431-HDBK-000052
Effective Date: 9/1/2006
Expiration Date: 9/1/2011

Lunar Reconnaissance Orbiter (LRO)

Telemetry and Command Formats Handbook

August 23, 2006

LRO GSFC CMO

September 1, 2006

RELEASED

National Aeronautics and Space Administration

CM FOREWORD

This document is a Lunar Precursor and Robotic Program (LPRP) Configuration Management (CM)-controlled document. Changes to this document require prior approval of the applicable Configuration Control Board (CCB) Chairperson or designee. Proposed changes will be submitted to the LPRP CM Office (CMO), along with supportive material justifying the proposed change. Changes to this document will be made by complete revision.

Questions or comments concerning this document should be addressed to:

LPRP Configuration Management Office
Mail Stop 451
Goddard Space Flight Center
Greenbelt, Maryland 20771

Signature Page

Prepared by:

8/23/06

Jim Clapsadle — Date
LRO Ground Systems Engineer
HTSI/GSFC, Code 444

Reviewed by:

8/23/06

Richard Saylor Jr — Date
LRO Ground Systems and Operations Lead
HTSI/GSFC, Code 444

Reviewed by:

8/31/06

Mike Blau — Date
LRO Flight Software Manager
NASA/GSFC, Code 582

Approved by:

for — 8/23/06

Dave Everett — Date
LRO Mission Systems Engineer
NASA/GSFC, Code 599

ROBOTIC LUNAR EXPLORATION PROGRAM

DOCUMENT CHANGE RECORD

Sheet: 1 of 1

REV LEVEL	DESCRIPTION OF CHANGE	APPROVED BY	DATE APPROVED
Rev -	Released to 431-CCR-000180	D. Everett	8/23/2006

List of TBDs/TBRs

Item No.	Location	Summary	Ind./Org.	Due Date
1.	3.3.1	Decision on whether or not frame error control may be used to meet authentication requirements	J. Clapsadle	9/30/06
2.	4.6.5	Define engineering and housekeeping file size limitation	J. Clapsadle	9/30/06
3.	**Error! Reference source not found.**	Define spacecraft event file format	J. Clapsadle	9/30/06
4.	Table 4-8	Defined file types for all values in the table and verify the information with LROC which conflicts with the information provided in table 4-11	J. Clapsadle	9/30/06
5.	Table 4-9	Rick will verify with Mike Blau whether the instrument data formats will be represented in this documentation or if they will remain in the instrument data ICDs	R. Saylor	9/30/06
6.	Table 4-10	Provide valid values when they become available for the header field contents	R. Saylor	9/30/06
7.	Table 4-11	Table needs to be defined for all file types and default configurations need to be verified with the instrument teams	J. Clapsadle	9/30/06
8.	Table 5-7	Fault location for the EOF PDU needs further defined and the diagram may need to be updated to reflect this	J. Clapsadle	9/30/06

TABLE OF CONTENTS

Page

LIST OF FIGURES

<u>Figure</u> <u>Page</u>

LIST OF TABLES

<u>Table</u> <u>Page</u>

1.0 Introduction

This Consultative Committee on Space Data Systems (CCSDS) implementation document records LRO requirements (management decisions) on the use of CCSDS recommendations on the LRO Mission. The document also notes where LRO will deviate from CCSDS recommendations and explains some particulars about the use of CCSDS data structures for the LRO Mission.

1.1 Purpose

The purpose of this document is to define fully the telemetry and command formatting standards for LRO. This document will define the LRO implementation of the CCSDS recommendations for the communication of packetized data. This document will serve as an interface control document between the spacecraft designers and the ground system designers, as well as between the spacecraft integrators and the ground system operators.

1.2 Scope

The scope of this document covers the formats of the LRO telemetry and commands. It does not attempt to define the actions of the commands or the results of the telemetry. Sections 3 and 4 specify the CCSDS defined telemetry and command transport protocol layers; physical, coding, transfer, segmentation, packetization, and system management/application. Section 5 specifies the use and decomposition of the CCSDS File Delivery Protocol (CFDP).

This document applies to the following interfaces as shown in figure 1-1.

1- The CCSDS data structures utilized for the transfer of telemetry over the Ka-Band RF Link
2- The CCSDS data structures utilized for the transfer of telemetry and commands over the S-Band RF Link
3- The file data structures utilized for the orbiter and the instruments

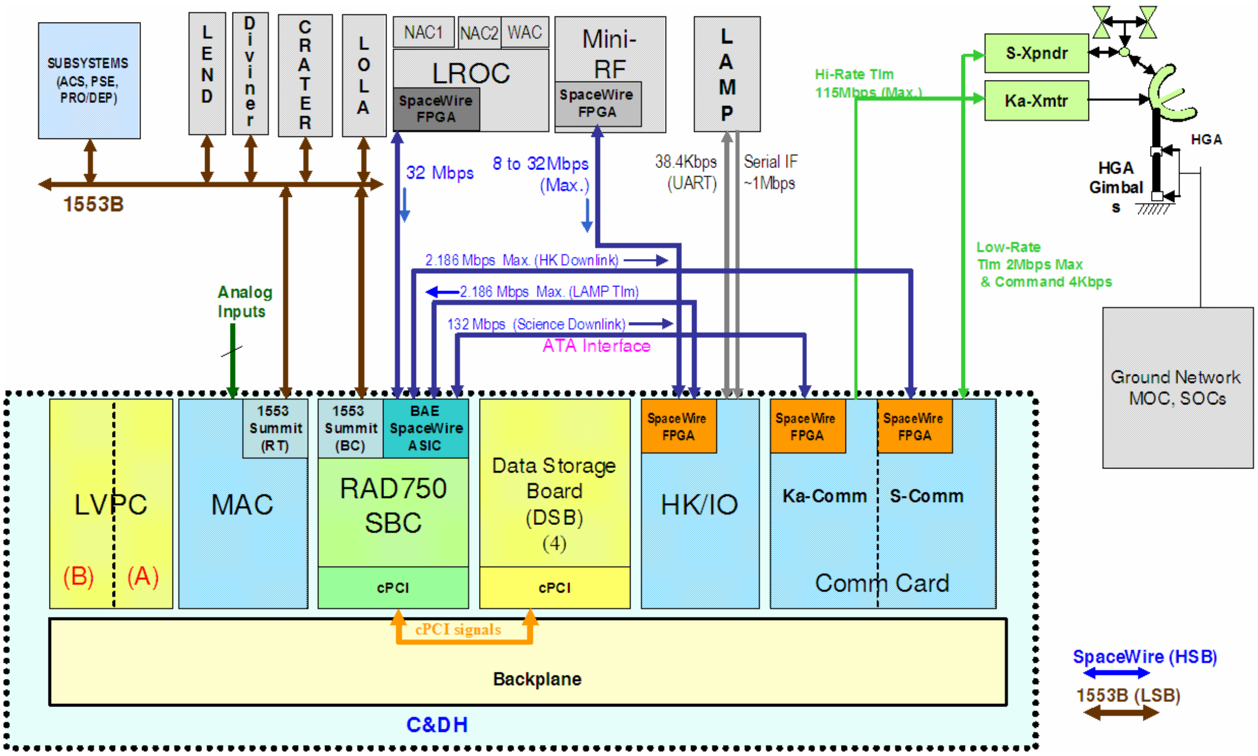

Figure 1-1. LRO CCSDS Data Interfaces Scope

1.3 Terminology and Definitions

Within this document, the terms *octet* and *byte* are used interchangeably. The terms refer to eight contiguous bits in the telemetry and command data streams. Octets or bytes are always offset by a multiple of eight bits from the first bit in a packet, transfer frame, or code block. The term *word*, when used in the context of telemetry word, is analogous to the terms octet and byte unless otherwise noted. The terms *word* and *longword*, when used in reference to data types, refers to 16 and 32 bit integer data fields, respectively. Numeric values in this document are expressed in decimal notation unless otherwise noted.

In addition to familiarity with the CCSDS concepts and standards, this document assumes general knowledge of NASA/GSFC institutional functions, standards and practices, documentation, and terminology.

1.4 Applicable Documents

Documents applicable to this handbook are listed below.

1- CCSDS, Space Packet Protocol, CCSDS 133.0-B-1, Blue Book September 2003
2- CCSDS, Advanced Orbiting Space Data Link Protocol, CCSDS 732.0-B-1, Blue Book September 2003
3- CCSDS, TM Space Data Link Protocol, CCSDS 132.0-B-1, Blue Book, September 2003

4- CCSDS, TM Synchronization and Channel Coding, CCSDS 131.0-B-1, Blue Book September 2003
5- CCSDS, Time Code Formats, CCSDS 301.0-B-3, Blue Book, January 2002
6- CCSDS, TC Space Data Link Protocol, CCSDS 232.0-B-1, Blue Book, September 2003
7- CCSDS, Communications Operations Procedure-1, CCSDS 232.1-B-1, Blue Book, September 2003
8- CCSDS, TC Synchronization and Channel Coding, CCSDS 231.0-B-1, Blue Book, September 2003
9- CCSDS File Delivery Protocol (CFDP). Recommendation for Space Data System Standards, CCSDS 727.0-B-3, Blue Book, June 2005
10- CCSDS File Delivery Protocol—Part 1: Introduction and Overview. Report Concerning Space Data Systems Standards, CCSDS 720.2-G-1, Green Book, September 2003
11- CCSDS File Delivery Protocol—Part 2: Implementers Guide. Report Concerning Space Data Systems Standards, CCSDS 720.2-G-2, Green Book, September 2003
12- 451-RFICD-LRO/GN/DSN/SN - Lunar Reconnaissance Orbiter Space-to-Ground Radio Frequency Interface Control Document
13- 431-HDBK-000076 - LRO Mission Telemetry and Command Database Style Guide
14- 431-PLAN-000051 - LRO Telemetry and Command Database Management Plan
15- 431-SPEC-000078 - Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification
16- 431-ICD-000049 - Lunar Reconnaissance Orbiter Ground System Interface Control Document
17- 0163-Telecomm - Space Link Extension Forward Link Service and Return Link Service
18- 431-HDBK-000471 – LRO Command and Data Handling Flight Software User’s Guide
19- 431-HDBK-000053 – LRO Telemetry and Command Database Handbook Volume 2

2.0 CCSDS Spacecraft Identification

LRO has requested and received approval to use the following Global Spacecraft Identifier (GSCID) for the mission.

The LRO Spacecraft Identifier (SCID) will be unique for commands and telemetry. A value of 0xA5 or 00 1010 0101 binary will be used for command frames and a value of 0xFF or 1111 1111 binary will be used for telemetry frames.

For Telecommand Transfer Frames the Version Number will be VN=00 and thus the Global Spacecraft Identifier will be GSCID =VN.SCID = 00.00 1010 0101.

For Advanced Orbiting Systems(AOS) frames for the S-Band and Ka-Band RF Links the Version Number will be VN=01 and thus the GSCID will be
GSCID = VN.SCID = 01.1111 1111.

3.0 TELECOMMAND

The LRO implementation of telecommand capabilities is a subset of what the CCSDS standards provide (CCSDS 231.0-B-1, CCSDS 232.0-B-1 and CCSDS 133.0-B-1). The size and scope of LRO does not warrant the use of all of the options provided in the CCSDS standard. The complete telecommand data structure overview is shown in Figure 3-1 and will be decomposed into its specific elements through the rest of this section.

