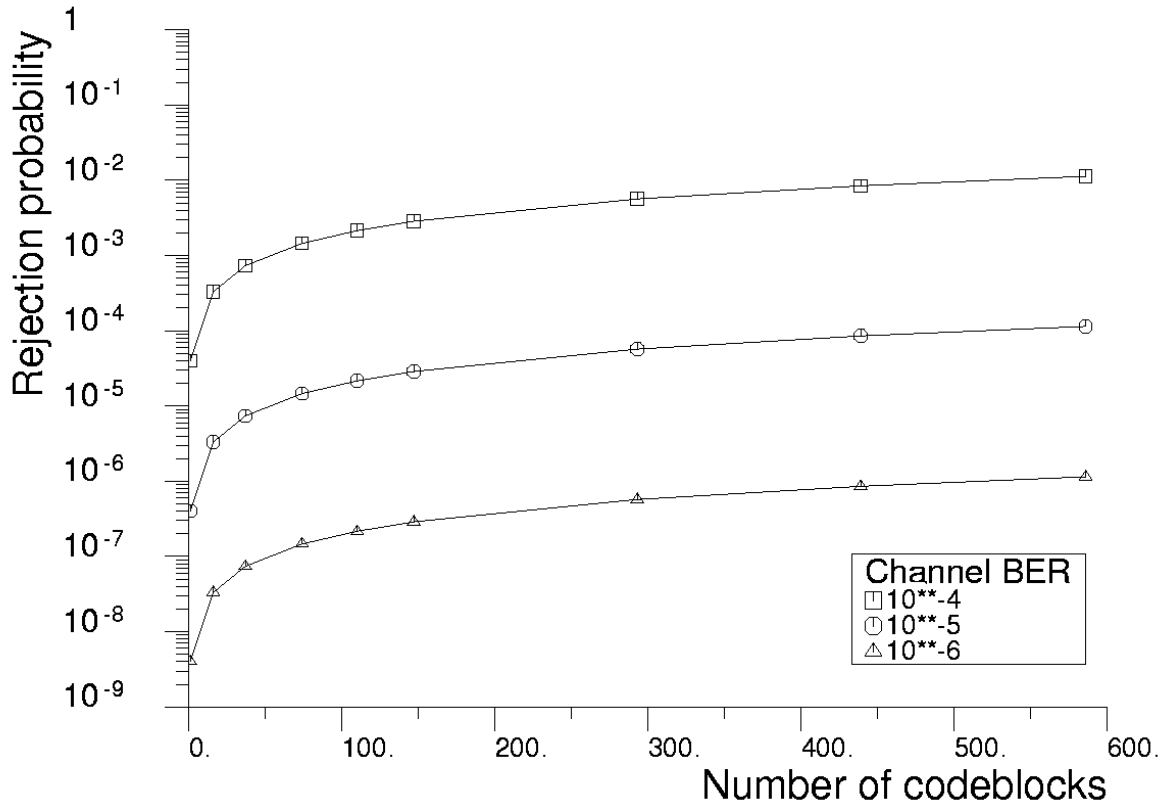
$$P_{F2Y} = P_{TY} + (1 - P_{TY}) ( P_{SY} + (1 - P_{SY}) P_{CY} )$$

Table 8-9 shows the values of the probabilities  $P_{F2X}$  and  $P_{F2Y}$  for the three channel bit error rates. Because the values of  $P_{TX}$  are small compared to  $P_{SX}$  and  $P_{CX}$ , the values of  $P_{F2X}$  in the table are the same as the values of  $P_{FX}$  in table 8-8. Figure 8-3 shows a graph of the  $P_{F2Y}$  values. See figure 8-1 above for a graph of the  $P_{F2X}$  values.

**Table 8-9: Frame Rejection Probabilities,  $P_{F2X}$  and  $P_{F2Y}$ , for the Last or Only Frame in a CLTU in a Sequence of CLTUs (PLOP-2)**

Number of codeblocks in the CLTU	Total frame lengths up to (octets)	$P_{F2X}$ $P_{F2Y}$	Channel Bit Error Rate		
			$10^{-4}$	$10^{-5}$	$10^{-6}$
1	7	$P_{F2X}$ (TED)	$7.87 \times 10^{-3}$	$7.90 \times 10^{-4}$	$7.90 \times 10^{-5}$
		$P_{F2Y}$ (SEC)	$4.01 \times 10^{-5}$	$4.02 \times 10^{-7}$	$4.03 \times 10^{-9}$
16	112	$P_{F2X}$ (TED)	$9.73 \times 10^{-2}$	$1.02 \times 10^{-2}$	$1.02 \times 10^{-3}$
		$P_{F2Y}$ (SEC)	$3.32 \times 10^{-4}$	$3.33 \times 10^{-6}$	$3.33 \times 10^{-8}$
37	259	$P_{F2X}$ (TED)	$2.09 \times 10^{-1}$	$2.32 \times 10^{-2}$	$2.34 \times 10^{-3}$
		$P_{F2Y}$ (SEC)	$7.40 \times 10^{-4}$	$7.43 \times 10^{-6}$	$7.43 \times 10^{-8}$
74	518	$P_{F2X}$ (TED)	$3.74 \times 10^{-1}$	$4.57 \times 10^{-2}$	$4.67 \times 10^{-3}$
		$P_{F2Y}$ (SEC)	$1.46 \times 10^{-3}$	$1.47 \times 10^{-5}$	$1.47 \times 10^{-7}$
110	770	$P_{F2X}$ (TED)	$5.01 \times 10^{-1}$	$6.71 \times 10^{-2}$	$6.92 \times 10^{-3}$
		$P_{F2Y}$ (SEC)	$2.16 \times 10^{-3}$	$2.17 \times 10^{-5}$	$2.17 \times 10^{-7}$
147	1029	$P_{F2X}$ (TED)	$6.05 \times 10^{-1}$	$8.86 \times 10^{-2}$	$9.23 \times 10^{-3}$
		$P_{F2Y}$ (SEC)	$2.88 \times 10^{-3}$	$2.89 \times 10^{-5}$	$2.89 \times 10^{-7}$
293	2051	$P_{F2X}$ (TED)	$8.42 \times 10^{-1}$	$1.69 \times 10^{-1}$	$1.83 \times 10^{-2}$
		$P_{F2Y}$ (SEC)	$5.70 \times 10^{-3}$	$5.74 \times 10^{-5}$	$5.74 \times 10^{-7}$
439	3073	$P_{F2X}$ (TED)	$9.37 \times 10^{-1}$	$2.42 \times 10^{-1}$	$2.73 \times 10^{-2}$
		$P_{F2Y}$ (SEC)	$8.52 \times 10^{-3}$	$8.59 \times 10^{-5}$	$8.59 \times 10^{-7}$
586	4102	$P_{F2X}$ (TED)	$9.75 \times 10^{-1}$	$3.09 \times 10^{-1}$	$3.63 \times 10^{-2}$
		$P_{F2Y}$ (SEC)	$1.14 \times 10^{-2}$	$1.15 \times 10^{-4}$	$1.15 \times 10^{-6}$





**Figure 8-3: Frame Rejection Probability in SEC Mode,  $P_{F2Y}$ , for the Last or Only Frame in a CLTU in a Sequence of CLTUs (PLOP-2)**

Figure 8-3 shows the probability  $P_{F2Y}$  of frame rejection in SEC mode. The values apply for the last or only frame in a CLTU in a sequence of CLTUs (PLOP-2, table 8-9). See figure 8-2 above for the probability in SEC mode in an independent CLTU.

## 8.5 FACTORS AFFECTING FRAME UNDETECTED ERROR RATE

### 8.5.1 GENERAL

The second performance requirement deals with undetected error: it requires that the probability of accepting a frame with an undetected error is smaller than  $10^{-9}$ . Again, the unit of accountability is the frame.

The optional Frame Error Control Field specified in reference [4] can give additional protection against undetected errors. It is a 16-bit field providing a cyclic redundancy check (CRC) over the contents of the frame. At the receiving end, the Frame Error Control Field is validated by the Frame Validation Check Procedure in the Data Link Protocol Sublayer (see 6.6.2.3).

### 8.5.2 SOURCES OF UNDETECTED ERRORS

The use of SEC mode for codeblock decoding reduces the frame rejection rate, but as the following discussion shows, it increases the probability of an undetected error in the frame.

Subsection 8.4.4.3 above describes the different cases that arise during the decoding of BCH codeblocks. Table 8-10 shows the cases which lead to undetected errors in the decoding process. In SEC mode, cases A2 and D2 can both occur. In TED mode, only case A2 can occur.

**Table 8-10: Sources of Undetected Errors**

Case	Description	Modes
A2	The received codeblock has an even number of errors (at least 4) and the errors are not detected.	TED & SEC
D2	The received codeblock has an odd number of errors (at least 3) and the SEC mode correction delivers a codeblock with an even number of errors (at least 2).	SEC

An upper limit for A2 is given by adding the probabilities of 4, 6, 8, ... errors occurring. Similarly, an upper limit for D2 is given by adding the probabilities of 3, 5, 7, ... errors occurring. As table 8-11 shows, the probabilities decrease rapidly as the number of errors increases, so the upper limit for A2 can be considered as the probability of four errors occurring in the codeblock and the upper limit for D2 can be considered as the probability of three errors occurring in the codeblock.

**Table 8-11: Probability of  $n$  Errors Occurring in a Codeblock**

Number of errors, $n$	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
1	$6.26 \times 10^{-3}$	$6.30 \times 10^{-4}$	$6.30 \times 10^{-5}$
2	$1.94 \times 10^{-5}$	$1.95 \times 10^{-7}$	$1.95 \times 10^{-9}$
3	$3.95 \times 10^{-8}$	$3.97 \times 10^{-11}$	$3.97 \times 10^{-14}$
4	$5.92 \times 10^{-11}$	$5.95 \times 10^{-15}$	$5.96 \times 10^{-19}$
5	$6.99 \times 10^{-14}$	$7.02 \times 10^{-19}$	$7.03 \times 10^{-24}$
6	$6.76 \times 10^{-17}$	$6.79 \times 10^{-23}$	$6.79 \times 10^{-29}$
7	$5.50 \times 10^{-20}$	$5.53 \times 10^{-27}$	$5.53 \times 10^{-34}$
8	$3.85 \times 10^{-23}$	$3.87 \times 10^{-31}$	$3.87 \times 10^{-39}$
9	$2.35 \times 10^{-26}$	$2.37 \times 10^{-35}$	$2.37 \times 10^{-44}$

In a simulation of the decoder in figure 6-3, all combinations of single, double, triple and quadruple errors were tested (reference [11]) in SEC mode. The tests demonstrated that the decoder has an error-detection performance which is better than the upper limits. Extending the results to include TED mode, the performance is shown in table 8-12.

**Table 8-12: Error Detection Performance for a Codeblock in SEC and TED Modes**

Number of errors	Combinations	Performance in SEC	Performance in TED
1	63	All corrected	All detected
2	1953	All detected	All detected
3	39711	651 detected	All detected
		39060 undetected*	
4	595665	585900 detected	585900 detected
		9765 undetected	9765 undetected

\*In all these cases, the SEC correction delivers a codeblock with four errors.

### 8.5.3 UNDETECTED ERROR PROBABILITIES WITHOUT CRC

Tables 8-11 and 8-12 show error probabilities and decoding performance for a codeblock. The following two tables show the probabilities of an undetected error in a frame for the case that the optional CRC is not used. Table 8-13 shows the probabilities for TED mode and table 8-14 shows the probabilities for SEC mode.

**Table 8-13: Probability of Undetected Error in a Frame, TED Mode, No CRC**

Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
2	$1.89 \times 10^{-12}$	$1.91 \times 10^{-16}$	$1.91 \times 10^{-20}$
10	$9.47 \times 10^{-12}$	$9.53 \times 10^{-16}$	$9.53 \times 10^{-20}$
20	$1.89 \times 10^{-11}$	$1.91 \times 10^{-15}$	$1.91 \times 10^{-19}$
30	$2.84 \times 10^{-11}$	$2.86 \times 10^{-15}$	$2.86 \times 10^{-19}$
40	$3.79 \times 10^{-11}$	$3.81 \times 10^{-15}$	$3.81 \times 10^{-19}$
50	$4.74 \times 10^{-11}$	$4.76 \times 10^{-15}$	$4.77 \times 10^{-19}$
60	$5.68 \times 10^{-11}$	$5.72 \times 10^{-15}$	$5.72 \times 10^{-19}$
70	$6.63 \times 10^{-11}$	$6.67 \times 10^{-15}$	$6.67 \times 10^{-19}$
80	$7.58 \times 10^{-11}$	$7.62 \times 10^{-15}$	$7.62 \times 10^{-19}$
90	$8.53 \times 10^{-11}$	$8.57 \times 10^{-15}$	$8.58 \times 10^{-19}$
100	$9.47 \times 10^{-11}$	$9.53 \times 10^{-15}$	$9.53 \times 10^{-19}$
110	$1.04 \times 10^{-10}$	$1.05 \times 10^{-14}$	$1.05 \times 10^{-18}$
120	$1.14 \times 10^{-10}$	$1.14 \times 10^{-14}$	$1.14 \times 10^{-18}$
130	$1.23 \times 10^{-10}$	$1.24 \times 10^{-14}$	$1.24 \times 10^{-18}$

**Table 8-14: Probability of Undetected Error in a Frame, SEC Mode, No CRC**

Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
2	$7.75 \times 10^{-8}$	$7.79 \times 10^{-11}$	$7.80 \times 10^{-14}$
10	$3.88 \times 10^{-7}$	$3.90 \times 10^{-10}$	$3.90 \times 10^{-13}$
20	$7.75 \times 10^{-7}$	$7.79 \times 10^{-10}$	$7.80 \times 10^{-13}$
30	$1.16 \times 10^{-6}$	$1.17 \times 10^{-9}$	$1.17 \times 10^{-12}$
40	$1.55 \times 10^{-6}$	$1.56 \times 10^{-9}$	$1.56 \times 10^{-12}$
50	$1.94 \times 10^{-6}$	$1.95 \times 10^{-9}$	$1.95 \times 10^{-12}$
60	$2.33 \times 10^{-6}$	$2.34 \times 10^{-9}$	$2.34 \times 10^{-12}$
70	$2.71 \times 10^{-6}$	$2.73 \times 10^{-9}$	$2.73 \times 10^{-12}$
80	$3.10 \times 10^{-6}$	$3.12 \times 10^{-9}$	$3.12 \times 10^{-12}$
90	$3.49 \times 10^{-6}$	$3.51 \times 10^{-9}$	$3.51 \times 10^{-12}$
100	$3.88 \times 10^{-6}$	$3.90 \times 10^{-9}$	$3.90 \times 10^{-12}$
110	$4.26 \times 10^{-6}$	$4.29 \times 10^{-9}$	$4.29 \times 10^{-12}$
120	$4.65 \times 10^{-6}$	$4.68 \times 10^{-9}$	$4.68 \times 10^{-12}$
130	$5.04 \times 10^{-6}$	$5.07 \times 10^{-9}$	$5.07 \times 10^{-12}$

#### 8.5.4 UNDETECTED ERROR PROBABILITIES WITH CRC

The CRC gives additional protection against undetected errors in a frame. In the tests (reference [11]) the performance of the CRC was analyzed for the cases where the codeblock decoding delivers a codeblock with errors. The CRC performance for a frame with errors in more than one codeblock was also considered.

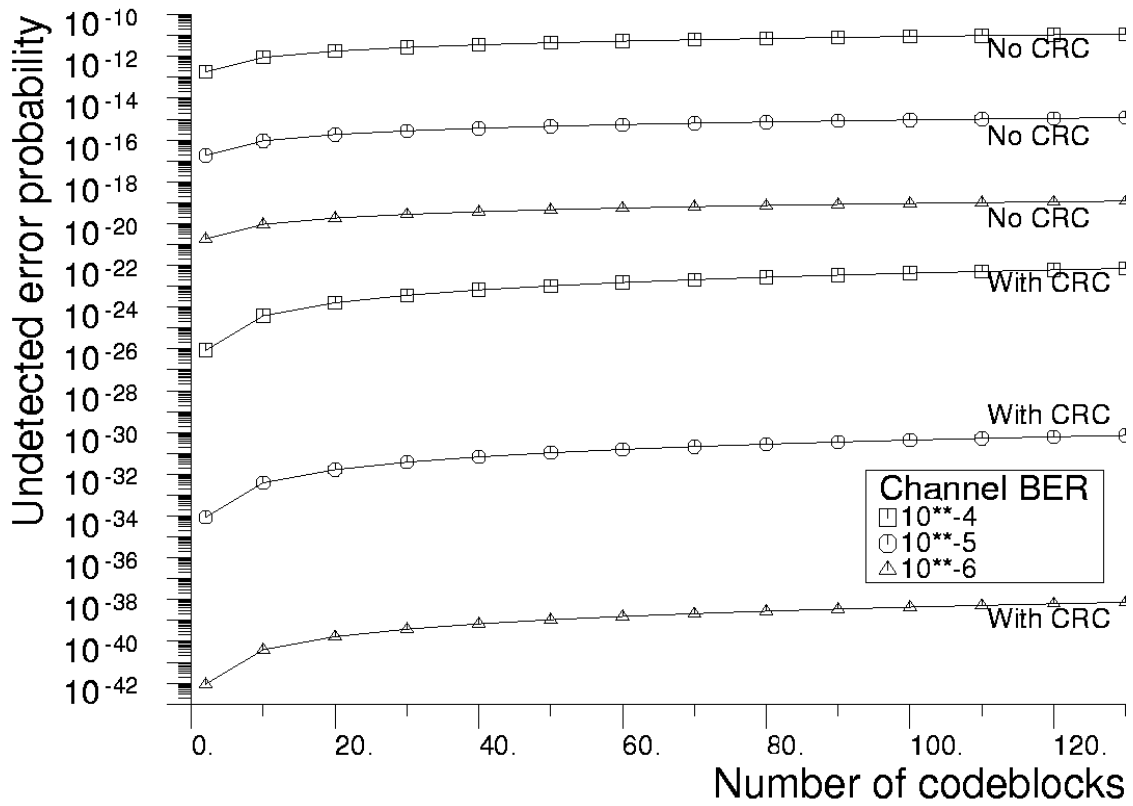
Tables 8-15 and 8-16 show the probabilities of an undetected error in a frame when the CRC is used.

**Table 8-15: Probability of Undetected Error in a Frame, TED Mode, with CRC**

Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
2	$8.77 \times 10^{-26}$	$8.86 \times 10^{-34}$	$8.87 \times 10^{-42}$
10	$3.94 \times 10^{-24}$	$3.99 \times 10^{-32}$	$3.99 \times 10^{-40}$
20	$1.67 \times 10^{-23}$	$1.68 \times 10^{-31}$	$1.69 \times 10^{-39}$
30	$3.81 \times 10^{-23}$	$3.85 \times 10^{-31}$	$3.86 \times 10^{-39}$
40	$6.84 \times 10^{-23}$	$6.91 \times 10^{-31}$	$6.92 \times 10^{-39}$
50	$1.07 \times 10^{-22}$	$1.09 \times 10^{-30}$	$1.09 \times 10^{-38}$
60	$1.55 \times 10^{-22}$	$1.57 \times 10^{-30}$	$1.57 \times 10^{-38}$
70	$2.12 \times 10^{-22}$	$2.14 \times 10^{-30}$	$2.14 \times 10^{-38}$
80	$2.77 \times 10^{-22}$	$2.80 \times 10^{-30}$	$2.80 \times 10^{-38}$
90	$3.51 \times 10^{-22}$	$3.55 \times 10^{-30}$	$3.55 \times 10^{-38}$
100	$4.34 \times 10^{-22}$	$4.39 \times 10^{-30}$	$4.39 \times 10^{-38}$
110	$5.26 \times 10^{-22}$	$5.31 \times 10^{-30}$	$5.32 \times 10^{-38}$
120	$6.26 \times 10^{-22}$	$6.33 \times 10^{-30}$	$6.33 \times 10^{-38}$
130	$7.35 \times 10^{-22}$	$7.43 \times 10^{-30}$	$7.44 \times 10^{-38}$