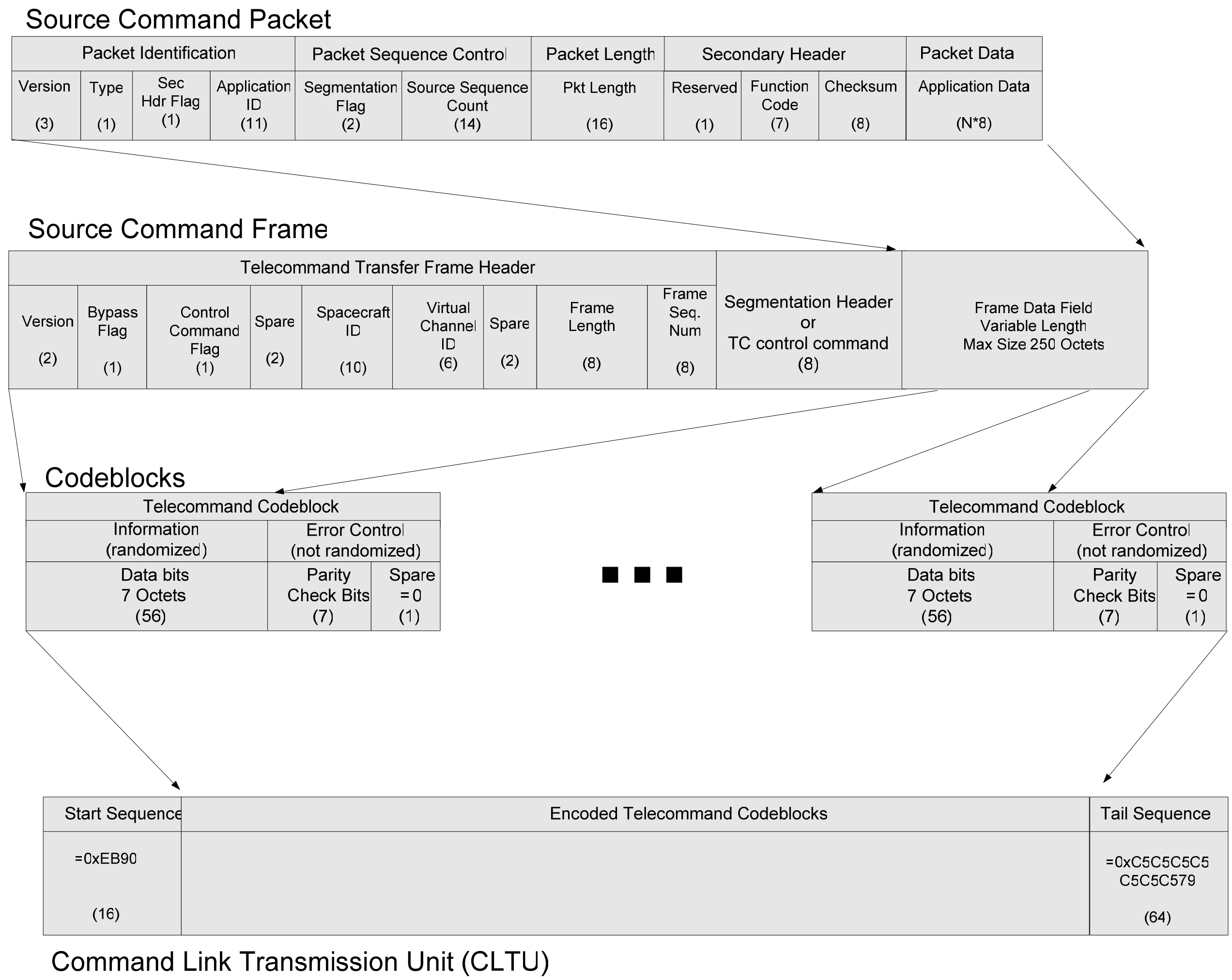

Figure 3-1. LRO Command CCSDS Data Structures Overview

3.1 PHYSICAL LAYER

The Physical layer provides the radio frequency data path which connects the transmitting station to the spacecraft, and it's associated Physical Layer Operational Procedures (PLOPs), in order to support the transmission of telecommand data. For signal formatting, modulation, and RF characteristics refer to the Lunar Reconnaissance Orbiter Space-to-Ground Radio Frequency

Interface Control Document (451-RFICD-LRO/GN/DSN/SN).The standard data structures within this layer are the Acquisition Sequence, CLTU, and the Idle Sequence. They are used to provide synchronization of the symbol stream, and are described below, along with the PLOPs.

3.1.1 Acquisition Sequence

The acquisition sequence is a data structure forming a preamble that provides for initial symbol synchronization within the incoming stream of detected symbols. The length of the acquisition sequence will be at least 132 bits. The pattern of the acquisition sequence will be alternating "ones" and "zeros", starting with either a "one" or a "zero".

3.1.2 Command Link Transmission Unit (CLTU)

The CLTU is the data structure containing the symbol sequence. The structure is furnished from the Coding Layer and is described in section 3.2.2. It is used to synchronize and delimit the bit stream. The CLTU guarantees at least two data transitions per codeblock.

3.1.3 Idle Sequence

The Idle Sequence is the data structure that provides for maintenance of symbol synchronization in the absence of CLTUs. The bit pattern is a sequence of alternating "ones" and "zeros", beginning with a "zero". The length of the idle sequence is an unconstrained number of bits.

3.1.4 Physical Link Operation Procedures (PLOPs)

A PLOP consists of the sequential application of various Carrier Modulated Modes to activate and deactivate the physical telecommand channel. PLOP-2 will be used by the LRO spacecraft and ground system as described in section 6.5 of CCSDS 231.0-B-1. PLOP-2 is the procedure where physical telecommand channel is not deactivated after each transmitted CLTU. The termination of an individual CLTU is provided only through the data path, using the CLTU Tail Sequence and, optionally, Idle Sequences; and not the decoder itself.

3.2 CODING LAYER

The Coding layer establishes the reliable, error-controlled data channel through which user telecommand data bits may be transferred. The data is encoded to reduce the effects of noise in the Physical Layer on the user data. There are two standard data structures used in the Coding Layer, the TC Codeblock and the CLTU.

3.2.1 Codeblock

The LRO code blocks conform to the CCSDS recommendations as described in section 3.2.1 of CCSDS 231.0-B-1.

Each telecommand transfer frame is piece-wise encoded into a series of short, sequential, fixed length telecommand code blocks which provide error detection capabilities. TC Code Blocks shown in Figure 3-2 are 64-bit structures that consist of 7 octets (56-bits) of data, 7 parity bits (one per octet) and a spare bit that must always be 0. The complements of the seven parity bits are stored in this last octet. The complements are used to aid in maintaining bit synchronization and detection of bit slippage. The last fill bit is always set to "0". The parity check bits for the telecommand code blocks are computed according to section 3.3.2 CCSDS 231.0-B-1. LRO code blocks received by the spacecraft that fail the parity check will be discarded and no attempt to correct the code block will be made.

Telecommand Codeblock		
Information (randomized)	Error Control (not randomized)	
Data bits 7 Octets (56)	Parity Check Bits (7)	Spare =0 (1)

Figure 3-2. Telecommand Codeblock

3.2.1.1 TC Codeblock Encoding

The TC codeblocks are block coded with a (63, 56) modified Bose-Chaudhuri-Hocquenghem (BCH) code. The parity bits in the last octet of the codeblock are the complements of the BCH code parity bits. The generator polynomial to produce the seven parity bits is:

$$g(x) = x^7 + x^6 + x^2 + x^0$$

TC Codeblock encoding is performed on all data excluding the CLTU start and tail sequences.

3.2.1.2 TC Codeblock Fill Bits

Fill bits inserted in any code block will conform to the recommendation in section 3.4 of CCSDS 231.0-B-1. If the "Input Data" does not fit exactly within an integral number of TC Codeblocks, the last octet(s) and ONLY the last octet(s) of the last Codeblock within the CLTU may contain "Fill" bits. The pattern of the fill will consist of a sequence of alternating "ones" and "zeros" starting with a "zero".

3.2.2 Command Link Transfer Units (CLTUs)

The LRO CLTU will conform to the CCSDS recommendations in section 4.2 of CCSDS 231.0-B-1.

The CLTU is the data structure that carries the TC data as a contiguous series of encoded TC Codeblocks across the coding and physical layers of CCSDS. A Command Link Transmission unit consists of 3 parts as noted in figure 3-3. The first part is a start sequence, which is the fixed pattern 0xEB90 also known as the Barker Code. The second part is a series of one or more code blocks. Code blocks are described in the previous section. The last part is a tail sequence, which is the fixed pattern 0xC5C5C5C5C5C5C579.

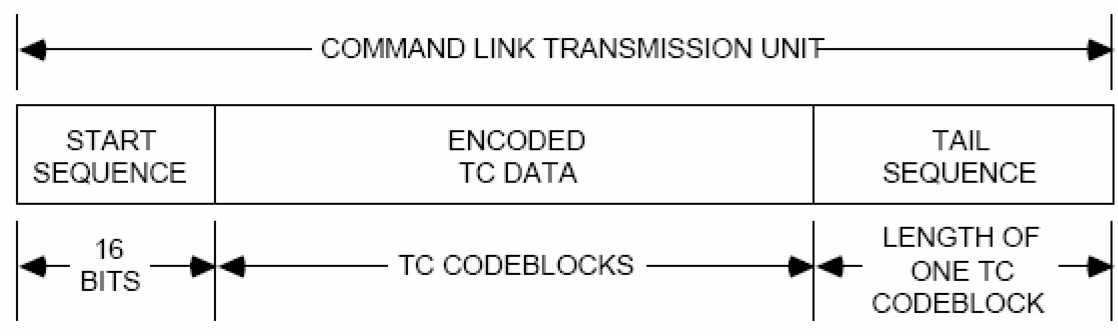

Figure 3-3. LRO Command Link Transmission Unit

There is a 1:1 correlation between a CLTU and command frames, where the command frames can be variable in size up to 256 bytes. The maximum CLTU length is 306 bytes and is calculated as follows:

Total byte size = 2 bytes (start sequence) + maximum number of codeblocks * 8 bytes + 8 bytes (tail sequence)

Maximum number of codeblocks = 259 x 8/7 = 37, where 259 equals maximum frame size plus 3 bytes of fill data to create an equal amount of codeblocks.

Therefore;
Total byte size = 2 bytes (start sequence) + 37 codeblocks * 8 bytes + 8 bytes (tail sequence)
Total byte size = 306 bytes

3.2.3 Pseudo- Randomization

Pseudo-Randomization is a bandwidth-efficient technique of algorithmically translating the data bits to insure frequent bit transitions in the communications channel. In order to maintain bit (or symbol) synchronization with the received communications signal, LRO requires that the incoming signal must have a minimum bit transition density. LRO has chosen to pseudo-randomize the uplink per CCSDS recommendations as specified in section 5 of CCSDS 231.0-B-1.

3.3 TRANSFER LAYER

3.3.1 Transfer Frames

The transfer frame is the data structure that is "uplinked" to the spacecraft. The transfer frame Figure 3-4 is made up of two major fields: Frame Header, and Frame Data Field. LRO will not support Frame Error Control (TBR).

Telecommand Transfer Frame Header									Segmentation Header or TC control command (8)	Frame Data Field Variable Length Max Size 250 Octets
Version (2)	Bypass Flag (1)	Control Command Flag (1)	Spare (2)	Spacecraft ID (10)	Virtual Channel ID (6)	Spare (2)	Frame Length (8)	Frame Seq. Num (8)		

Figure 3-4. Telecommand Transfer Frame

Table 3-1: Telecommand Transfer Frame

Field Name	# of bits	Value	LRO Usage
Version No.	2	‘00’	Indicate a CCSDS version 1 TC Transfer Frame is used.
ByPass Flag	1	0-1	This one bit field will be set to indicate whether "Frame Acceptance Checks" associated with the Command Operation Procedure (COP) will be performed (set to "0") or bypassed (set to "1"). Commands may be sent in bypass mode during conditions that call for commanding with no telemetry feedback. COP is described in 3.3.1.3.
Control Command Flag	1	0-1	This one bit field will be set to "0" if the frame carries a transfer layer Data Unit. Otherwise the bit will be set to "1" to indicate the frame carries a transfer layer Control Command.
Spare	2	‘00’	Not used by LRO
Spacecraft Identification	10	‘0010100101’	These 10 bits carry the spacecraft identifier. The spacecraft ID for LRO commanding is specified in section 2.0, it includes a VN=00 plus the SCID for commands of 0xA5.
Virtual Channel Identification	6	‘000001’	These 6 bits are used to indicate which virtual channel is being used. VC1 for all normal commanding activities.
Spare	2	‘00’	Not used by LRO
Frame Length	8	Variable up to 255	This eight bit field will contain the length of the transfer frame in octets. The count is one less than the total octets in the TC Transfer Frame. The length is measured from the first bit of the frame header to the last bit of the frame data field. The maximum length of the transfer frame is 256 bytes. The minimum length restriction is 6 bytes for control commands and 14 bytes for all other commands.