**Table 8-16: Probability of Undetected Error in a Frame, SEC Mode, with CRC**

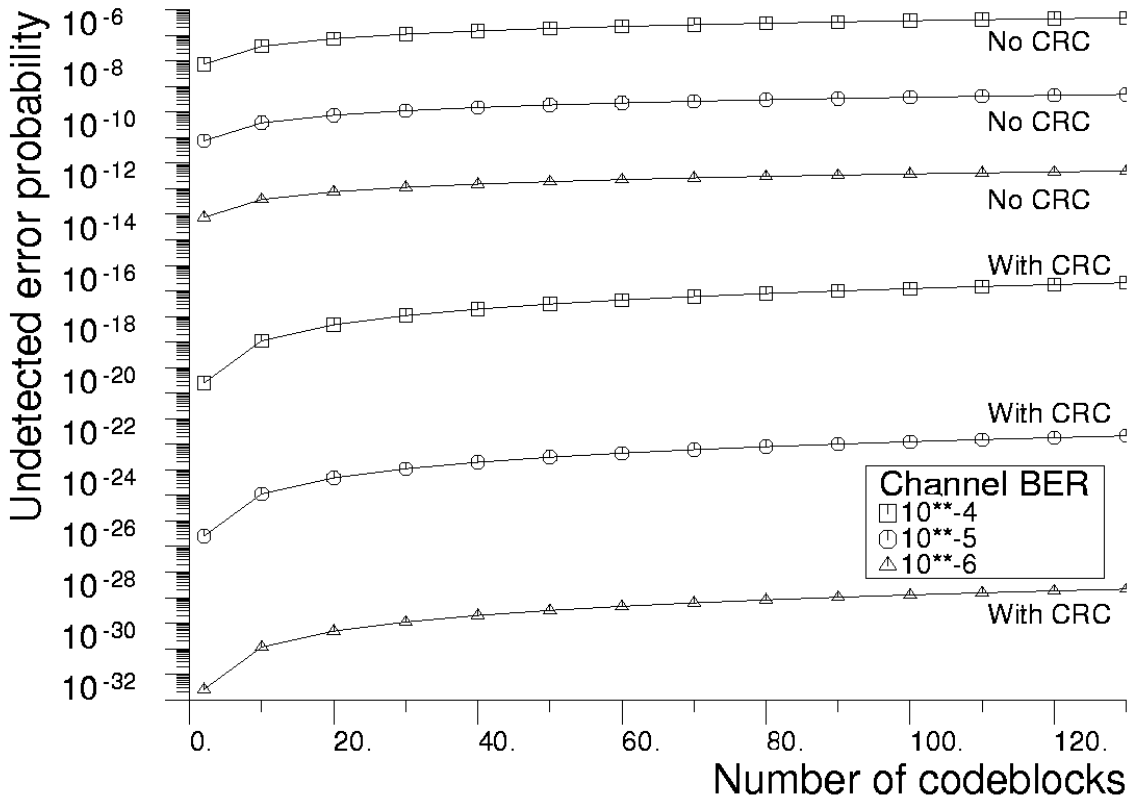
Number of Codeblocks N	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
<b>2</b>	$2.55 \times 10^{-20}$	$2.58 \times 10^{-26}$	$2.58 \times 10^{-32}$
<b>10</b>	$1.15 \times 10^{-18}$	$1.16 \times 10^{-24}$	$1.16 \times 10^{-30}$
<b>20</b>	$4.85 \times 10^{-18}$	$4.91 \times 10^{-24}$	$4.91 \times 10^{-30}$
<b>30</b>	$1.11 \times 10^{-17}$	$1.12 \times 10^{-23}$	$1.12 \times 10^{-29}$
<b>40</b>	$1.99 \times 10^{-17}$	$2.01 \times 10^{-23}$	$2.02 \times 10^{-29}$
<b>50</b>	$3.13 \times 10^{-17}$	$3.16 \times 10^{-23}$	$3.17 \times 10^{-29}$
<b>60</b>	$4.52 \times 10^{-17}$	$4.57 \times 10^{-23}$	$4.58 \times 10^{-29}$
<b>70</b>	$6.17 \times 10^{-17}$	$6.24 \times 10^{-23}$	$6.24 \times 10^{-29}$
<b>80</b>	$8.07 \times 10^{-17}$	$8.16 \times 10^{-23}$	$8.17 \times 10^{-29}$
<b>90</b>	$1.02 \times 10^{-16}$	$1.03 \times 10^{-22}$	$1.04 \times 10^{-28}$
<b>100</b>	$1.26 \times 10^{-16}$	$1.28 \times 10^{-22}$	$1.28 \times 10^{-28}$
<b>110</b>	$1.53 \times 10^{-16}$	$1.55 \times 10^{-22}$	$1.55 \times 10^{-28}$
<b>120</b>	$1.82 \times 10^{-16}$	$1.84 \times 10^{-22}$	$1.85 \times 10^{-28}$
<b>130</b>	$2.14 \times 10^{-16}$	$2.17 \times 10^{-22}$	$2.17 \times 10^{-28}$



**Figure 8-4: Probability of Undetected Error in a Frame in TED Mode**

Figure 8-4 brings together the results of table 8-13 and table 8-15 to show the probabilities of undetected error in a frame in TED mode, with and without the CRC.





**Figure 8-5: Probability of Undetected Error in a Frame in SEC Mode**

Figure 8-5 brings together the results of table 8-14 and table 8-16 to show the probabilities of undetected error in a frame in SEC mode, with and without the CRC.

## ANNEX A

### GLOSSARY

**Acquisition Sequence:** A specific high transition density bit pattern, to enable the receiver to acquire bit synchronization. See 5.2.2.

**Bit Transition Generator:** A generator that produces the random sequence of 255 bits, used in the optional randomization procedure at the sending and receiving ends. See 7.3.

**Carrier Modulation Modes:** Different states of data modulation upon the RF carrier that creates the Physical Channel. See 5.2.4.

**Codeblock:** In this report, a BCH Codeblock as described in 3.3.

**Communications Link Transmission Unit:** A Synchronization and Channel Coding Sublayer data structure consisting of a Start Sequence, a set of BCH Codeblocks, and a Tail Sequence. See 4.3.

**Communications Session:** A continuous period of time during which the signal path is established for the communications channel. See 5.2.1.

**Cyclic Redundancy Check:** The error detection method employed in the optional Frame Error Control Field of a TC Transfer Frame. The Cyclic Redundancy Check (CRC) provides a means to detect errors introduced into the Frame during transmission. At the sending end, the CRC value is calculated from the contents of the Frame and placed in the Frame Error Control Field. At the receiving end, the CRC value is calculated and compared with the value in the Frame Error Control Field. If the values are different, the Frame contains errors. The CRC is used only for error detection, not for error correction.

**Data Link Protocol Sublayer:** The sublayer above the Synchronization and Channel Coding Sublayer. At the sending end, it delivers TC Transfer Frames to the Synchronization and Channel Coding Sublayer. At the receiving end, the Synchronization and Channel Coding Sublayer delivers candidate TC Transfer Frames to the Data Link Protocol Sublayer.

**Decoding:** The processing of a received BCH Codeblock, through which the information and parity bits of the codeblock are used to detect or correct errors. The BCH decoding has two modes of operation: error detection (see 6.5.2) and error correction (see 6.5.3). The data contents of successfully decoded BCH Codeblocks are transferred to the Data Link Protocol Sublayer.

**Encoding:** The processing of the information bits in a BCH Codeblock at the sending end to generate the parity bits. See section 3.

**Event:** In this report, an occurrence that causes the CLTU Reception Procedure to perform actions and change states. See 6.2.

**Fill bits:** Bits used to complete an integral number of BCH Codeblocks. Because the data received from the Data Link Protocol Sublayer may not fit exactly in an integral number of BCH Codeblocks, the Information field of the last BCH Codeblock in a CLTU may contain fill bits. An octet of fill bits has the hexadecimal value 55. The fill bits are transmitted as part of the CLTU. At the receiving end, the fill bits are detected and removed by the Data Link Protocol Sublayer. See 3.4 and 6.6.2.2.

**Filler Bit:** The last bit of the last octet of a BCH Codeblock. The Filler Bit is always zero. See 3.3.

**Frame:** In this report, a TC Transfer Frame.

**Idle Sequence:** A specific high transition density bit pattern transmitted during a communications session in the absence of a CLTU, to enable the receiver to maintain bit synchronization. See 5.2.3.

**Octet:** A contiguous string of eight bits; an eight-bit word.

**Physical Channel:** A stream of bits transferred over a space link in a single direction.

**Physical Layer:** The layer below the Synchronization and Channel Coding Sublayer, which provides the RF channel. At the sending end it provides the radio frequency and modulation techniques required to create and operate the channel. At the receiving end it provides the reception, demodulation, and symbol synchronization for the channel.

**Physical Layer Operations Procedure:** A specific procedure of the Physical Layer designed to activate and deactivate the physical communications channel by invoking RF carrier and modulation techniques. See section 5.

**Pseudo-randomization:** A bandwidth-efficient technique of algorithmically translating the data bits to ensure frequent bit transitions in the communications channel, also called randomization in this report. No additional bits are added by this process.

**Randomization:** Pseudo-randomization. For brevity, ‘randomization’ is used in place of ‘pseudo-randomization’ in this report.

**Reliable:** Meets the quality, quantity, continuity and completeness criteria specified by the communications system.

**Start Sequence:** A specific bit pattern that delimits the start of the first BCH Codeblock and, if necessary, resolves the sense of a ‘1’ and ‘0’ in the CLTU. See 4.4.

**Symbol:** A bit in an encoded data stream.

**Synchronization and Channel Coding Sublayer:** The sublayer responsible for 1) the provision of data structures to enable bit synchronization to be achieved and maintained and 2) the coding of Frames to provide a forward error detection and correction capability. The sublayer is the subject of this report and of the Recommended Standard (reference [2]).

**TC Transfer Frame:** The data unit generated by the Data Link Protocol Sublayer of the TC Space Data Link Protocol. The TC Transfer Frame is specified in reference [4].

**Tail Sequence:** A specific data pattern that delimits the end of a CLTU. The Tail Sequence field has the same length as a BCH Codeblock, but its contents differ from a valid or correctable BCH Codeblock.

**ANNEX B****ACRONYMS AND ABBREVIATIONS**

AOS	Advanced Orbiting Systems
BCH	Bose-Chaudhuri-Hocquenghem
BR	Buffer Register
BTG	Bit Transition Generator
CCSDS	Consultative Committee for Space Data Systems
CLCW	Communications Link Control Word
CLTU	Communications Link Transmission Unit
CMM	Carrier Modulation Mode
CR	Codeblock Rejection
CRC	Cyclic Redundancy Check
EOD	Even/Odd Detector
ESA	European Space Agency
OSI	Open Systems Interconnection
PAR	Parity indicator
PLOP	Physical Layer Operations Procedure
PLR	Position Location Register
R-S	Reed-Solomon
RF	Radio Frequency
SB	Syndrome Binary
SEC	Single Error Correction
SR	Syndrome Register
SYND	Syndrome value
TED	Triple Error Detection
TC	Telecommand
TM	Telemetry

## ANNEX C

### THEORETICAL BACKGROUND OF THE RANDOMIZATION SEQUENCE

NOTE – This annex contains an edited version of *Selection of New Randomizer* by Larsen, Justesen, and Paaske, January 1994. The report was produced for ESA and was an input for the CCSDS Panel 1A meeting in July 1994.