Field Name	# of bits	Value	LRO Usage
Frame Sequence Number	8	Variable	The Frame Sequence Number is an up-counting binary number (modulo 256). If the TC frame is a Type-B (frame acceptance checks to be bypassed), then the frame sequence number is set to "00000000".
Segmentation Header or TC control command	8	See Section References in Usage column	When the Control Command Flag in the Frame Header is set to "0", the segmentation layer is in effect and the data field will contain a TC Data Unit. When the flag is set to "1", this field contains a TC Control Command. Segmentation is described in section 3.4 TC Control Command is described in section 3.3.1.1.2
Frame Data Field	Up to 250		The data field can contain either TC Data Unit which includes the segmentation layer and command packets or a TC Control Command but not both. When the Control Command Flag in the Frame Header is set to "0", the data field will contain a TC Data Unit. When the flag is set to "1", the data field will contain a TC Control Command.

3.3.1.1 Frame Data Field

3.3.1.1.1 TC Data Unit

This is a variable length field of up to a maximum 251 octets. A TC Data Unit will contain a TC Segment Header, which consists of a 1 octet header and up to a 250 octet segment data field.

3.3.1.1.2 TC Control Command

A control command (Figure 3-5) consists of only a control specifier which comes after the frame header (LRO does not use chained control qualifiers). TC Control Commands are used to specify to the spacecraft the governing Frame Acceptance Reporting Mechanism (FARM) parameters.

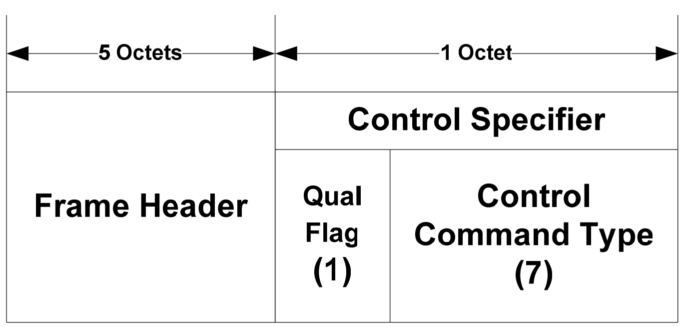

Figure 3-5. Telecommand Control Command

Table 3-2: Telecommand Control Command

Field Name	# of bits	Value	LRO Usage
Qualifier Flags	1	‘0’	This field will be set to "0" to indicate there are no control qualifier fields (they are not supported).
Control Command Type	7	‘0000000’ or ‘0000111’	The control command type is a seven bit field. This field is used to define what kind of control command is being sent. The only control commands supported are: "0000000" "UNLOCK" "0000111" "SET NEXT EXPECTED FRAME SEQUENCE NUMBER TO ZERO" The "UNLOCK" control command is used to "unlock" the Frame Acceptance and Reporting Mechanism (FARM) in the event the FARM enters a lockout state. The "SET NEXT EXPECTED FRAME SEQUENCE TO ZERO" Control Command is used to set the Next Expected Frame Sequence Number (V(R)) maintained by the FARM, to the value "all zeros".

3.3.1.2 "Frame Validation Check" Criteria

All frames must pass the Frame Validation Check. If a frame fails the validation check then the frame will be discarded. The recommended CCSDS Frame Validation Check will be performed:

a. TC Frame must have the expected VERSION NUMBER.

b. TC Frame must have the expected SPACECRAFT ID.

c. TC Frame header must not contain any states that are not consistent with the TC Frame values defined above.

d. TC FRAME LENGTH must be consistent with the number of octets that are present.

e. A TC control command must only be UNLOCK or SET NEXT EXPECTED FRAME SEQUENCE NUMBER TO ZERO.

3.3.1.3 Communications Operation Procedure (COP)

LRO will support COP-1. COP-1 is a closed loop telecommand protocol that uses sequential ("go-back-n") re-transmission techniques to correct TC Frames that were rejected by the spacecraft because of an error. A full description of COP-1, including FOP state machine tables, is provided in CCSDS 232.1-B-1.

The FARM SLIDING WINDOW will be a fixed size of W=126 transfer frames. The FARM POSITIVE and NEGATIVE EDGE will be set at W/2=63, where the NEGATIVE EDGE is the lower bound and is the maximum amount less than the expected command value and the POSITIVE EDGE is the upper bound which is the maximum amount greater than the expected command value.

The status of COP-1 will be reported back in the Command Link Control Word (CLCW), refer to section 4.3.1.2.3.1 for details.

3.3.1.4 Virtual Channel Assignment

The FARM only needs to support a single virtual channel. All normal commanding activities will take place on VC1. In this mode of operation the VC ID in the Transfer Frame (TF) Header will be set to "000001", and the Bypass Flag will be set to "0". When the Control Command Flag in the TF header is set to "0" the data field of the transfer frame will contain a Segmentation Header followed by a Command Packet, which contains the desired application command. When the Control Command Flag is set to "1" the frame data field will contain a Control Specifier, which then contains the desired Control Command.

3.4 SEGMENTATION LAYER

The segmentation layer is one method which allows commands longer than one transfer frame to be transmitted. For the Segmentation Layer, LRO will not support command packets segmented across transfer frames. However the ground will format the segmentation fields properly as they will be validated on the orbiter.

3.4.1 Telecommand Segment

A TC Segment represented in Figure 3-6 is made up of a segment header and a segment data field.

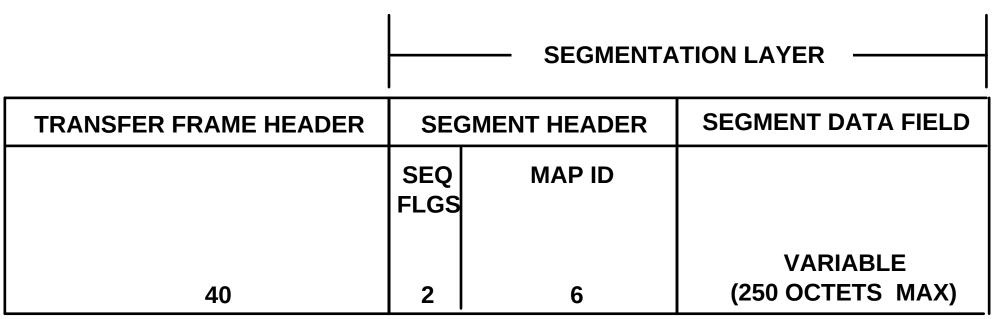

Figure 3-6. Telecommand Segment

Table 3-3: Telecommand Segment

Field Name	# of bits	Value	LRO Usage
Sequence Flags	2	‘11’	Indicate an unsegmented user data unit.

Map ID	6	'000001'	Indicate multiplexing is not used.
TC Data Packet	Up to 250 Octets	Variable	TC Packets are defined in Section 3.5.

3.5 PACKETIZATION LAYER

3.5.1 Telecommand Packet

The TC packet is the only data structure of the Packetization Layer. The TC packet is similar to the telemetry packet. The general CCSDS packet structure is the same for the primary and secondary headers as shown in Figure 3-7. Note that LRO will implement a single command packet per telecommand frame.

The length of TC packets is limited to 250 octets to prevent a single transmission from monopolizing the command link. Though optional under CCSDS, LRO TC packets will always use a secondary header. The command packet is made up of three major fields: a primary header, a secondary header, and application data, sized as follows:

Primary Header	48 bits	(6 bytes)
Secondary Header	16 bits	(2 bytes)
Application Data	1936 bits max	(242 bytes max)

Primary header											
Packet Identification				Packet Sequence Control		Packet Length	Secondary Header			Packet Data
Version (3)	Type (1)	Sec Hdr Flag (1)	Application ID (11)	Segment Flag (2)	Source Sequence Count (14)	Pkt Length (16)	Reserved (1)	Function Code (7)	Checksum (8)	Application Data

Figure 3-7. Telecommand Packet

Table 3-4: Telecommand Packet

Field Name	# of bits	Value	LRO Usage
Version No.	3	'000'	Indicate CCSDS version 1 packets are used.
Type	1	'1'	Indicate it is a telecommand and not a telemetry packet.
Secondary Header Flag	1	'1'	Indicate the presence of the secondary header.

Field Name	# of bits	Value	LRO Usage
Application ID	11	Variable (0 to 2047)	The 11-bit application process identifier (APID) field will be used to address commands to different subsystems of the spacecraft. Assignment of commands to APIDs is described in Volume II of this T&C Handbook.
Segment Flags	2	'11'	Indicate that TC packet segmentation is not supported by LRO.
Source Sequence Count	14	0	This field typically increments with each new command but is not used for LRO.
Packet Length	16	Variable up to 243	The 16-bit field packet length is a count, in octets, of the application data field length in octets minus one. It is measured from the first bit of the secondary header to the last bit application data.
Reserved	1	'0'	Indicate that it is a non-CCSDS defined type header.
Function Code	7	0-127	This will contain a function code to provide a numerical identifier for an individual command. Assignment of function codes is provided in Volume II of this T&C Handbook.
Checksum	8	Variable	A checksum is used to verify the integrity of the packet contents after transmission of a packet. The detailed algorithm for calculation of the checksum is described in section 3.5.1.1.
Application Data	Variable up to 242 bytes	Variable	The application data field will contain either command application data which typically includes data indicating a selected option (e.g., off, low, high), a set-to value, or other data necessary for command execution or CFDP PDUs which could be in the form of response or data PDUs.

3.5.1.1 Checksum

Before transmission, a checksum algorithm is applied to the contents of the packet, from the beginning of the primary header to the end of the application data, including the checksum bits, which are initialized to zero. The resulting checksum is inserted into the secondary header of the packet. Onboard, a checksum is derived by starting with the value 0xFF and then performing an XOR with the first byte of the packet. The result is XOR'ed with the second byte, and so on. The resultant checksum should produce a value of 0. The algorithm to calculate the checksum value is exactly the same on the ground/sending end except that the checksum field is initialized with zeroes. The following checksum algorithm is used:

```
uint8 CCSDS_ComputeCheckSum (CCSDS_CmdPkt_t *PktPtr)
{
uint16 PktLen = CCSDS_RD_LEN(PktPtr->PriHdr);
uint8 *BytePtr = (uint8 *)PktPtr;
uint8 CheckSum;

CheckSum = 0xFF;
while (PktLen--) CheckSum ^= *(BytePtr++);

return CheckSum;

} /* END CCSDS_ComputeCheckSum() */
```

If the two checksums match, there is a high probability that data integrity was retained during transmission. If the two checksums disagree, it is assumed that data corruption has occurred and the packet is retransmitted.

3.5.1.2 Application Data

3.5.1.2.1 Application Data for Spacecraft Commanding

Command application data typically includes data indicating a selected option (e.g., off, low, high), a set-to value, or other data necessary for command execution. For memory/table loads, the application data consists of a string of binary data. Application data procedures are described further in section 3.6.2.

3.5.1.2.2 Application Data for CFDP Commanding

CCSDS File Delivery Protocol (CFDP) is the primary method for uploading and downloading file data to and from the spacecraft. Details associated with overall concept and implementation is presented in the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078). In this section, the protocol breakdown is discussed with regards to how the CFDP application and structure fits into the commanding scheme. There are two options associated with CFDP ground to spacecraft commanding; file uploads and directives. In both instances and in general terms the CFDP unit of transfer is a protocol data unit (PDU). In either a file load or directive case, from a CCSDS telecommand (TC) perspective the CFDP PDU is treated as ordinary application data.

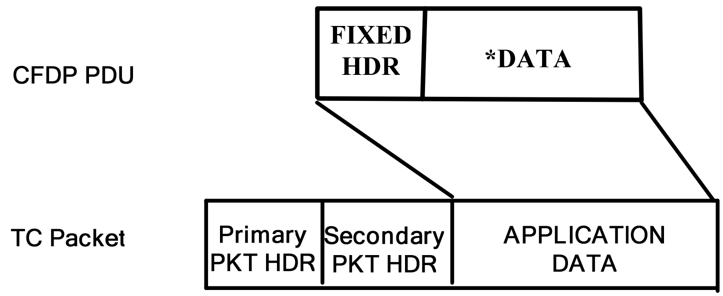

Figure 3-8. CFDP PDU as Command Application Data

The first item to point out is that a specific application identifier is assigned to the CFDP application so that the spacecraft and the ground can identify that CFDP is the mode of transfer. The APID is defined above in the TC packet header. The CFDP PDUs all follow the same basic structure; each has a fixed header attached to the user data contained within. Because the CFDP decomposition is the same whether the CFDP application is being used for telemetry or commands, it is presented in section 5.0. The main difference in the CFDP data between telemetry and commands is the actual file data itself. Command files are presented in section 3.6.2.2.5.

3.6 SYSTEM MANAGEMENT/APPLICATION LAYER

The application layer of the command architecture provides many functions to the user. These functions include: maintaining a data base of mission scheduling information, translating user requests into command directives (STOL), providing facilities to suspend, resume, or abort the transfer of commands, generating telecommand sessions, and generating control instructions to the receiving end. Many of the data formats for the sending (ground) end of the command architecture are implementation dependent and will not be addressed in this document.

3.6.1 Standard Data Types

All application data generated from the mission database (MDB) for building commands will be composed only of specific, pre-defined data types. Standard data types will be used to represent the major data types to facilitate the interchange of data between organizations.
Table 3-5 defines the data types applicable for LRO and maps the bytes from Most Significant Byte (MSB) to Least Significant Byte (LSB) order into their transmission order for each data type. The Octet Order column provides this mapping. The byte identifiers in this column (N | N+1, etc.) indicate the position of a byte in a given data type. The standard definition of byte ordering within any data type is defined in Table 3-5:

Table 3-5: LRO Data Types

MSB **LSB**

N	N + 1	N + 2	N + 3	N + 4	N + ...

Data Description	Type Name	Length	Octet Order
unsigned byte	UB	1	N
signed byte	SB	1	N
unsigned word	UI	2	N | N+1
unsigned swapped word	UI386	2	N+1|N
signed word	SI	2	N | N+1
unsigned longword	ULI	4	N+2 | N+3 | N | N+1
unsigned unswapped longword	ULI320	4	N|N+1|N+2|N+3
signed longword	SLI	4	N+2 | N+3 | N | N+1
single precision real	SFP	4	N+2 | N+3 | N | N+1
double precision real	DFP	8	N+6 | N+7 | N+4 | N+5 | N+2 | N+3 | N | N+1
CUC (no pre, sec 1, subsec 2)	TIME12	3	N | N+1 | N+2
CUC (no pre, sec 4, subsec 4)	TIME44	6	N | N+1 | N+2 | N+3 | N+4 | N+5| N+6 | N+7
CUC (no pre, sec 4, subsec 2)	TIME42	6	N | N+1 | N+2 | N+3 | N+4 | N+5
CUC(no pre, sec 4, subsec 0)	TIME40	4	N|N+1|N+2|N+3
Strings	STRING	Variable	N|N+1|N+2|N+3|N+4|N+…

3.6.1.1 Integer Data
Integer binary data will be formatted as 8, 16, or 32 bit unsigned or signed (2's complement) fields.

3.6.1.1.1 Signed Integer
Signed binary integer data fields will store the sign in the most significant bit of the most significant byte for the field. When signed data values are defined by a non-octet-integral number of bits, the sign bit will be extended up to the next largest 8, 16, or 32-bit length.

3.6.1.1.2 Unsigned Integer
Unsigned binary integer fields will be zero extended up to the next larger 8, 16, or 32-bit length.

3.6.1.2 Floating Point Data
LRO will use the typical IEEE-754 standard for real numbers.

3.6.1.3 Time Code Data
All time codes (TIME42, TIME40, TIME12) will be formatted to correspond to a subset of the CCSDS Unsegmented Time Code (CUC) as defined in CCSDS 301.0-B-3. For LRO, no more than a four-octet seconds field and a two-octet subseconds field will be used. Additionally, individual spacecraft data sources may not require the full range provided by all six bytes. These may have the seconds and/or the subseconds truncated on word boundaries, but only in one of the combinations specified in Table 3-5. Note that TIME40 and TIME42 are used for Absolute Time. TIME12 is used for Relative Time only.

3.6.2 Standard Procedures

To uniquely identify each command, which will execute only one event, an APID/function code combination must be specified.

3.6.2.1 Command Identification

Each APID will typically be associated with multiple function codes, though only one command can be sent in each command packet. The same function code values (e.g., 01, 02, 03, etc.) are typically used for multiple APIDs. To uniquely identify a command, a combination of both the APID and function code must be specified.

3.6.2.2 Spacecraft Commands

Each APID/function code combination will refer to a specific command packet of fixed format and length, and will be identified by a defined command mnemonic. The application data in any command packet can consist of only those data types defined in Table 3-5. A database will be maintained by the LRO project containing the specification records that define the command template for each mnemonic. All mnemonics will be generated using standardized naming conventions. The naming conventions and MDB are described in LRO Mission Telemetry and Command Database Style Guide (431-HDBK-000076).

3.6.2.2.1 Data Field Structuring

Standard octet order, bit order, and data alignment procedures will be used to facilitate the transfer of data between different computer architectures.

3.6.2.2.2 Octet Ordering

The octet order of transmission for the different data types is defined in Table 3-5. The first octet transmitted is referred to as N, with each subsequent octet numbered one greater than the previous octet. It is the responsibility of the ground to do any necessary byte swapping.

3.6.2.2.3 Bit Ordering

The CCSDS numbering convention for bit ordering will be used. The first bit in a data field is defined to be "bit 0". For a data field X bits in length, bit 0 will specify the most significant bit (2^{X-1}) of that data field, while bit X-1 will specify the least significant bit (2^0). For signed data fields, the sign bit will be the most significant bit specified for the data field.
This nomenclature applies to a data field prior to being byte-swapped as described in the previous section.

3.6.2.2.4 Data Alignment

The first octet of the packet will be referred to as octet zero. All subsequent octets in the packet will be referred to incrementally. Data types in the data field will be placed contiguously. The leading bit of any defined data type field will begin on an octet boundary. Bit sub-fields within a data type used for specific telemetry values may begin anywhere in an octet.

3.6.2.2.5 File Loads

Description of file loads (memory and table) structures are described in section4.6.

3.7 ADDITIONAL COMMAND TRANSMIT STRUCTURES

A transmit header is attached to the entire command structure by the Telemetry and Command system and is used by the station front ends, then stripped before forwarding the command to the spacecraft. There are two types of structures used based upon the network selected, a ground generated CLTU with an attached ground message header based upon EOS defined standards is used for the commercial network and the white sands ground station. For the DSN the ground generated CLTU will be formatted into a Service Link Extension (SLE) Forward Command Link Transmission Unit (F-CLTU). The details associated with the field definitions for the transmit structures are documented in the Lunar Reconnaissance Orbiter Ground System Interface Control Document (431-ICD-000049).

4.0 TELEMETRY

The LRO implementation of telemetry capabilities is a subset of what the CCSDS AOS and packet standards provide (CCSDS 133.0-B-1 and CCSDS 132.0-B-1). The size and scope of LRO does not warrant the use of all of the options provided in the CCSDS AOS standard. The complete telemetry data structure overview is shown in Figure 4-1 and will be decomposed into its specific elements through the rest of this section.

S-Band telemetry is used on the LRO spacecraft for transmitting orbiter real-time housekeeping and engineering telemetry and memory and table files to the ground. As a contingency the S-Band could be used to downlink recorded housekeeping, engineering and science telemetry. The Ka-Band is used primarily to downlink housekeeping, engineering, and measurement data files. The virtual channel assignments for LRO are identified in Table 4-1. Note that a VC is associated with either the S-band or Ka-band link, except for VC0 which can be downlinked in either. All file types can be sent down in either the S-band or Ka-band but the appropriate VC must be selected.

Table 4-1: LRO Virtual Channel Assignment

LRO CCSDS Virtual Channels	
Telemetry	Virtual Channel ID (decimal)
Real Time Telemetry S-Band (& contingency Ka-band)	0
Stored Telemetry/files downlinked using the S-Band system	1
Stored Telemetry/files downlinked using the Ka-Band system – Primarily housekeeping files	2
Stored Telemetry/files downlinked using the Ka-Band system – Primarily Science Files	3
Fill Frames	63

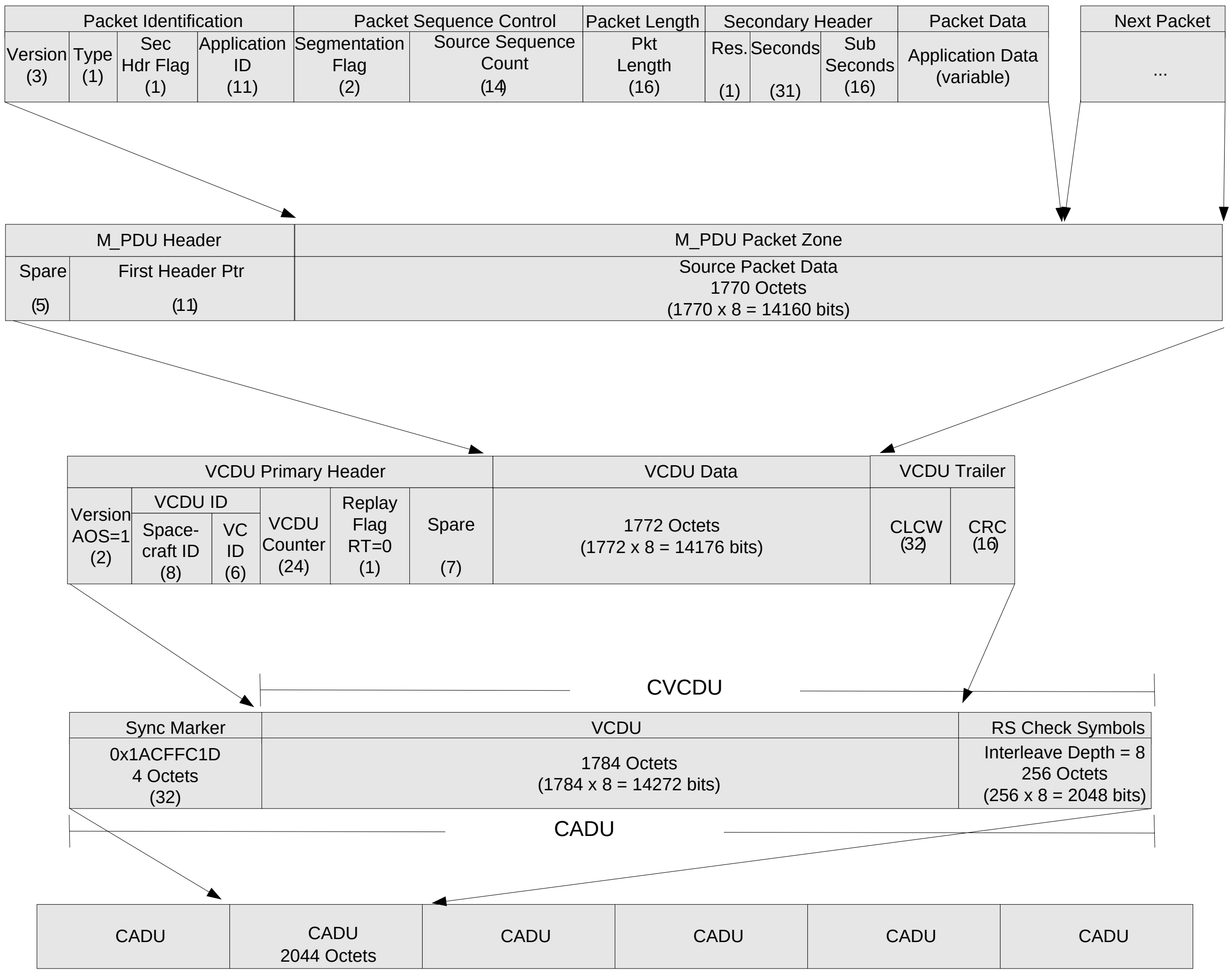

Figure 4-1. LRO Telemetry CCSDS Data Structures Overview (value in bits)

4.1 PHYSICAL LAYER

The physical layer provides the physical connection between the spacecraft and the ground station using radio frequency signals. For signal formatting, modulation, and RF characteristics refer to the Lunar Reconnaissance Orbiter Space-to-Ground Radio Frequency Interface Control Document (451-RFICD-LRO/GN/DSN/SN).