#### C1 INTRODUCTION

The generator polynomial for the randomizer specified in reference [2] for TC channel coding is the one recommended in the conclusion (C4) of this annex. Because a Reed-Solomon (R-S) code may be chosen in the future for TC channel coding, it was decided to choose a randomizer sequence which avoids interaction with the R-S codes.

The problem addressed in this annex comes from the discovery that the randomizer sequence specified for TM channel coding in reference [5] is also a codeword of the standard (255,223) Reed-Solomon code in reference [5]. In C2, the reasons for this are explained. Since this is an unfortunate choice of the randomizer sequence, C3 describes a search for randomizer sequences that:

- are not codewords in the standard R-S (255,223) code in reference [5] at any interleaving depth;
- are not codewords in the R-S (255,239) code in reference [10] at any interleaving depth;
- have large distances from the two synchronization markers in reference [5] and their inverses.

In C4, a randomizer sequence fulfilling all these requirements is given. The proposed sequence has the property that it is marked as not decodable by a proper decoder, e.g., the decoder in reference [10], used at any interleaving depth in the range one to eight.

#### C2 THEORY OF RANDOM SEQUENCES AND CODES

The theory of this section is mostly found in reference [12] and is applied here to the case of binary pseudo-random sequences as used in reference [5]. A pseudo-random binary sequence:

$$x_0x_1x_2\dots x_t\dots$$

is generated by the linear recursion:

$$x_t = r_{m-1}x_{t-1} + r_{m-2}x_{t-2} + \dots + r_1x_{t-m+1} + r_0x_{t-m}$$

where  $r_j$  are binary coefficients. The properties of the sequence depend on the characteristic polynomial (or randomizer polynomial):

$$R(z) = z^m + r_{m-1}z^{m-1} + \dots + r_1z + r_0$$

The sequence has period  $2^m-1$  if and only if  $R(z)$  has degree  $m$ , is irreducible and primitive when calculations are done in the binary field  $GF(2)$ . The sequences generated by such a polynomial of degree  $m$  all have the same distribution of runs of '0's and '1's. The run distribution for sequences of period  $2^m-1$  is described in the table below

**Table C-1: Run Distribution for Sequences of Period  $2^m-1$**

Length	0-runs	1-runs
1	$2^{m-3}$	$2^{m-3}$
2	$2^{m-4}$	$2^{m-4}$
$r$	$2^{m-r-2}$	$2^{m-r-2}$
$m-2$	1	1
$m-1$	1	0
$m$	0	1

A consequence of the characteristic 2 of this field is that the recursion with the characteristic polynomial

$$R_p(z) = z^{m2^p} + r_{m-1}z^{(m-1)2^p} + \dots + r_1z^{2^p} + r_0$$

is also fulfilled by the sequence  $x_0x_1\dots$ . If the sequence is divided into groups of  $2^p$  bits, and these groups are considered as elements of the field  $GF(2^p)$  represented in some basis, usage of  $R_p(z)$  on the corresponding bits in each group gives a recursion on the elements in  $GF(2^p)$ :

$$R(z) = z^m + r_{m-1}z^{m-1} + \dots + r_1z + r_0$$

Calculations are now done in  $GF(2^p)$  and the sequence has the period  $2^m-1$  if and only if  $R(z)$  is irreducible over  $GF(2)$  and is a primitive polynomial. If the sequence is interpreted as a polynomial:

$$X(z) = x_0 + x_1z + x_2z^2 + \dots + x_{2^m-2}z^{2^m-2}$$

the implication of the recursion is that  $R(z)$  is a parity check polynomial for a cyclic code containing  $X(z)$ . The length of this code is  $2^m-1$  and the generator polynomial is

$(z^{2^m}-1)/R(z)$ . When  $X(z)$  is evaluated at a root of the generator polynomial  $z = \sigma$ , we get  $X(\sigma) = 0$  (a syndrome for the code). We notice that the roots of the generator polynomial are all roots of  $z^{2^m}-1$  that are not roots in  $R(z)$ .

Consider now the relations between a Reed-Solomon code over  $GF(2^p)$  with generator  $G(z)$  and length  $2^p-1$  and random sequences  $X(z)$  with period  $2^p-1$ . If the roots of  $G(z)$  are denoted  $\sigma_{RS}$ , we get  $X(\sigma_{RS}) = 0$  if and only if all these roots  $\sigma_{RS}$  are distinct from the roots of  $R(z)$  since in that case all roots of  $G(z)$  are roots of the generator polynomial for the code that  $X(z)$  belongs to.

Example: Among the 16 different polynomials with degree 8 that are irreducible and primitive is  $z^8 + z^7 + z^5 + z^3 + 1$ , which is used in reference [5] as randomizer polynomial for TM channel coding. The field  $GF(2^8)$  is defined by the polynomial  $z^8 + z^6 + z^4 + z^3 + z^2 + z + 1$ , and  $b$  is the primitive root of this polynomial. The roots of  $R(z)$  are  $b^{37}, b^{74}, b^{148}, b^{41}, b^{82}, b^{164}, b^{73}$  and  $b^{146}$ , and they are all outside the roots of the R-S (255,223) code:  $b^{112}, \dots, b^{143}$ . Thus the random sequence is a codeword in the R-S code.

If  $c, c^2, c^4, c^8, \dots$  are the roots of  $R(z)$ , the sequence is a codeword of the R-S code that is interleaved with interleaving depth  $I$ , if and only if  $c^I, c^{2I}, c^{4I}, c^{8I}, \dots$  are not roots of the generator polynomial for the R-S code. Since the characteristic of the field is two, the properties for the sequence are identical for all interleaving depths  $I=2^p$ , i.e., 1, 2, 4, 8, 16. The other values of  $I$  must be checked separately.

### C3 SEARCH FOR RANDOMIZER POLYNOMIALS

The search described in this section aims at fulfilling the following properties for the pseudo-random sequence with period 255 generated by the randomizer polynomial  $R(z)$  of degree 8:

- The sequence is not a codeword in the standard (255,223) code at any interleaving depth, as explained in C2 above. The distance to codewords should be large and preferably should be greater than 16.
- The sequence is not a codeword (at any interleaving depth) in the (255,239) code defined in reference [10], where the generator polynomial has roots  $b^{120}, b^{113}, \dots, b^{135}$ , and  $b$  is defined in the example in C2 above.
- The sequence has a large minimum distance from the two synchronization markers, Sync1: 1A CF FC 1D (hex), Sync2: 35 2E F8 53 (hex), and their inverses.

The results for all 16 possible randomizer polynomials giving period 255 are shown in the table below.



**Table C-2: Results for the 16 Randomizer Polynomials**

<b>R(z) (hex)</b>	<b>Interleaving depths for which the sequence is a codeword</b>		<b>Minimum distance to</b>			
	<b>in code (255,223)</b>	<b>in code (255,239)</b>	<b>Sync1</b>	<b>Sync1 inverse</b>	<b>Sync2</b>	<b>Sync2 inverse</b>
171	None	3,5,6,7	9	7	7	7
1A9	1,2,4,8	1,2,4,5,8	8	9	8	8
169	None	1,2,3,4,6,8	8	6	7	9
165	7	5,7	7	9	6	9
1F5	None	None	7	8	9	8
18D	7	3,6,7	8	5	9	6
1C3	None	1,2,3,4,5,6,8	9	8	9	8
1E7	1,2,4,8	1,2,4,7,8	8	8	8	8
11D	None	3,5,6,7	9	8	8	9
12B	1,2,4,8	1,2,4,5,8	8	8	6	10
12D	None	1,2,3,4,6,8	8	8	9	9
14D	7	5,7	9	9	9	8
15F	None	None	8	8	8	9
163	7	3,6,7	8	8	9	7
187	None	1,2,3,4,5,6,8	8	9	9	8
1CF	1,2,4,5	1,2,4,7,8	8	8	10	7

The polynomial 1A9 (hex) is the one used in reference [5] as randomizer polynomial for TM channel coding. From the table it is seen that 15F (hex) is the better choice of the two polynomials that do not generate codewords in any of the two codes at any interleaving depth, since it has the best distance properties with respect to the synchronization markers. The sequence generated has distances to codewords that are so large that it is not decoded at any interleaving depth in the range 1 to 8.

#### C4 CONCLUSION

The randomizer polynomial should be selected as:

$$z^8 + z^6 + z^4 + z^3 + z^2 + z + 1$$

which is identical to the polynomial used for generating the field.

The first 40 binary symbols generated by this polynomial are

1111 1111 0011 1001 1001 1110 0101 1010 0110 1000

or in hexadecimal notation

FF 39 9E 5A 68

The sequence generated has a minimum distance to the synchronization markers and their inverses of at least 8. The sequence thus generated gives non-zero syndromes for both R-S codes for all interleaving depths in the range 1 to 8, and the sequence is not decoded by a proper working decoder at any of these interleaving depths.

## ANNEX D

### THEORETICAL BACKGROUND OF THE CLTU TAIL SEQUENCE

NOTE – This annex contains an edited version of *Uncorrectable Sequences and Telecommand* by Ekroot, McEliece, Dolinar, and Swanson, May 15, 1993. The original report was an input for the CCSDS Panel 1A meeting in September 1993 and contributed to the selection of the current CLTU Tail Sequence described in 4.5.1. At the time the report was written, the CCSDS-recommended CLTU Tail Sequence was the one described in 4.5.3, and it had multiple lengths as described in 4.5.2.

#### D1 ABSTRACT

The purpose of a tail sequence for command link transmission units is to fail to decode, so that the command decoder will begin searching for the start of the next unit. A tail sequence used by several missions and recommended for this purpose by CCSDS is analyzed. A single channel error can cause the sequence to decode. An alternative sequence requiring at least two channel errors before it can possibly decode is presented. (No sequence requiring more than two channel errors before it can possibly decode exists for this code.)

#### D2 INTRODUCTION

When a command link transmission unit (CLTU) consisting of many codeblocks is received by a spacecraft, the command decoder verifies either that each codeblock is a valid codeword and accepts it, or that it is a slightly corrupted codeword and corrects it, or that it is too far from a valid codeword and rejects it. Rejecting a codeblock causes the receiver to give up on the unit and begin searching for the start of the next unit. At the end of the CLTU, there is a tail sequence designed to be rejected as a codeword, sending the decoder into a 'search mode'. This report analyzes the performance of the tail sequence recommended by CCSDS and used by several missions. So instead of the usual question about a code, i.e., how many errors can the code correct or detect, the question here is how many errors can occur before an uncorrectable sequence becomes correctable.

#### D3 ANALYSIS OF UNCORRECTABLE SEQUENCES

##### D3.1 GENERAL

In order for a sequence to be uncorrectable, it must be far enough from a codeword to cause the decoder not to decode. At the very least, it must differ from the nearest codeword in more positions than the decoder is able to correct. However, channel errors can make such a sequence decodable. The more errors that must occur before the sequence becomes correctable, the less likely it is that the sequence will accidentally decode. In order to

maximize the number of channel errors before the sequence will decode, it is necessary to characterize and find sequences that are as far away from codewords as possible.

A code is designed to have codewords that are maximally far away from each other. If, as a simple example, a new code is created by using a subset of the codewords, then the unused codewords are still far away from the codewords of the new code. Intuitively, these unused codewords are candidates for uncorrectable sequences.

The codes discussed in this article include the perfect (63,57) Hamming code, the (63,56) expurgated Hamming code, and shortened versions of the expurgated code. The codewords for the shortened codes correspond to subsets of the codewords of the (63,56) code, which are themselves the even codewords from the (63,57) code. By making use of the larger code's properties and keeping track of what happens as the codeword sets get smaller, sequences that are uncorrectable in each of the smaller codes can be found.

### D3.2 THE PERFECT (63,57) HAMMING CODE

The generator polynomial for the (63,57) code is the sixth-degree primitive polynomial:

$$g_p(x) = x^6 + x + 1 \quad (\text{D1})$$

The (63,57) code is perfect and has minimum distance 3, i.e., every binary sequence of length 63 either is a codeword or is Hamming distance one away from exactly one codeword. For a perfect code, there are no holes left when the space of binary sequences is filled with spheres of radius one centered at the codewords.

The even-weight words are a promising subset to use for a new code, leaving the odd-weight words as candidates for uncorrectable sequences. In fact the (63,56) code described in the next section uses the even-weight words.

### D3.3 THE (63,56) EXPURGATED HAMMING CODE

The generator polynomial for the (63,56) code is given by:

$$g(x) = x^7 + x^6 + x^2 + 1 = (x + 1)(x^6 + x + 1) \quad (\text{D2})$$

Since the generator polynomial is the product of  $x + 1$  and the generator polynomial  $g_p(x)$  of the perfect (63,57) code, the (63,56) code consists of only the even-weight codewords from the perfect (63,57) code. This code has minimum distance four and can correct at most one error.

Note that any odd-weight codeword in the (63,57) code is exactly Hamming distance three away from the nearest even-weight codewords, and the even-weight codewords are the codewords of the (63,56) code. Any odd-weight binary sequence of length 63 differs from a

nearest codeword by either three bits or one bit. Similarly, any even-weight sequence either is a codeword or is Hamming distance two from the nearest codewords.

The following example illustrates how the concepts of distance relate to a sequence that is not a codeword in the perfect (63,57) code. The sequence selected for the example plays a role in the CCSDS Recommended Standards for telecommand; the description of that role is deferred to annex subsection D4.

**Example 1:** The length 63 sequence:

01 1010101

has even weight, and is therefore either a codeword in the (63,56) code or two away from a codeword. The parts of the two-part *syndrome*  $(s_1(\mathbf{r}), s_2(\mathbf{r}))$  of a sequence  $\mathbf{r} = r_{N-1}r_{N-2}\dots r_1r_0$  are given by:

$$s_1(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha^6 + \alpha + 1} \quad (\text{D3})$$

$$s_2(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha + 1} \quad (\text{D4})$$

where  $\alpha$  is a root of the generator polynomial  $g(x)$ , and  $N$  is the length of the codewords. The two-part syndrome tells either if the sequence is a codeword, how to correct it if it is distance one from a codeword, or that it is not near enough to a single codeword. Specifically, if the syndrome is (0,0), then the sequence is a codeword; if the syndrome is (0,1), then the sequence has odd weight and differs from the nearest codeword in three positions; if the syndrome is  $(\alpha^j, 1)$ , then the sequence has odd weight and differs from the nearest codeword in the  $j$ th position; if the syndrome is  $(\alpha^j, 0)$ , then the weight is even and the sequence differs from the nearest codewords in two positions.

For this example sequence, the first part of the syndrome (modulo  $\alpha^6 + \alpha + 1$ ) is:

$$s_1(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha^6 + \alpha + 1} \quad (\text{D5})$$

$$= 1 + \alpha^2 + \alpha^4 + \alpha^6 + \alpha^7 + \alpha^9 + \alpha^{11} + \dots + \alpha^{61} \quad (\text{D6})$$

$$= 1 + \alpha^2 + \alpha^4 + (\alpha + 1) + \frac{\alpha^7 + \alpha^{63}}{1 + \alpha^2} \quad (\text{D7})$$

$$= \alpha + \alpha^2 + \alpha^4 + (\alpha + 1) + \frac{\alpha\alpha^6 + 1}{1 + \alpha^2} \quad (\text{D8})$$

$$= \frac{(\alpha + \alpha^2 + \alpha^4)(1 + \alpha^2)}{1 + \alpha^2} + \frac{\alpha(\alpha + 1) + 1}{1 + \alpha^2} \quad (\text{D9})$$

$$= \frac{\alpha + \alpha^2 + \alpha^4 + \alpha^3 + \alpha^4 + \alpha^6 + \alpha^2 + \alpha + 1}{1 + \alpha^2} \quad (\text{D10})$$

$$= \frac{\alpha^3 + \alpha^6 + 1}{1 + \alpha^2} \quad (\text{D11})$$

$$= \frac{\alpha^3 + (1 + \alpha) + 1}{1 + \alpha^2} \quad (\text{D12})$$

$$= \frac{\alpha^3 + \alpha}{1 + \alpha^2} \quad (\text{D13})$$

$$= \alpha \quad (\text{D14})$$

where equations D7, D9, and D12 follow from equivalence modulo  $\alpha^6 + \alpha + 1$ . The second part of the syndrome (modulo  $\alpha + 1$ ) is:

$$s_2(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha + 1} \quad (\text{D15})$$

$$= \text{weight}(\mathbf{r}) \pmod{2} \quad (\text{D16})$$

$$= 32 \pmod{2} = 0 \quad (\text{D17})$$

The syndrome  $(\alpha, 0)$  indicates that the sequence is not a codeword in the (63,56) code, and is two away from the nearest codewords. A single error in any bit except  $r_1$  will make the sequence differ from a codeword by one and be thus decodable. (If the error is in  $r_1$ , the syndrome becomes  $(0, 1)$ , indicating that it is three away from a codeword.)

If instead a sequence is considered that is an odd-weight word in the perfect (63,57) code, it is three away from the codewords of the (63,56) code. This means that two errors must occur before it becomes decodable by a single-error-correcting (63,56) decoder. Such a sequence would be a better choice for a tail sequence because it is more resistant to accidental decoding in the presence of errors.

In selecting a particular sequence for the command coding application, the effects on the distance and uncorrectability properties as the code is shortened must be taken into account.

### D3.4 SHORTENING THE (63,56) CODE

Select a subset of a code, where all the codewords in the subset have zeros in some specified positions. Since all of the codewords in the subset have zeros in the specified positions, those positions carry no information and can be ignored. The resulting set of codewords forms a *shortened code*. Shortening cannot decrease the minimum distance, and will only increase the minimum distance if the code is shortened severely. For the (63,56) code, it can be shown that as long as the shortened length is greater than 32, the minimum distance will remain 4, and the maximum distance of any sequence to the nearest codewords will remain 3.

For this application, shortening will be done by taking only the codewords which have zeros in the leftmost or first  $m$  positions. Note that for each shortened word there is a corresponding full-length word that has zeros in the first  $m$  positions.

**Example 2:** Consider the sequence of length 55:

01 1010101

It corresponds to the full-length sequence:

0000000001 1010101

with weight 28.

The first part of the syndrome for the full-length sequence is:

$$s_1(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha^6 + \alpha + 1} \quad (\text{D18})$$

$$= 1 + \alpha^2 + \alpha^4 + \alpha^6 + \alpha^7 + \alpha^9 + \alpha^{11} + \dots + \alpha^{53} \quad (\text{D19})$$

$$= 1 + \alpha^2 + \alpha^4 + (1 + \alpha) + (\alpha^7 + \alpha^9 + \alpha^{11} + \dots + \alpha^{53}) \quad (\text{D20})$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha^7 + \alpha^{55}}{1 + \alpha^2} \quad (\text{D21})$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha(\alpha + 1) + \alpha(\alpha + 1)^9}{(1 + \alpha)^2} \quad (\text{D22})$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha(1 + (\alpha + 1)^8)}{1 + \alpha} \quad (\text{D23})$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha(1 + \alpha^8 + 1)}{1 + \alpha} \quad (\text{D24})$$

$$= \alpha + \alpha^2 + \alpha^4 + \frac{\alpha\alpha^2(\alpha+1)}{1+\alpha} \quad (\text{D25})$$

$$= \alpha + \alpha^2 + \alpha^4 + \alpha^3 \quad (\text{D26})$$

$$= \alpha(\alpha+1)^3 \quad (\text{D27})$$

$$= \alpha^{19} \quad (\text{D28})$$

The second part of the syndrome is zero since the weight is even. The two-part syndrome  $(\alpha^{19}, 0)$  indicates that the sequence is not a codeword, and that it is two away from the nearest codewords. A single error can make the sequence differ from a codeword by one and be thus decodable.

### D3.5 FINDING A GOOD UNCORRECTABLE SEQUENCE

The concepts in D3.3 and D3.4 lead to the definition of a *good uncorrectable sequence* as one for which it and all the desired truncations of it are maximally distant from the codewords in the corresponding code, i.e., an odd-weight codeword in the perfect (63,57) code. The syndrome for good uncorrectable sequences is (0,1). The analysis below shows that an uncorrectable sequence can be chosen so that, when it is truncated by octets, its syndrome does not change, and therefore it does not become correctable.

If a given sequence is truncated by  $m$  bits, and if it is desired that the syndrome not be changed by the truncation, then the smallest possible  $m$  is eight. This is because in order not to change either part of the syndrome, the truncated bits must correspond to a polynomial that is zero modulo both  $\alpha + 1$  and  $\alpha^6 + \alpha + 1$ . The lowest order nonzero polynomial satisfying that requirement is  $g(\alpha) = \alpha^7 + \alpha^6 + \alpha^2 + 1$ . Therefore the shortest such nonzero sequence is 11000101, which has length eight.

There are engineering reasons for truncating by octets in the application considered in annex subsection D4. Also, bit synchronization requirements often make it preferable to have many transitions in the sequence, and thus a mostly zeros sequence is undesirable. For the remainder of the report, it is assumed that shortening will be done only by multiples of eight bits, and that octets of zeros are not of interest.