4.2 CODING LAYER

The coding layer reduces the probability of bit errors. The coding layer allows less energy to be expended per information bit. The coding layer uses error correcting codes to provide these services.

4.2.1 Convolutional

LRO will use the CCSDS recommended 1/2 rate; constraint length seven with alternate symbol inversion. The characteristics of the CCSDS convolutional code are:

(1)	Nomenclature:	Convolutional code with maximum-likelihood (Viterbi) decoding
(2)	Code rate:	1/2 bit per symbol
(3)	Constraint length:	7 bits
(4)	Connection Vectors:	G1 = 1111001; G2 = 1011011
(5)	Phase Relationship:	G1 is associated with first symbol
(6)	Symbol inversion:	On output path of G2

4.2.2 Reed-Solomon

Reed-Solomon (R-S) encoding provides the capability to detect errors in the telemetry transfer frame and correct them. Concatenating the R-S code with the convolutional code will provide a significant improvement in the error performance of the channel. The Reed-Solomon code will be generated using the following parameters:

J = 8 bits per R-S symbol.
E = 16 R-S symbol error correction capability within an R-S code word.
I = 8 for the depth of interleaving.

The encoding scheme follows the CCSDS Recommendation for Telemetry Channel Coding specified in section 4 of CCSDS 131.0-B-1. The generation of the R-S will be over the transfer frame including the final 16-bit parity field and excludes the attached sync marker.

4.2.3 Pseudo-Randomization

As with commands, pseudo-randomization will be applied to the telemetry using CCSDS defined standards as specified in section 7 of CCSDS 131.0-B-1.

4.3 TRANSFER LAYER

The transfer layer provides a reliable error-controlled transfer of data through the noisy channel. All the data structures of the transfer layer will be formatted per CCSDS telemetry data standards.

The CCSDS numbering convention for bit ordering will be used. The first bit in a data field is defined to be "bit 0". For a data field X bits in length, bit 0 will specify the most significant bit (2^{X-1}) of that data field, while bit X-1 will specify the least significant bit (2^0). In the data structures defined in the following sections, the order of transmission will start from the left side of the figures.

4.3.1 Channel Access Data Units (CADU)

The CADU is a fixed size data unit of 2044 octets consisting of a Coded Virtual Channel Data Unit (CVCDU) with the preceding synchronization marker.

<table>
<tr><td colspan="3">CADU – 2044 Octets</td></tr>
<tr><td>Sync Marker</td><td colspan="2">CVCDU – 2040 Octets</td></tr>
<tr><td rowspan="2">0x1ACFFC1D
(32)</td><td>VCDU</td><td>RS Check Symbols</td></tr>
<tr><td>1784 Octets
(1784 x 8 = 14272 bits)</td><td>Interleave Depth = 8
256 Octets
(256 x 8 = 2048 bits)</td></tr>
</table>

Figure 4-2. Channel Access Data Unit

4.3.1.1 Frame Synchronization Marker

The frame synchronization marker used will be the CCSDS recommended 32-bit marker:

0001 1010 1100 1111 1111 1100 0001 1101
^ ^
Bit 0 Bit 31

The hexadecimal notation is: 1ACFFC1D.

4.3.1.2 Coded Virtual Channel Data Unit (CVCDU)

The CVCDU is a fixed size data unit of 2040 octets and consists of a Virtual Channel Data Unit (VCDU) (1784 octets) and the Reed Solomon Check symbols (2048 bits or 256 octets).

4.3.1.2.1 Reed-Solomon

The Reed-Solomon check symbols are a 256 octet field applied to each VCDU to create the CVCDU. Reed-Solomon is described in section 4.2.2

4.3.1.2.2 Transfer Frame VCDUs

The transfer frame VCDU is made up of the primary header, data field, and trailer. The length of the Transfer Frame components will be 14,272 bits (1784 bytes), allocated among the Transfer Frame components as follows:

Primary Header	48 bits	(6 bytes)
Data Field	14176 bits	(1772 bytes)
Trailer	48 bits	(6 bytes)

<table>
<tr><td colspan="6">VCDU Primary Header</td><td rowspan="3">VCDU Data

1772 Octets
(1772 x 8 = 14176 bits)</td><td colspan="2">VCDU Trailer</td></tr>
<tr><td rowspan="2">Version
AOS=1
(2)</td><td colspan="2">VCDU ID</td><td rowspan="2">VCDU
Counter
(24)</td><td rowspan="2">Replay
Flag
RT=0
(1)</td><td rowspan="2">Spare

(7)</td><td rowspan="2">CLCW
(32)</td><td rowspan="2">CRC
(16)</td></tr>
<tr><td>Space-
craft ID
(8)</td><td>VC
ID
(6)</td></tr>
</table>

Figure 4-3. Virtual Channel Data Unit

Table 4-2: VCDU Primary Header Fields

Field Name	# of bits	Value	LRO Usage
Version	2	'01'	Indicate use of CCSDS AOS frame
Spacecraft ID	8	'11111111'	The LRO Telemetry spacecraft ID is defined in section 2.0, 0xFF
VC ID	6	0,1,2,3,63	The Virtual channel ID are defined in Table 4-1
VCDU Counter	24	Variable	This counter is incremented each time a frame from the same VC is transmitted from the spacecraft to the ground. The counter will simply roll over when it reaches its maximum count. It is used by the ground to verify that it is receiving sequential frames for the corresponding VC
Replay Flag	1	'0'	This flag to identify retransmitted VCDUs. This flag is not currently used in the LRO implementation.
Spare	7	'0000000'	Not currently used in the LRO

4.3.1.2.3 VCDU Trailer

The VCDU trailer is a 6 octet trailer that contains the CLCW and the CRC.

Table 4-3: VCDU Trailer

Field Name	# of bits	Value	LRO Usage
CLCW	32	Table 4-4	Command Link Control Word refer to section 4.3.1.2.3.1
CRC	16	Variable	Hardware inserts the CRC refer to section 4.3.1.2.3.2

4.3.1.2.3.1 Command Link Control Word (CLCW)

A CLCW is downlinked by the spacecraft in every telemetry transfer frame trailer. The CLCW is a four octet data structure shown in Figure 4-4. The CLCW will be attached to all telemetry VCDUs, though it will only be valid in the VC0. For all other VCs the flight software will apply a static value for the CLCW. The breakdown of the fields is described in Table 4-4.

Ctrl Word Type	CLCW Ver.	Status Field	COP	Virtual Channel ID	Spare	FLAGS					FARM B Count	Report Type	Report Value
						No RF	No Lock	Lock-out	Wait	Re-xmit			
(1)	(2)	(3)	(2)	(6)	(2)	(1)	(1)	(1)	(1)	(1)	(2)	(1)	(8)

Figure 4-4. Command Link Control Word

Table 4-4: Command Link Control Word

Field Name	# of bits	Value	LRO Usage
CTRL Word Type	1	‘0’	For VC0 this bit will be “0” to indicate this is a CLCW in the telemetry transfer frame. For VC1, VC2, VC3 and VC63, this bit will be set to "1".
CLCW Version	2	‘00’	Indicates version number 1 CLCW is used.
Status Field	3	‘000’	The three bit status field will not be used by LRO.
COP	2	‘01’	Indicate that COP-1 is being used.
VC Identifier	6	1	The VCID will be set to "1" to indicate the command telecommand VC.
Spare	2	‘00’	The two bit field will not be used by LRO.
No RF	1	0-1	For VC0 this bit will indicate whether or not the transponder has carrier lock or not. A "0" indicates that carrier. A "1" indicates that the no carrier. For VC1, VC2, VC3 and VC63, this bit will be set to "0".
No Lock	1	0-1	For VC0 this bit field will indicate the status of the comm card, whether or not it detects an idle pattern. If the comm. card is in bit lock then this flag will be set to "0". If the comm. card is not in lock then it will be set to "1" to indicate that the RF channel is not available. For VC1, VC2, VC3 and VC63, this bit will be set to "0".
Lockout	1	0-1	For VC0 the lockout flag will be set to "1" to indicate lockout condition has been entered. Otherwise the flag will be set to "0". For VC1, VC2, VC3 and VC63, this bit will be set to "0".
Wait	1	0-1	The WAIT flag will not be used by LRO and will always be set to zero.
Re-Transmit	1	0-1	For VC0 the RETRANSMIT flag will be set to "1" to indicate to the Frame Operation Procedure (FOP) a need

Field Name	# of bits	Value	LRO Usage
			to re-transmit one or more frames. If the flag is set to "0" then there are no outstanding frame rejections. For VC1, VC2, VC3 and VC63, this bit will be set to "0".
FARM B Count	2	Variable	For VC0 this field will contain the least two significant bits of the FARM-B COUNTER. This is used to verify Type-B (bypass mode) TC transfer frames and Control Commands. For VC1, VC2, VC3 and VC63, this bit will be set to "0".
Report Type	1	‘0’	This field is used only by COP-2.
Report Value	8	Variable	For VC0 this 8-bit field contains the current observed value of the FARM's NEXT EXPECTED FRAME SEQUENCE NUMBER (V(R)). For VC1, VC2, VC3 and VC63, this bit will be set to "0".

4.3.1.2.3.2 Frame Error Control Calculation

The Cyclical Redundancy Check (CRC) is a 2 octet field in the VCDU trailer that provides the capability to detect errors in the telemetry VCDU. LRO will follow the CCSDS recommendations for frame error control as specified in section 4.1.6 of CCSDS 732.0-B-1. In general, the polynomial used to generate the CRC will be:

$$g(x) = x^{16} + x^{12} + x^{5} + 1$$

Both the encoder and decoder are initialized to the "all-ones" state for each VCDU.
The generation of the CRC will be over the VCDU (less the final 16-bit parity field) and excludes the attached sync marker.

4.3.1.2.4 VCDU Data Field

LRO uses a VCDU data field with a size of 1772 octets. The VCDU data field will contain the M_PDU header and the M_PDU packet zone as shown in Figure 4-5.

M_PDU Header		M_PDU Packet Zone
Spare (5)	First Header Ptr (11)	Source Packet Data 1770 Octets (1770 x 8 = 14160 bits)

Figure 4-5. VCDU Data Field

4.3.1.2.4.1 Multiplexing Protocol Data Unit (M_PDU)

The M_PDU header structure is used to allow packets to span VCDU frames. It contains a 2 octet header that contains a pointer to the first packet header in the M_PDU packet zone. Table 4-5 describes the usage of the MPDU data fields. The M_PDU packet zone contains the CCSDS telemetry packets.

Table 4-5: M_PDU Fields

Field Name	# of bits	Value	LRO Usage
Spare	5		Not used
First header pointer	11		An octet pointer to the first CCSDS Header in the MPDU Packet Zone. If zero indicates that the first packet header is at the first octet in the M_PDU in the Packet Zone. If there is no packet header in the frame this field will be set to 0x7FF. If the data in this frame is all fill packets or there is no packet data in the frame this field will be set to 0x7FE.
Packet Zone	14112		Source packet data

4.3.1.2.4.2 Telemetry Packets

LRO telemetry data will be transferred in CCSDS telemetry packets. LRO Telemetry Packets will contain primary and secondary headers. LRO has two types or telemetry packet structures, one associated with real-time data and one associated with playback data. The general CCSDS packet structure is the same and described as shown in Figure 4-6.

Primary Header	48 bits	(6 bytes)
Secondary Header	48 bits	(6 bytes)
Application Data	65536 bits max	(8192 bytes max)

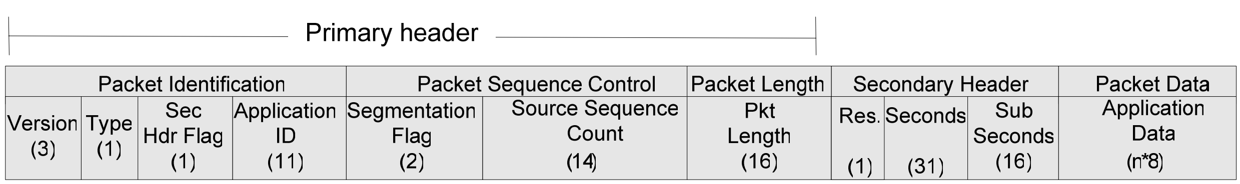

Figure 4-6. CCSDS Telemetry Packet Structure

4.3.1.2.4.2.1 Primary Header

The primary header is divided in into three major fields: the packet identification, packet sequence control and packet length field. All subfield use and restrictions are described in Table 4-6.