Since the second part of the syndrome must be 1 for the sequence and all of its truncations, and the truncated bits 11000101 have even weight, the nontruncated part of the sequence must have odd weight. Since the first part of the syndrome must be 0 for the sequence and all of its truncations, the nontruncated part of the sequence must correspond to a polynomial which is zero modulo  $\alpha^6 + \alpha + 1$ . The shortest such sequence is 1000011.

A simple construction of a good uncorrectable sequence is a concatenation of octets of the form 11000101 with the seven bits 1000011 in the rightmost positions. This is not the only



good uncorrectable sequence, but it does have good distance and bit synchronization properties. The syndrome for this sequence is confirmed in the next example.

**Example 3:** Consider:

11000101 11000101 11000101 11000101 11000101 11000101 11000101 1000011

This sequence has weight 31, so the second part of the syndrome  $s_2(\mathbf{r})$  is 1. The first part of the syndrome (modulo  $\alpha^6 + \alpha + 1$ ) for the full-length sequence is

$$s_1(\mathbf{r}) = \sum_{i=0}^{N-1} r_i \alpha^i \pmod{\alpha^6 + \alpha + 1} \quad (\text{D29})$$

$$= 1 + \alpha^2 + \alpha^6 + \sum_{k=0}^6 \alpha^{7+8k} (1 + \alpha^2 + \alpha^6 + \alpha^7) \quad (\text{D30})$$

$$= 1 + \alpha^2 + \alpha^6 + \sum_{k=0}^6 \alpha^{7+8k} (1 + \alpha + \alpha^6)(1 + \alpha) \quad (\text{D31})$$

$$= 0 \quad (\text{D32})$$

Thus, the sequence is three away from the nearest codewords. A single error cannot make this sequence decodable.

## D4 COMMAND LINK CODING

### D4.1 GENERAL

The CCSDS Recommended Standard uses a tail sequence that is specially constructed not to decode. By not decoding, it causes the receiver to begin searching for the next CLTU. It will be shown that the sequence that has been recommended is not the best sequence in terms of distance, and a better one will be given.

In order to apply the results of annex subsection D3 to the command coding problem, the operations of the CLTU must be detailed.

### D4.2 TELECOMMAND CODEBLOCK

The telecommand codeblock has  $K$  information bits, 7 *inverted* parity check bits, and a fill bit for a total length of  $L$ . Because of the code selected and the desire to shorten in units of 8 bits, the number of information bits and block lengths considered are  $K = 32, 40, 48,$  and  $56,$  and  $L = 40, 48, 56,$  and  $64,$  respectively.

### D4.3 COMMAND LINK TRANSMISSION UNIT

The CLTU consists of

- a 16-bit start sequence, namely 1110101110010000;
- a number of telecommand codeblocks (the information may be padded with fill to make the information into a multiple of  $K$  bits);
- a tail sequence which is a sequence of bits the same length as a codeblock and designed to be uncorrectable. The idea is to cause the receiver to stop decoding and begin looking for the start sequence of the next CLTU.

### D4.4 TAIL SEQUENCES

The CCSDS Recommended Standard specifies that the tail sequence  $\mathbf{t}$  be a sequence of alternating zeros and ones beginning with a zero. Ignoring the fill bit, and noting that the parity bits are inverted, it can be seen that for a block length  $L$  of 64, this corresponds to the sequence in Example 1. As illustrated in the example, this sequence has distance properties such that a single channel error can, and almost certainly will, make the sequence decodable.

For the shorter block lengths  $L = 40, 48,$  and  $56$ , the sequences of alternating zeros and ones beginning with a zero have corresponding full-length sequences with zeros filled in the first positions. All three of these corresponding sequences have even weights, and have nonzero syndromes. The calculation for  $L = 56$  is done in Example 2.

If instead the sequence:

```
11000101 11000101 11000101 11000101 11000101 11000101 11000101 0111100 0
```

is used as the tail sequence, it corresponds (when the fill bit is removed and the parity bits are inverted) to the sequence in Example 3. If one channel error occurs, then this sequence will still be uncorrectable.

### D4.5 AUGMENTATION USING THE FILL BIT

The CCSDS Recommended Standard proposes to use the fill bit as a flag to tell the decoder to operate in error-detect mode only. This improves the probability of spotting the tail sequence by not allowing the decoder to correct any errors when the fill bit is 1. In this mode, two errors are sufficient to make the CCSDS tail sequence (of alternating zeros and ones) decodable, while a minimum of three errors is required to make the sequence presented here decodable.

## D5 CONCLUSIONS

The analysis in this report shows that there are uncorrectable sequences that can tolerate one more channel error than the CCSDS tail sequence before becoming decodable. It is also

shown that this property may be preserved for the shortened as well as the full-length codes recommended by CCSDS. A sequence satisfying these requirements should be considered for the role of the CCSDS tail sequence since it is more resistant to channel errors than the proposed tail sequence, and still has many transitions to aid bit synchronization.

## ANNEX E

### PERFORMANCE OF OBSOLETE FEATURES

#### E1 INTRODUCTION

This annex contains performance data for obsolete features of TC channel coding. The features were included in earlier versions of CCSDS Recommendations or were covered in CCSDS Reports. These features are not included in the current issue of the Recommended Standard (reference [2]), but their performance data are preserved here for comparative and historical reference. The features covered by this annex are:

- shorter lengths for BCH Codeblock (see 3.3.2);
- earlier pattern for Tail Sequence (see 4.5.3);
- use of two Tail Sequences with the earlier pattern;
- Filler Bit Augmentation algorithm (see 6.5.3.3).

#### E2 CODEBLOCK REJECTION

In TED mode, the probability  $P_{CX}$  of one of the BCH Codeblocks in a CLTU causing a codeblock rejection is:

$$P_{CX} = 1 - [(1 - p)^n]^N$$

where  $n$  is one less than the codeblock length and  $N$  is the number of codeblocks in the CLTU. For a given frame length, the number of codeblocks required to hold the frame increases as the codeblock size decreases. Tables E-1 to E-4 show the values of the probability  $P_{CX}$  for the three channel bit error rates for different frame lengths.

**Table E-1: Probability  $P_{CX}$  of Codeblock Rejection in TED Mode,  
Frame Length 20 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
<b>64</b>	<b>3</b>	$1.87 \times 10^{-2}$	$1.89 \times 10^{-3}$	$1.89 \times 10^{-4}$
<b>56</b>	<b>4</b>	$2.18 \times 10^{-2}$	$2.20 \times 10^{-3}$	$2.20 \times 10^{-4}$
<b>48</b>	<b>4</b>	$1.86 \times 10^{-2}$	$1.88 \times 10^{-3}$	$1.88 \times 10^{-4}$
<b>40</b>	<b>5</b>	$1.93 \times 10^{-2}$	$1.95 \times 10^{-3}$	$1.95 \times 10^{-4}$

**Table E-2: Probability  $P_{CX}$  of Codeblock Rejection in TED Mode, Frame Length 50 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	8	$4.92 \times 10^{-2}$	$5.03 \times 10^{-3}$	$5.04 \times 10^{-4}$
56	9	$4.83 \times 10^{-2}$	$4.94 \times 10^{-3}$	$4.95 \times 10^{-4}$
48	10	$4.59 \times 10^{-2}$	$4.69 \times 10^{-3}$	$4.70 \times 10^{-4}$
40	13	$4.94 \times 10^{-2}$	$5.06 \times 10^{-3}$	$5.07 \times 10^{-4}$

**Table E-3: Probability  $P_{CX}$  of Codeblock Rejection in TED Mode, Frame Length 120 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	18	$1.07 \times 10^{-1}$	$1.13 \times 10^{-2}$	$1.13 \times 10^{-3}$
56	20	$1.04 \times 10^{-1}$	$1.09 \times 10^{-2}$	$1.10 \times 10^{-3}$
48	24	$1.07 \times 10^{-1}$	$1.12 \times 10^{-2}$	$1.13 \times 10^{-3}$
40	30	$1.10 \times 10^{-1}$	$1.16 \times 10^{-2}$	$1.17 \times 10^{-3}$

**Table E-4: Probability  $P_{CX}$  of Codeblock Rejection in TED Mode, Frame Length 256 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	37	$2.08 \times 10^{-1}$	$2.30 \times 10^{-2}$	$2.33 \times 10^{-3}$
56	43	$2.11 \times 10^{-1}$	$2.34 \times 10^{-2}$	$2.36 \times 10^{-3}$
48	52	$2.17 \times 10^{-1}$	$2.41 \times 10^{-2}$	$2.44 \times 10^{-3}$
40	64	$2.21 \times 10^{-1}$	$2.47 \times 10^{-2}$	$2.49 \times 10^{-3}$

In SEC mode, the probability  $P_{CY}$  of one of the BCH Codeblocks in a CLTU causing a codeblock rejection is:

$$P_{CY} = 1 - [(1 - p)^n + np(1 - p)^{(n-1)}]^N$$

where  $n$  is one less than the codeblock length and  $N$  is the number of codeblocks in the CLTU. Tables E-5 to E-8 show the values of the probability  $P_{CY}$  for the three channel bit error rates for different frame lengths.

**Table E-5: Probability  $P_{CY}$  of Codeblock Rejection in SEC Mode, Frame Length 20 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	3	$5.84 \times 10^{-5}$	$5.86 \times 10^{-7}$	$5.86 \times 10^{-9}$
56	4	$5.92 \times 10^{-5}$	$5.94 \times 10^{-7}$	$5.94 \times 10^{-9}$
48	4	$4.31 \times 10^{-5}$	$4.32 \times 10^{-7}$	$4.32 \times 10^{-9}$
40	5	$3.70 \times 10^{-5}$	$3.70 \times 10^{-7}$	$3.70 \times 10^{-9}$

**Table E-6: Probability  $P_{CY}$  of Codeblock Rejection in SEC Mode, Frame Length 50 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	8	$1.56 \times 10^{-4}$	$1.56 \times 10^{-6}$	$1.56 \times 10^{-8}$
56	9	$1.33 \times 10^{-4}$	$1.34 \times 10^{-6}$	$1.34 \times 10^{-8}$
48	10	$1.08 \times 10^{-4}$	$1.08 \times 10^{-6}$	$1.08 \times 10^{-8}$
40	13	$9.61 \times 10^{-5}$	$9.63 \times 10^{-7}$	$9.63 \times 10^{-9}$

**Table E-7: Probability  $P_{CY}$  of Codeblock Rejection in SEC Mode, Frame Length 120 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	18	$3.50 \times 10^{-4}$	$3.51 \times 10^{-6}$	$3.52 \times 10^{-8}$
56	20	$2.96 \times 10^{-4}$	$2.97 \times 10^{-6}$	$2.97 \times 10^{-8}$
48	24	$2.59 \times 10^{-4}$	$2.59 \times 10^{-6}$	$2.59 \times 10^{-8}$
40	30	$2.22 \times 10^{-4}$	$2.22 \times 10^{-6}$	$2.22 \times 10^{-8}$

**Table E-8: Probability  $P_{CY}$  of Codeblock Rejection in SEC Mode, Frame Length 256 Octets**

Codeblock length	Number of Codeblocks N	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	37	$7.19 \times 10^{-4}$	$7.22 \times 10^{-6}$	$7.23 \times 10^{-8}$
56	43	$6.36 \times 10^{-4}$	$6.38 \times 10^{-6}$	$6.39 \times 10^{-8}$
48	52	$5.60 \times 10^{-4}$	$5.62 \times 10^{-6}$	$5.62 \times 10^{-8}$
40	64	$4.73 \times 10^{-4}$	$4.74 \times 10^{-6}$	$4.74 \times 10^{-8}$

### E3 TAIL SEQUENCE FACTOR

When operating with PLOP-2, the Tail Sequence factor is relevant to the performance. As described in 8.4.5, if the Tail Sequence is not recognized, then the following CLTU may be missed.