Table 4-6: Telemetry Source Packet Primary Header

Field Name	# of bits	Value	LRO Usage
Version No.	3	'000'	Will be B'000' for LRO Telemetry Packets
Type Indicator	1	'0'	0 identifies telemetry packets
Secondary Header Flag	1	'1'	For LRO Packets this field will always be a "1" since the secondary header is mandatory.
Application Identifier	11	Variable (0 to 2047)	LRO Application Identifiers assignment is in Appendix A APID 2047 will only be used per CCSDS recommendations for fill data packets.
Segmentation Flags	2	'11'	Will be B'11' to identify unsegmented packets. Segmented packets are not allowed for LRO.
Source Sequence Count	14	0-16383	LRO Senders of packets will increment this number by 1 for each copy of the packet created.
Packet Length	16	Variable up to 65528	Length field = Length of Packet in octets – Length of primary header in octets (6 octets) –1 The length of LRO packets will be an even number of octets (this means that the length field will be an odd number).

4.3.1.2.4.2.2 Secondary Header

Table 4-7: Telemetry Source Packet Secondary Header

Field Name	# of bits	Value	LRO Usage
Reserved	1	'0'	Indicates non-standard header
Seconds	31	0-7FFFFFFF	Seconds since epoch
Subseconds	16	0-65535	Subseconds

LRO Source packets will contain one reserved bit and a 47 bit time stamp field as the secondary header. The 47-bits correspond to 31-bits of seconds and 16-bits of sub-seconds, both in relation to the amount of time since epoch. The epoch time for LRO is January 1, 2001 00:00:00. The LRO Time Code epoch conforms to the CCSDS recommendations for Time Code epochs. For all data, the time code in the packet secondary header will be generated by the LRO spacecraft and will place a time code in the packet at the time it is generated.

4.3.1.2.4.2.3 Telemetry Packet Data Field

4.3.1.2.4.2.3.1 Real-time telemetry

The telemetry packet data field is where there is significant difference between the real-time and file data. For real-time data the telemetry packet data field is user defined and consists of a series of sequential telemetry points. All telemetry points will always be at the same (octet) offset from the beginning of a packet if they are present in the packet. This is necessary because on-board telemetry monitoring is done on the basis of a telemetry points offset and size. No sub commutation of telemetry points in a packet is allowed.

File data formats are discussed in section 4.6.

4.3.1.2.4.2.3.2 CFDP Telemetry

CCSDS File Delivery Protocol (CFDP) is the primary method for uploading and downloading file data to and from the spacecraft. Details associated with overall concept and implementation is presented in the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078). In this section, the protocol breakdown is discussed with regards to how the CFDP application and structure fits into the telemetry scheme. There are two options associated with CFDP spacecraft to ground operations; file downloads and directives. In both instances and in general terms the CFDP unit of transfer is a protocol data unit (PDU). In either a file download or directive case, from a CCSDS telemetry packet perspective the CFDP PDU is treated as ordinary application data.

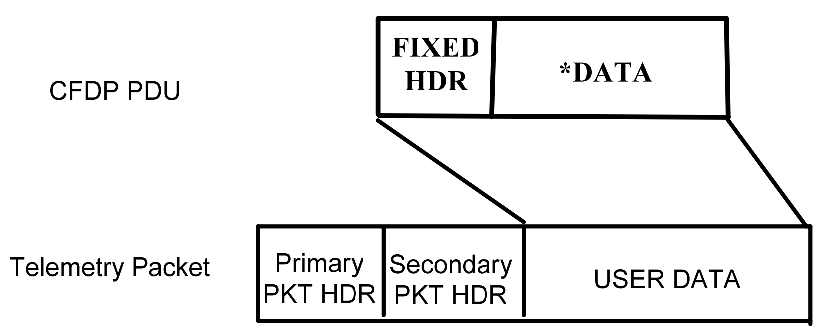

Figure 4-7. CFDP PDU as Telemetry Application Data

The first item to point out is that a specific application identification is assigned to the CFDP application so that the spacecraft and the ground can identify that CFDP is the mode of transfer. The APID is defined above in the Telemetry packet header. The CFDP PDUs all follow the same basic structure; each has a fixed data header attached to the user data contained within. Because the CFDP decomposition is the same whether the CFDP application is being used for telemetry or commands, it is presented in section 5.0. The main difference in the CFDP data between telemetry and commands is the actual file data itself.

4.3.2 Standard Procedures

4.3.2.1 ApID to Virtual Channel Mapping

Packet application identifiers (ApIDs) for LRO are documented in LRO Telemetry and Command Database Handbook Volume 2 (431-HDBK-000053). All real-time packets are forwarded in VC0. All other VCs utilize specific LRO file formats and are transported between the spacecraft and the ground using CCSDS File Delivery Protocol (CFDP), refer to the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078).

4.3.2.2 Fill Frames

Fill frames may be necessary to guarantee the contiguous transmission of transfer frames to the ground. (Fill packets may also be used to complete a partially filled transfer frame. This is handled at the packet layer.) Virtual channel 63 will be used for fill frames, which will follow the standard data structures of a transfer frame, including use of primary and secondary header. A fill frame's data field will contain data consisting of a 256 byte, repeating, base $_{16}$ pattern (i.e. 00, 01, 02, 03 ... FF).

4.3.2.3 Frame Count Generation

4.3.2.3.1 VC Counter

The VC count value is incremented at the time of the downlink.

4.4 SEGMENTATION LAYER

The LRO spacecraft C&DH subsystem does not use packet segmentation.

4.5 SYSTEM MANAGEMENT/APPLICATION LAYERS

The system management layer of the telemetry system on the spacecraft translates the sensor data into process-interpretable telemetry data. It puts the data into units suitable for packets. It also schedules the collection of the data for building of the packets. On the ground the system management layer can be viewed as delivering the packets to the telemetry application process. The Application layer on the spacecraft collects the telemetry for the users from the sensor’s application process. While on the ground the application layer provides the interfaces to the user to allow interpretation of the data. The application layer converts the data into understandable formats, i.e. engineering units, graphs, charts, etc.

Mnemonics are an alias for each telemetry point (parameter). Mnemonics will be used to allow the user to define the conversion of the telemetry to interpretable form and to define the layout of the packet data field.

Any spacecraft data, including science data, that is to be processed by NASA/GSFC will adhere to the structures defined in this section. If the data is strictly for science processing only by the IGSE then the contents of the data packets may deviate from these standards in this section as desired. The responsibility of processing this data is then left to the science team.

4.5.1 Standard Data Types

Standard data types will be used to represent the major data types to facilitate the interchange of data between organizations. Table 3-5 defines the data types applicable for LRO.

4.5.2 Standard Procedures

4.5.2.1 Definition of Telemetry Packet Source Data

4.5.2.1.1 Packet Data Field Formats (Mnemonics)

All parameters will be defined using mnemonics to identify the packet data fields to be extracted. The total number of octets to be extracted for each mnemonic will be implicit by the data type (see Table 3-5). A database will be maintained by the LRO project containing the specification records that define each mnemonic. The project data base (PDB) is described in LRO Telemetry and Command Database Management Plan (431-PLAN-000051). All mnemonics will be generated using a standardized naming convention defined in the LRO Mission Telemetry and Command Database Style Guide (431-HDBK-000076).

4.5.2.2 Data Field Structuring

Standard octet order, bit order, and data alignment procedures will be used to facilitate the transfer of data between different computer architectures.

4.5.2.2.1 Octet Ordering

The octet order of transmission for the different data types is defined in Table 3-5. It is the responsibility of the ground to do any necessary byte swapping.

4.5.2.2.2 Bit Ordering

The CCSDS nomenclature for bit ordering will be used as described in section 3.6.2.2.3.

4.5.2.2.3 Data Alignment

The first octet of the packet will be referred to as octet zero. All subsequent octets in the packet will be referred to incrementally. Data types in the data field will be placed contiguously. The leading bit of any defined data type field will begin on an octet boundary. Bit sub-fields within a data type used for specific telemetry values may begin anywhere in an octet.

4.6 LRO Data Files

The LRO spacecraft utilizes several file formats for different data types. File formats will either be discussed below or the appropriate documentation will be referenced.

4.6.1 Overview

The Flight Software (FSW) or the Instrument Manager (IM) handles opening, writing, and closing all files. All instruments except LROC have two types of files, science and housekeeping. The science and housekeeping files are always opened and closed together. All science files have the extension ".sci" and all housekeeping files will have the extension ".hk". The default file name and maximum size for each file type is maintained in a loadable software table. The instrument manager writes a standard file header to each file after opening but before writing any instrument data, whereas the FSW handles these operations for all other file types. The LROC instrument directly requests the opening and closing of its own science data files, so IM handles them differently (see the LROC Data ICD for details). However, LROC's housekeeping files are handled the same as the other instruments' data files.

4.6.2 File Names

The FSW maintains a file configuration table that specifies the default file name for each instrument's data files and all other associated data files. This name includes the full path and base filename, but no file extension (for instruments the file extensions are fixed to either ".hk", or ".sci"). These names may be up to 40 characters long, including the path. The instrument file names should include a series of up to seven number-sign (#) characters at the end of the filename prior to the file extension, which the instrument manager will replace with an incrementing file counter. After a cold reset, the file counter for each instrument will be set to 1. The file counter can also be set to a different value by ground command. Aside from the # characters, the filename can contain any sequence of letters, numbers, and underscores.

The file naming formats for memory, table, orbiter housekeeping and event files are specified in the ground commands that create the files.

4.6.3 File Sizes

The maximum allowable file size for downlink is 257 MB. Most instruments will want much smaller files, though. The Instrument Manager maintains a maximum size for each instrument file type in the file configuration table. When a file reaches its maximum size, IM closes that file, opens a new file, and increments the file count. Housekeeping and science files will both be closed/re-opened whenever either reaches its maximum size.

The Instrument manager also maintains a maximum file open time for each instrument. When an instrument's file has been open for the maximum allowed time, both science and housekeeping files will be closed and new files opened.

The Instrument Manager also has commands to close/re-open files. Operationally these can be used to begin new files on conditions other than size. For example, an instrument might want to create a separate file for a special calibration observation. In this case, the S/C operators would schedule IM_OPEN_FILE commands before and after the special observation.

The maximum allowable file size for uplink is 1 MB, which is to ensure that files can be loaded to the orbiter within a single contact period at the 4 Kb uplink rate.

4.6.4 Data File Headers

The FSW or the IM will add a 64 byte file header to each file type before the first byte of specific data. The data file header will have the following format as specified in Table 4-8.

Table 4-8: LRO Data File Header Format

Parameter	**Size (bytes)**	**Value**		**Note**
File Type ID	4	Sci 200 210 220 230 240 250 260	HK 201 211 221 231 241 259 261 TBR	 CRaTER Diviner LAMP LEND LOLA LROC (TBR) Mini-RF Orbiter Housekeeping
Spare	4	0		Alignment or Expansion
Start Time (seconds)	4	0-7FFFFFFF		Spacecraft time when the file was opened for writing
Start Time (sub-seconds)	4	0-FFFFFFFF		Spacecraft time when the file was opened for writing
Stop Time (seconds)	4	0-7FFFFFFF		Spacecraft time when the file was closed*
Stop Time (sub-seconds)	4	0-FFFFFFFF		Spacecraft time when the file was closed*
FileName	40	ASCII (with 0 padding)		Filename of the file as it appeared onboard the S/C (including path)

***Note:** It may not always be possible for the IM to fill in the stop time. The IM will fill these fields with 0s when the header is first written. If the file gets closed normally, the IM will rewind the file and overwrite the stop time with the current S/C time.

4.6.5 Spacecraft Housekeeping Data File Format

LRO spacecraft engineering data is stored on the recorder or in memory and the file structure consists of the file data header and then a series of data packets until the maximum file size has been reached or will be exceeded up to 1 MB (TBR).

4.6.6 Instrument Housekeeping File Format

Following the file header, housekeeping data files will be formatted as a series of CCSDS packets. The size and application ID of these housekeeping packets is specified in the individual Instrument Data ICDs as referenced in Table 4-9. The packets are stored back to back, with no added markers or padding. Most instruments have fixed-size housekeeping packets, so finding the packet boundaries is trivial. If the packets are variable size, however, this is more complicated. In this case, readers must follow the chain of packets from the beginning of the file, using the length field in each packet to find the beginning of the next packet.

Table 4-9: Instrument Data ICD References (TBR)

Instrument	Document Title	Document Number
CRaTER	CRaTER Instrument Data Interface Control Document	431-ICD-000104
Diviner	DLRE Data Interface Control Document	431-ICD-000105
LAMP	LAMP Instrument Data Interface Control Document	431-ICD-000106
LEND	LEND Instrument Data Interface Control Document	431-ICD-000107
LOLA	LOLA Instrument Data Interface Control Document	431-ICD-000108
LROC	LROC Instrument Data Interface Control Document	431-ICD-000109
Mini-RF	Mini-RF to Spacecraft Data Interface Control Document	431-ICD-000160

4.6.7 Instrument Science File Format

The format of each science file is unique to the instrument. The format of the science files is specified in the Instrument Data ICD as referenced in Table 4-9. However, all science files begin with the standard file header described in Table 4-8.

4.6.8 Core Flight Executive (cFE) File Headers

The cFE offers onboard software services, including time management, event handling, table management, file management, and network services. For orbiter memory, table and event files an associated 64 byte cFE file header will be appended to the beginning of each file. The file header will have the following format as specified in Table 4-10.

Table 4-10: LRO cFE File Header Format (TBR)

Parameter	Size (bytes)	Data Type	Value
cFE Content Type	4	ULI	

Type of Content Type	4	ULI	
Length of Primary Header	4	ULI	
Spacecraft ID	4	ULI	
Processor ID	4	ULI	
Application ID	4	ULI	
CreateTime (seconds)	4	ULI	
Create Time (sub-seconds)	4	ULI	
File Description	32	CHAR	

4.6.9 Memory File Format

LRO memory dumps are stored on the recorder and the file structure consists of the file data header and then the raw memory address range selected to be dumped to a file via spacecraft command.

Table File Format

LRO table dumps are stored on the recorder and the file structure consists of a file header and then the contents of the spacecraft table. The individual table files are defined in the LRO Command and Data Handling Flight Software User's Guide (431-HDBK-000471).

Orbiter Event File Format

Event messages are stored in free-form "raw" text for use in debugging spacecraft anomalies. This feature will be used early in the mission life-cycle. The event log file has a cFE header followed by events in binary CCSDS packet format.

Default File Configuration Table Contents

After a cold reset of the spacecraft processor, the instrument file configuration table will have the following contents:

Table 4-11: Default File Configuration Table Contents (TBR)

Inst.	File Header ID	Max Time	Max File Length	Base Filename	Appended File Extension
CRaTER Sci	200	10 min	10MB	/SDR1/CRATER/Crater######	.sci
CRaTER HK	201	10 min	50kB	/SDR1/CRATER/Crater######	.hk
DLRE Sci	210	10 min	3MB	/SDR1/DLRE/Diviner######	.sci

Inst.	File Header ID	Max Time	Max File Length	Base Filename	Appended File Extension
DLRE HK	211	10 min	100kB	/SDR1/DLRE/Diviner######	.hk
LAMP Sci	220	113 min	10MB	/SDR1/LAMP/LAMP######.	.sci
LAMP HK	221	113 min	5MB	/SDR1/LAMP/LAMP######	.hk
LEND Sci	230	113 min	5MB	/SDR1/LEND/LEND######	.sci
LEND HK	231	113 min	100kB	/SDR1/LEND/LEND######	.hk
LOLA Sci	240	113 min	20MB	/SDR1/LOLA/LOLA######	.sci
LOLA HK	241	113 min	10MB	/SDR1/LOLA/LOLA######	.hk
LROC hk	253	60 min	300kB	/SDR1/LROC/LROC######	.hk
Mini-RF Sci	260	30 min	100MB	/SDR1/MiniRF/MiniRF######	.sci
Mini-RF HK	261	30 min	100kB	/SDR1/MiniRF/MiniRF######	.hk
Table					
Memory					
Orbiter HK					
Orbiter Event					

4.7 ADDITIONAL TELEMETRY TRANSMIT STRUCTERS

A telemetry transmit header is attached to the entire VCDU by the station front ends and then forwarded to the Telemetry and Command and Data Processing systems. There are two types of headers used for S-band telemetry based upon the network selected, a telemetry header based upon SMEX/LEOT defined standards is used for the commercial network and the white sands ground station (WS1). The Service Link Extension (SLE) Return All Frames Service Specification will be used for telemetry traversing the Deep Space Network (DSN). There is a third type of messaging format that is used at WS1 only for the Ka-band data, which is the Cortex telemetry message format. The details associated with the field definitions for the telemetry transmit headers/messages are documented in the Lunar Reconnaissance Orbiter Ground System Interface Control Document (431-ICD-000049).

5.0 CCSDS FILE DELIVERY PROTOCOL

This section presents the formats of the CFDP File Delivery Unit (FDU) and the Protocol Data Units (PDU), as well as how CFDP is incorporated in both CCSDS AOS telemetry and telecommand. For specific details on operations concept, implementation and configuration refer to the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078) and associated CCSDS documentation, CCSDS 727.0-B-3, CCSDS 720.2-G-1, and CCSDS 720.0-G-2.

The CFDP FDU is the functional concatenation of a data file and related metadata. During a CFDP transaction every data file for LRO, science, engineering, table, housekeeping, and memory, will have an associated metadata file created containing information about the data file. The composite data file and metadata file make up the CFDP FDU. The FDU is more of a logical term and has no other formatting and constraints other than associating the metadata to the file data. Note that an FDU may consist of metadata only, where the metadata consists of a user operation, details described in the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078).

FDU	
Meta Data	Data File

Figure 5-1. File Delivery Unit

The CFDP FDU is decomposed into PDUs which the actual units of transfer exchanged between CFDP entities and, therefore, both their contents and their formats are defined. All PDUs consist of two components: the Fixed PDU Header and the PDU Data Field.

PDU	
Fixed PDU Header	Fixed or Variable Data File

Figure 5-2. Protocol Data Unit

Though PDUs are the low level transfer units for CFDP, they are treated as data with reference to both the CCSDS telemetry and command data structures utilized for LRO, as specified in the telemetry and command sections of this document. For summary purposes the general telemetry and command structures and where the CFDP PDUs are located are shown in the next two sections.

5.1 CCSDS Telecommands

From a CCSDS telecommand (TC) perspective the CFDP PDU is treated as ordinary user data. The CFDP application on the ground will create a metadata PDU and break the data file intended for upload to the orbiter into data PDUs, then for transit it will insert the PDUs into the TC structure's TC packet data field. There will be one PDU per TC packet and one TC packet per command frame. The orbiter, upon receipt of the CLTU, will deconstruct the TC to extract the CFDP PDU for processing.

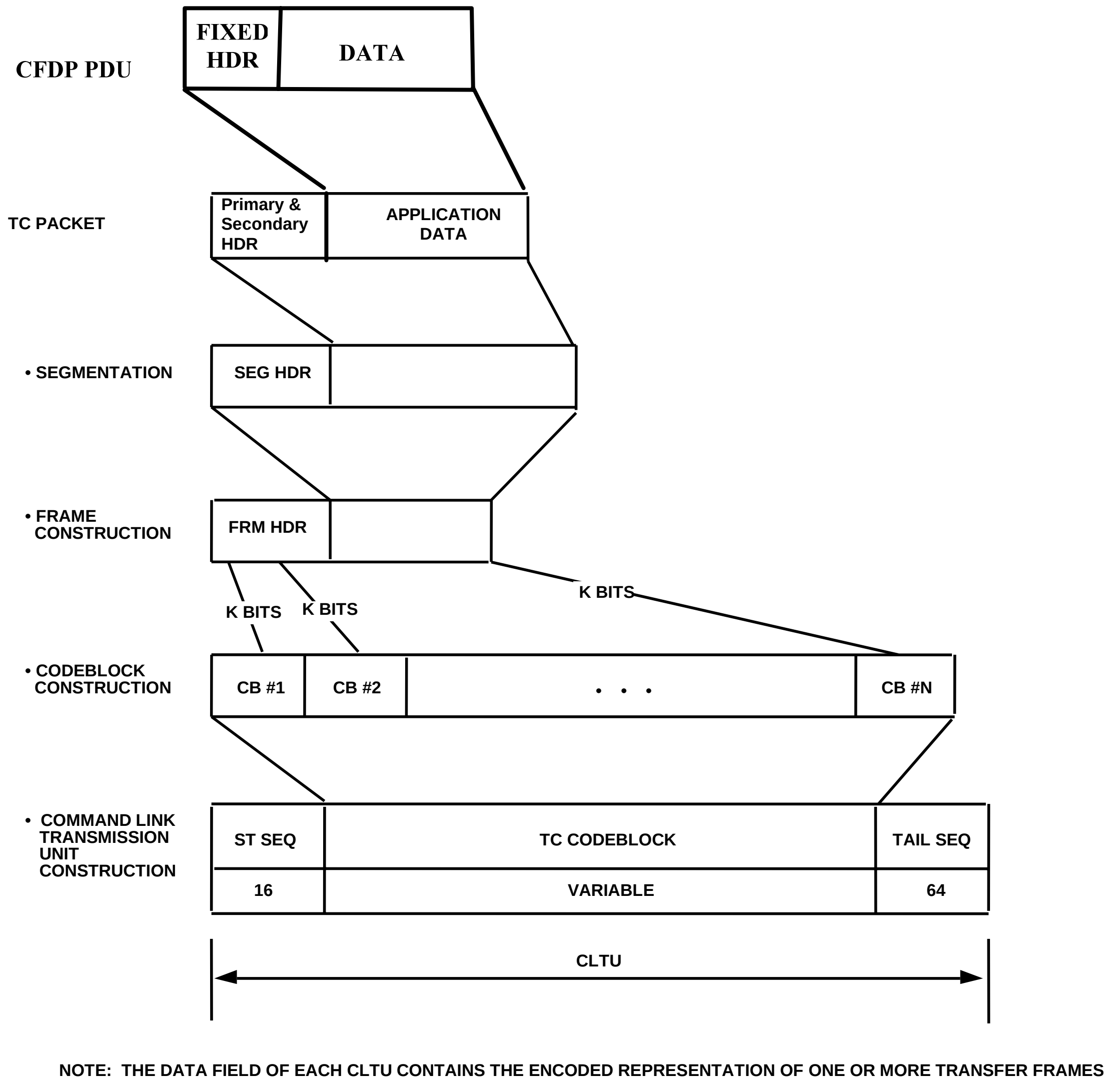

Figure 5-3. Summary Command Structure

5.2 CCSDS AOS Telemetry

Similar to the telecommand from a CCSDS AOS telemetry perspective, the CFDP PDU is also treated as ordinary user data. The CFDP application on the spacecraft will create a metadata PDU and break the data file intended for download to the ground into data PDUs, then for transit it will insert the PDUs into the telemetry packet. There will be one PDU per packet per VCDU. The data processing system on the ground, upon receipt of the VCDU, will deconstruct the VCDU to extract the CFDP PDU for processing.