Earlier CCSDS Recommendations had a different pattern for the CLTU Tail Sequence (see 4.5.3). With the earlier pattern, a single error could cause the Tail Sequence to be missed in SEC mode and two errors could cause it to be missed in TED mode.

Table E-9 shows the probabilities of missing the Tail Sequence for the earlier Tail Sequence pattern. The probability  $P_{TX}$  is for decoding in TED mode and  $P_{TY}$  is for decoding in SEC mode. The table shows the probabilities for different Tail Sequence lengths, because the length of the Tail Sequence is the same as the length of a BCH Codeblock.

The values in the table are based on the probabilities of one or two errors in the Tail Sequence. However, errors in some positions do not cause the Tail Sequence to be missed. For example, only 31 of the 1953 possible double errors in a 63-bit codeblock cause the Tail Sequence to become a valid codeblock. These factors have been taken into account in the table.

**Table E-9: Probabilities  $P_{TX}$  and  $P_{TY}$  of Missing a Tail Sequence with the Earlier Pattern**

Tail Sequence length (bits)	$P_{TX}$ $P_{TY}$	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	$P_{TX}$ (TED)	$3.08 \times 10^{-7}$	$3.10 \times 10^{-9}$	$3.10 \times 10^{-11}$
	$P_{TY}$ (SEC)	$6.16 \times 10^{-3}$	$6.20 \times 10^{-4}$	$6.20 \times 10^{-5}$
56	$P_{TX}$ (TED)	$2.39 \times 10^{-7}$	$2.40 \times 10^{-9}$	$2.40 \times 10^{-11}$
	$P_{TY}$ (SEC)	$4.77 \times 10^{-3}$	$4.80 \times 10^{-4}$	$4.80 \times 10^{-5}$
48	$P_{TX}$ (TED)	$1.69 \times 10^{-7}$	$1.70 \times 10^{-9}$	$1.70 \times 10^{-11}$
	$P_{TY}$ (SEC)	$3.38 \times 10^{-3}$	$3.40 \times 10^{-4}$	$3.40 \times 10^{-5}$
40	$P_{TX}$ (TED)	$1.20 \times 10^{-7}$	$1.20 \times 10^{-9}$	$1.20 \times 10^{-11}$
	$P_{TY}$ (SEC)	$2.39 \times 10^{-3}$	$2.40 \times 10^{-4}$	$2.40 \times 10^{-5}$

Various techniques were introduced to reduce the risk of missing a CLTU because of failing to recognize the Tail Sequence. One of these involved inserting an extra Tail Sequence at the end of each CLTU, so that each CLTU had a double Tail Sequence.



If the first Tail Sequence was not recognized, there was a high probability that the second Tail Sequence would be recognized and so the next CLTU would not be missed. Table E-10 shows the probabilities of missing both Tail Sequences (earlier pattern) when the double Tail Sequence technique is used.

**Table E-10: Probabilities  $P_{TXD}$  and  $P_{TYD}$  of Missing Both Tail Sequences (Earlier Pattern) When a Double Tail Sequence Is Used**

Tail Sequence length (bits)	$P_{TXD}$ $P_{TYD}$	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	$P_{TXD}$ (TED)	$9.5 \times 10^{-14}$	$9.6 \times 10^{-18}$	$9.6 \times 10^{-22}$
	$P_{TYD}$ (SEC)	$3.8 \times 10^{-5}$	$3.8 \times 10^{-7}$	$3.8 \times 10^{-9}$
56	$P_{TXD}$ (TED)	$5.7 \times 10^{-14}$	$5.8 \times 10^{-18}$	$5.8 \times 10^{-22}$
	$P_{TYD}$ (SEC)	$2.3 \times 10^{-5}$	$2.3 \times 10^{-7}$	$2.3 \times 10^{-9}$
48	$P_{TXD}$ (TED)	$2.9 \times 10^{-14}$	$2.9 \times 10^{-18}$	$2.9 \times 10^{-22}$
	$P_{TYD}$ (SEC)	$1.1 \times 10^{-5}$	$1.2 \times 10^{-7}$	$1.2 \times 10^{-9}$
40	$P_{TXD}$ (TED)	$1.4 \times 10^{-14}$	$1.4 \times 10^{-18}$	$1.4 \times 10^{-22}$
	$P_{TYD}$ (SEC)	$5.7 \times 10^{-6}$	$5.8 \times 10^{-8}$	$5.8 \times 10^{-10}$

The Filler Bit Augmentation (FBA) algorithm described in 6.5.3.3 was another technique to reduce the risk of failing to recognize the Tail Sequence with the earlier pattern. The algorithm applies only to SEC mode. Table E-11 shows the probability  $P_{TYF}$  of missing the Tail Sequence when the algorithm is used.

**Table E-11: Probability  $P_{TYF}$  of Missing a Tail Sequence (Earlier Pattern) When Filler Bit Augmentation Is Used in SEC Mode**

Tail Sequence length (bits)	Channel Bit Error Rate		
	$10^{-4}$	$10^{-5}$	$10^{-6}$
64	$9.24 \times 10^{-7}$	$9.29 \times 10^{-9}$	$9.30 \times 10^{-11}$
56	$7.16 \times 10^{-7}$	$7.20 \times 10^{-9}$	$7.20 \times 10^{-11}$
48	$5.08 \times 10^{-7}$	$5.10 \times 10^{-9}$	$5.10 \times 10^{-11}$
40	$3.59 \times 10^{-7}$	$3.60 \times 10^{-9}$	$3.60 \times 10^{-11}$

For comparison, table E-12 shows the probabilities of missing the current Tail Sequence (see 4.5.1) for all lengths in TED mode, in SEC mode, and in SEC mode with Filler Bit Augmentation.

**Table E-12: Probabilities of Missing a Tail Sequence (Current Pattern) for Different Modes and Lengths**

Tail Sequence length (bits)	Mode	Channel Bit Error Rate		
		$10^{-4}$	$10^{-5}$	$10^{-6}$
64	TED	$6.47 \times 10^{-10}$	$6.51 \times 10^{-13}$	$6.51 \times 10^{-16}$
	SEC	$1.94 \times 10^{-5}$	$1.95 \times 10^{-7}$	$1.95 \times 10^{-9}$
	SEC + FBA	$2.59 \times 10^{-9}$	$2.60 \times 10^{-12}$	$2.60 \times 10^{-15}$
56	TED	$4.27 \times 10^{-10}$	$4.29 \times 10^{-13}$	$4.29 \times 10^{-16}$
	SEC	$1.28 \times 10^{-5}$	$1.29 \times 10^{-7}$	$1.29 \times 10^{-9}$
	SEC + FBA	$1.71 \times 10^{-9}$	$1.72 \times 10^{-12}$	$1.72 \times 10^{-15}$
48	TED	$2.60 \times 10^{-10}$	$2.61 \times 10^{-13}$	$2.61 \times 10^{-16}$
	SEC	$7.80 \times 10^{-6}$	$7.83 \times 10^{-8}$	$7.83 \times 10^{-10}$
	SEC + FBA	$1.04 \times 10^{-9}$	$1.04 \times 10^{-12}$	$1.04 \times 10^{-15}$
40	TED	$1.51 \times 10^{-10}$	$1.52 \times 10^{-13}$	$1.52 \times 10^{-16}$
	SEC	$4.54 \times 10^{-6}$	$4.56 \times 10^{-8}$	$4.56 \times 10^{-10}$
	SEC + FBA	$6.06 \times 10^{-10}$	$6.08 \times 10^{-13}$	$6.08 \times 10^{-16}$

## ANNEX F

### PRACTICAL EXAMPLES OF FRAMES AND CLTUS

This annex contains a set of practical examples of TC Transfer Frames and CLTUs.

The example TC Transfer Frames show the construction of the frames, including the calculated Frame Error Control Fields which contain the CRC for error detection. The TC Transfer Frame is specified in reference [4].

The example CLTUs show the CLTU components. The Error Control Field can be seen in the last octet of each BCH Codeblock. The CLTUs are shown with and without randomization. As described in 7.2.1, the randomization of any fill bits which may be present in the last BCH Codeblock of a CLTU is optional. In the CLTUs in this annex, the fill bits are not randomized.

The TC Transfer Frames and CLTUs are shown in hexadecimal notation. For the TC Transfer Frame and CLTU the left-most octet is transmitted first and the most significant bit of each octet is the first bit of the octet to be transmitted.

NOTE – The Acquisition Sequences and Idle Sequences before, between and after the CLTUs are not shown in the examples here. These sequences depend on the Physical Layer Operations Procedure in use, as described in section 5.

**Example 1: Type BC Frame with Control Command ‘Unlock’**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	BC (shows that the frame carries a control command)
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	1 octet
Frame Data Field content	00 (control command ‘Unlock’)
Frame length	8 octets
Frame Error Control Field	4C A9
<b>TC Transfer Frame</b>	30 1B 00 07 00 00 4C A9
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	30 1B 00 07 00 00 4C A4
2 <sup>nd</sup> BCH Codeblock	A9 55 55 55 55 55 55 7A
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	CF 22 9E 5D 68 E9 4A FC
2 <sup>nd</sup> BCH Codeblock	5C 55 55 55 55 55 55 F4
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 2: Type BC Frame with Control Command ‘Set V(R)’**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	BC (shows that the frame carries a control command)
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	3 octets
Frame Data Field content	82 00 00 (control command ‘Set V(R)’, V(R)= 0)
Frame length	10 octets
Frame Error Control Field	F6 F0
<b>TC Transfer Frame</b>	30 1B 00 09 00 82 00 00 F6 F0
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	30 1B 00 09 00 82 00 54
2 <sup>nd</sup> BCH Codeblock	00 F6 F0 55 55 55 55 D6
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	CF 22 9E 53 68 6B 06 0C
2 <sup>nd</sup> BCH Codeblock	F5 9A 79 55 55 55 55 06
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 3: Type BC Frame with Control Command ‘Set V(R)’**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	BC (shows that the frame carries a control command)
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	3 octets
Frame Data Field content	82 00 10 (control command ‘Set V(R)’, V(R)= 16)
Frame length	10 octets
Frame Error Control Field	E4 C1
<b>TC Transfer Frame</b>	30 1B 00 09 00 82 00 10 E4 C1
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	30 1B 00 09 00 82 00 54
2 <sup>nd</sup> BCH Codeblock	10 E4 C1 55 55 55 55 3E
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	CF 22 9E 53 68 6B 06 0C
2 <sup>nd</sup> BCH Codeblock	E5 88 48 55 55 55 55 EE
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 4: Type BC Frame with Control Command ‘Set V(R)’**

Spacecraft Identifier	1B
Virtual Channel Identifier	1
Frame Type	BC (shows that the frame carries a control command)
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	3 octets
Frame Data Field content	82 00 00 (control command ‘Set V(R)’, V(R)= 0)
Frame length	10 octets
Frame Error Control Field	F0 51
<b>TC Transfer Frame</b>	30 1B 04 09 00 82 00 00 F0 51
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	30 1B 04 09 00 82 00 E8
2 <sup>nd</sup> BCH Codeblock	00 F0 51 55 55 55 55 AA
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	CF 22 9A 53 68 6B 06 B0
2 <sup>nd</sup> BCH Codeblock	F5 9C D8 55 55 55 55 7A
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 5: Type BC Frame with Control Command ‘Set V(R)’**

Spacecraft Identifier	1B
Virtual Channel Identifier	1
Frame Type	BC (shows that the frame carries a control command)
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	3 octets
Frame Data Field content	82 00 10 (control command ‘Set V(R)’, V(R)= 16)
Frame length	10 octets
Frame Error Control Field	E2 60
<b>TC Transfer Frame</b>	30 1B 04 09 00 82 00 10 E2 60
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	30 1B 04 09 00 82 00 E8
2 <sup>nd</sup> BCH Codeblock	10 E2 60 55 55 55 55 42
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	CF 22 9A 53 68 6B 06 B0
2 <sup>nd</sup> BCH Codeblock	E5 8E E9 55 55 55 55 92
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79



**Example 6: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD (the frame carries data on Sequence-Controlled service)
Frame Sequence Number	255
Frame Data Field length	1 octet
Frame Data Field content	01
Frame length	8 octets
Frame Error Control Field	70 FB
<b>TC Transfer Frame</b>	00 1B 00 07 FF 01 70 FB
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 07 FF 01 70 E8
2 <sup>nd</sup> BCH Codeblock	FB 55 55 55 55 55 55 5C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 5D 97 E8 76 B0
2 <sup>nd</sup> BCH Codeblock	0E 55 55 55 55 55 55 D2
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 7: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	0
Frame Data Field length	2 octets
Frame Data Field content	01 02
Frame length	9 octets
Frame Error Control Field	BE 58
<b>TC Transfer Frame</b>	00 1B 00 08 00 01 02 BE 58
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 08 00 01 02 12
2 <sup>nd</sup> BCH Codeblock	BE 58 55 55 55 55 55 0C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 52 68 E8 04 4A
2 <sup>nd</sup> BCH Codeblock	4B 34 55 55 55 55 55 A0
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 8: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	1
Frame Data Field length	3 octets
Frame Data Field content	01 02 03
Frame length	10 octets
Frame Error Control Field	F2 93
<b>TC Transfer Frame</b>	00 1B 00 09 01 01 02 03 F2 93
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 09 01 01 02 0A
2 <sup>nd</sup> BCH Codeblock	03 F2 93 55 55 55 55 5C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 53 69 E8 04 52
2 <sup>nd</sup> BCH Codeblock	F6 9E 1A 55 55 55 55 8C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 9: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	2
Frame Data Field length	4 octets
Frame Data Field content	01 02 03 04
Frame length	11 octets
Frame Error Control Field	3C EB
<b>TC Transfer Frame</b>	00 1B 00 0A 02 01 02 03 04 3C EB
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 0A 02 01 02 22
2 <sup>nd</sup> BCH Codeblock	03 04 3C EB 55 55 55 44
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 50 6A E8 04 7A
2 <sup>nd</sup> BCH Codeblock	F6 68 B5 C4 55 55 55 0A
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 10: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	6
Frame Data Field length	8 octets
Frame Data Field content	01 02 03 04 05 06 07 08
Frame length	15 octets
Frame Error Control Field	14 BB
<b>TC Transfer Frame</b>	00 1B 00 0E 06 01 02 03 04 05 06 07 08 14 BB
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 0E 06 01 02 42
2 <sup>nd</sup> BCH Codeblock	03 04 05 06 07 08 14 12
3 <sup>rd</sup> BCH Codeblock	BB 55 55 55 55 55 55 3E
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 54 6E E8 04 1A
2 <sup>nd</sup> BCH Codeblock	F6 68 8C 29 A6 39 4A 5E
3 <sup>rd</sup> BCH Codeblock	B3 55 55 55 55 55 55 C6
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 11: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	7
Frame Data Field length	9 octets
Frame Data Field content	01 02 03 04 05 06 07 08 09
Frame length	16 octets
Frame Error Control Field	CF 90
<b>TC Transfer Frame</b>	00 1B 00 0F 07 01 02 03 04 05 06 07 08 09 CF 90
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 0F 07 01 02 5A
2 <sup>nd</sup> BCH Codeblock	03 04 05 06 07 08 09 8E
3 <sup>rd</sup> BCH Codeblock	CF 90 55 55 55 55 55 AE
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 55 6F E8 04 02
2 <sup>nd</sup> BCH Codeblock	F6 68 8C 29 A6 39 57 C2
3 <sup>rd</sup> BCH Codeblock	C7 50 55 55 55 55 55 AA
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 12: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	0
Frame Data Field length	11 octets
Frame Data Field content	C0 10 00 C0 00 00 03 2E AF 8A 06
Frame length	18 octets
Frame Error Control Field	9F 71
<b>TC Transfer Frame</b>	00 1B 00 11 00 C0 10 00 C0 00 00 03 2E AF 8A 06 9F 71
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 11 00 C0 10 14
2 <sup>nd</sup> BCH Codeblock	00 C0 00 00 03 2E AF 9E
3 <sup>rd</sup> BCH Codeblock	8A 06 9F 71 55 55 55 48
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 4B 68 29 16 4C
2 <sup>nd</sup> BCH Codeblock	F5 AC 89 2F A2 1F F1 D2
3 <sup>rd</sup> BCH Codeblock	82 C6 CD D9 55 55 55 1C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 13: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	0
Frame Data Field length	12 octets
Frame Data Field content	C1 10 00 FF FF 00 04 01 02 03 11 82
Frame length	19 octets
Frame Error Control Field	8D 80
<b>TC Transfer Frame</b>	00 1B 00 12 00 C1 10 00 FF FF 00 04 01 02 03 11 82 8D 80
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 12 00 C1 10 1C
2 <sup>nd</sup> BCH Codeblock	00 FF FF 00 04 01 02 E2
3 <sup>rd</sup> BCH Codeblock	03 11 82 8D 80 55 55 F0
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 48 68 28 16 44
2 <sup>nd</sup> BCH Codeblock	F5 93 76 2F A5 30 5C AE
3 <sup>rd</sup> BCH Codeblock	0B D1 D0 25 3B 55 55 98
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79



**Example 14: Type AD Frame for Data on the Sequence-Controlled Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	AD
Frame Sequence Number	0
Frame Data Field length	12 octets
Frame Data Field content	C1 11 04 C0 00 00 04 01 02 03 72 17
Frame length	19 octets
Frame Error Control Field	8D 80
<b>TC Transfer Frame</b>	00 1B 00 12 00 C1 11 04 C0 00 00 04 01 02 03 72 17 8D 80
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	00 1B 00 12 00 C1 11 96
2 <sup>nd</sup> BCH Codeblock	04 C0 00 00 04 01 02 F4
3 <sup>rd</sup> BCH Codeblock	03 72 17 8D 80 55 55 00
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	FF 22 9E 48 68 28 17 CE
2 <sup>nd</sup> BCH Codeblock	F1 AC 89 2F A5 30 5C B8
3 <sup>rd</sup> BCH Codeblock	0B B2 45 25 3B 55 55 68
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 15: Type BD Frame for Data on the Expedited Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	BD
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	1 octet
Frame Data Field content	E1
Frame length	8 octets
Frame Error Control Field	BB 22
<b>TC Transfer Frame</b>	20 1B 00 07 00 E1 BB 22
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	20 1B 00 07 00 E1 BB 38
2 <sup>nd</sup> BCH Codeblock	22 55 55 55 55 55 55 A2
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	DF 22 9E 5D 68 08 BD 60
2 <sup>nd</sup> BCH Codeblock	D7 55 55 55 55 55 55 2C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79

**Example 16: Type BD Frame for Data on the Expedited Service**

Spacecraft Identifier	1B
Virtual Channel Identifier	0
Frame Type	BD
Frame Sequence Number	0 (all type-B frames have sequence number 0)
Frame Data Field length	11 octets
Frame Data Field content	C0 10 00 C0 00 00 03 2E AF 8A 06
Frame length	18 octets
Frame Error Control Field	D9 65
<b>TC Transfer Frame</b>	20 1B 00 11 00 C0 10 00 C0 00 00 03 2E AF 8A 06 D9 65
<b>CLTU (unrandomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	20 1B 00 11 00 C0 10 E0
2 <sup>nd</sup> BCH Codeblock	00 C0 00 00 03 2E AF 9E
3 <sup>rd</sup> BCH Codeblock	8A 06 D9 65 55 55 55 2C
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79
<b>CLTU (randomized)</b>	
Start Sequence	EB 90
1 <sup>st</sup> BCH Codeblock	DF 22 9E 4B 68 29 16 B8
2 <sup>nd</sup> BCH Codeblock	F5 AC 89 2F A2 1F F1 D2
3 <sup>rd</sup> BCH Codeblock	82 C6 8B CD 55 55 55 78
Tail Sequence	C5 C5 C5 C5 C5 C5 C5 79