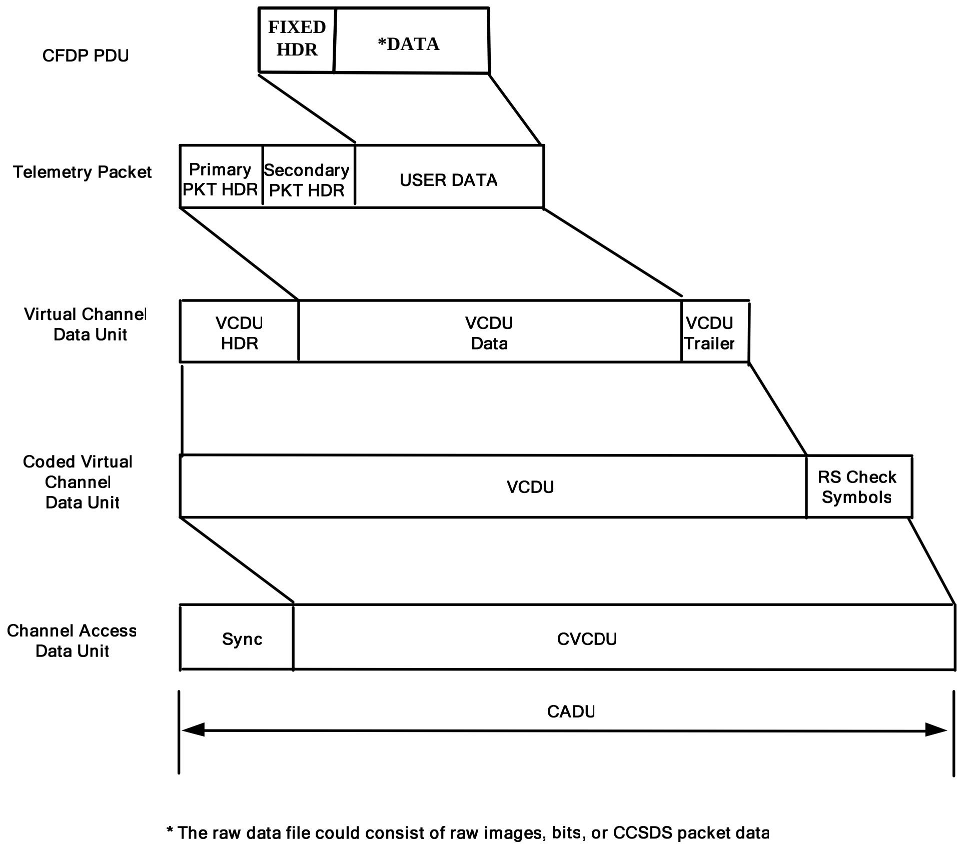

Figure 5-4. Summary Telemetry Structure

5.3 CFDP PROTOCOL DATA UNITS

As mentioned all PDUs consist of two components: the Fixed PDU Header and the PDU Data Field. The fixed data header is a 12 byte header. There are two types of PDUs, File Data and File Directive PDUs. The type of PDU is defined in the header under the field called PDU type.

Fixed PDU Header														PDU Data Field (Variable)
Version (3)	PDU Type (1)	Direction (1)	Xmit Mode (1)	CRC Flag (1)	Reserve (1)	Data Field Length (16)	Reserve (1)	Length of Entity ID (3)	Reserve (1)	Length of Transaction # (3)	Source Entity ID (16)	Transaction Sec. # (32)	Destination Entity ID (16)	

Figure 5-5. CFDP Fixed PDU Header

Table 5-1: Fixed PDU Header Fields

Field	Length (bits)	Values	**LRO Usage**
Version	3	’000’	For the first version.
PDU type	1	0-1	Discussed in Sections 5.3.1 and 5.3.2 ‘0’ File Directive ’1’ . File Data
Direction	1	0-1	Used to perform PDU forwarding. ‘0’ toward file receiver ‘1’ . toward file sender
Transmission Mode	1	0-1	This will primarily be set to ‘0’ unless in a contingency mode of operations ‘0’ . acknowledged ‘1’ . unacknowledged
CRC Flag	1	‘0’	LRO will use a value of ‘0’ because no CRC calculations will be performed, a value of ‘1’ implies CRC
Reserved for future use	1	‘0’	Not Used
PDU Data field length	16		Variable in length and the value is in octets.
Reserved for future use	1	’0’	Not Used

Length of entity IDs	3	Variable	Number of octets in entity ID less one; i.e., .0. means that entity ID is one octet. Applies to all entity IDs in the PDU header.
Reserved	1	’0’	Reserved for future use
Length of Transaction sequence number	3	Variable	Number of octets in sequence number less one; i.e., .0. means that sequence number is one octet.
Source entity ID	16	Variable	Uniquely identifies the entity that originated the transaction.
Transaction sequence number	32	Variable	Uniquely identifies the transaction, among all transactions originated by this entity.
Destination entity ID	16	Variable	Uniquely identifies the entity that is the final destination of the transaction’s metadata and file data.
Data	variable	Variable	Consists of file data, engineering, memory, images, measurement, … Or A file directive

5.3.1 FILE DATA PDU

The format of the data field of File Data PDUs, which are the PDUs used to deliver the actual file data, is shown below.

Fixed PDU Header	**Segment Offset in Octets 00000000-FFFFFFFF Hex (32)**	**File Data (Variable)**

Figure 5-6. File Data PDU

The file data PDU will contain any of the actual data file types selected for transfer to the ground or spacecraft defined in section 4.6. The segment offset field is the start byte offset in the file for the data in this PDU. For example, the first PDU containing file data would have an offset of 0. And though the file data field in the PDU is variable the overall size of the File Data PDU (including fixed header) is limited to 1758 bytes for telemetry and 242 bytes for commands as shown in Table 5-2.

Table 5-2: Maximum Size for CFDP Telemetry and Command PDU

Telemetry Field		**Size (bytes)**	**Command Field**		**Size (bytes)**
CADU (2044 bytes)	Synch Mark	4	CLTU (306 bytes)	Start Sequence	2
	VCDU	1784		Frame	256
	RS	256		Fill Data	3
CCSDS AOS VCDU (1784 bytes)	Primary Header	6		Tail sequence	8
	Data Field	1772	Frame (256 bytes)	Header	5
	CLCW	4		Segment Header	1
	CRC	2		Packet	250
Data Field (1772 bytes)	M_PDU header	2	CCSDS Packet (250 bytes)	Primary Header	6
	M_PDU Packet Zone	1770		Secondary Header	2
CCSDS Packet *	Primary Header	6		Application	Max 242
	Secondary Header	6	CFDP PDU (242 bytes)	Header	12
	Application Data	8192 bytes max (which can span M_PDU packet zones)		Data/Directive	Max 230
*Special Case					
CCSDS Packet for CFDP (1770 bytes)	Primary Header	6			
	Secondary Header	6			
	CFDP PDU	1758			
CFDP PDU (1758 bytes)	Header	12			
	Data/Directive	1746			

5.3.2 FILE DIRECTIVE PDU

All other types of PDUs are considered File Directive PDUs, which convey only metadata and other non-file information that advances the operation of the protocol. The data field of File Directive PDUs consists of a Directive Code octet followed by a Directive Parameter field. The File Directive Codes are shown in Table 5-3. The formats of each of the different file directive PDUs are shown in the following subsections.

Table 5-3: File Directive Codes

Directive Code (hexadecimal)	Action	Notes
00	Reserved	
01	Reserved	
02	Reserved	
03	Reserved	
04	EOF PDU	
05	Finished PDU	
06	ACK PDU	
07	Metadata PDU	
08	NAK PDU	
09	Prompt PDU	Not Implemented for LRO
0C	Keep Alive PDU	Not Implemented for LRO
0D–FF	Reserved	

In several cases, the Directive Parameter field of a File Directive includes a four-bit Condition Code. The Condition Code will in each case indicate one of the conditions shown in Table 5-4. Details associated with the definition of condition codes are described in the Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078).

Table 5-4: Condition Codes

Condition Code (binary)	Condition
0000	No error
0001	Positive ACK limit reached
0010	Keep alive limit reached

0011	Invalid transmission mode
0100	Filestore rejection
0101	File checksum failure
0110	File size error
0111	NAK limit reached
1000	Inactivity detected
1001	Invalid file structure
1010 – 1101	(reserved)
1110	Suspend.request received
1111	Cancel.request received

Each of the file directive PDUs can be categorized into one of the following:

- Operation PDU
- Monitor and Control PDU
- Termination PDU

5.3.2.1 OPERATION PDUs

5.3.2.1.1 METADATA PDU

Functionally attached to each file may be metadata of variable size, format, and semantics. The metadata consists of information associated with the file or the file transfer such as file names, filestore requests, or messages to the CFDP user. The metadata PDU refers to the unit exchanged between source and destination that contains data to aid the recipient to effectively utilize the file data, or additional application data such as user operation messages that provide a command or response to the appropriate recipient.

Typically the metadata PDU is either the first PDU sent in a file delivery transaction or a self contained PDU classified as a message to the user whose formats are described in section 5.3.2.1.1.1. Figure 5-7 displays the breakdown of a metadata PDU, note that in general some fields or values for fields change dependent on the type metadata, whether it is a file description, message to the user, or a combination of both. The metadata PDU for LRO is used only for file descriptions because there is no planned implementation for filestore requests or messages to the user, therefore there is a standard definition for the metadata.

Fixed PDU Header	File Directive Code 07 Hex (8)	Segment Ctrl Field (1)	Reserved (7)	File Size in Octets (32)	Source Length (8)	Source File Name (8x Source Length)	Destination Length (8)	Destination File Name (8x Destination Length)

Figure 5-7. Metadata PDU

Table 5-5: Metadata PDU

Field	**Length (bits)**	**Values**	**Comment**
File Directive Code	8	07 Hex	Identifying this as a Metadata PDU
Segmentation Control Field Contents	1	'0' - Record boundaries respected '1' - Record boundaries not respected	This should be set to '0' for LRO because the plan is to send only bounded files which implies all file sizes will be known at time of transfer.
Reserved	7	'0000000'	Not used
File Size	32	File size in octets	This would be set to all 0's if file is unbounded but there is no intent on sending unbounded files.
Length	8	0-40	LV Length field indicates number of bytes associated with the source file name. Though the filed is variable up to 255, the spacecraft limitation is 40 bytes.
Source File Name	8 x length (max 320)	Variable	LV Length field indicates zero length and LV value field omitted when there is no associated file, e.g., messages used for Proxy operations. (Note: the max length of this field is 320 bits to correspond to a maximum source file length of 40 bytes)
Length	8	0-40	LV Length field indicates number of bytes associated with the destination file name. Though the filed is variable up to 255, the spacecraft limitation is 40 bytes.

Field	Length (bits)	Values	Comment
Destination File Name	8 x length (max 320)	Variable	LV Length field indicates zero length and LV value field omitted when there is no associated file, e.g., messages used for Proxy operations. (Note: the max length of this field is 320 bits to correspond to a maximum source file length of 40 bytes)

5.3.2.1.1.1 USER OPERATIONS

User operation messages are contained within the metadata PDU. Currently, there is no planned implementation for LRO; therefore these fields are not defined for the metadata PDU.

5.3.2.1.2 NEGATIVE ACKNOWLEDGMENT (NAK) PDU

Negative Acknowledgements (NAKs) are used to request the retransmission of lost data. The receiver on a data file transaction will keep track of all file data PDUs received. Upon receipt of an End of File PDU (section 5.3.2.3.1) the receiver will send a maximum of one NAK PDU to the sender requesting the PDUs that were not received or accepted during the initial transfer. Refer to Lunar Reconnaissance Orbiter CCSDS File Delivery Protocol Specification (431-SPEC-000078) for detail NAK procedures.

The sender will respond to all received NAK PDUs by retransmitting the requested Metadata PDU and/or the extents of the data file defined by the start and end offsets of the segment requests in the NAK PDU.

Fixed PDU Header	File Directive Code 08 Hex (8)	Start of Scope (32)	End of Scope (32)	'N' segment requests (64x'N')	
				Start Offset in Octets (32)	End Offset in Octets (32)

Figure 5-8. Negative Acknowledgement PDU

Table 5-6: Negative Acknowledgement PDU

Field	Length (bits)	Values	Comment
File Directive Code	8	08 Hex	Identifying this as a NAK PDU

Start of Scope		32	Variable	The start of scope for a NAK sequence will be the end of scope of the previous NAK sequence issued for the same transaction, or zero if there has been no prior NAK sequence.
End of scope		32	Variable	The end of scope for a NAK sequence will be the entire length of the file if the transaction's EOF (No error) PDU has been received. Otherwise it will be the current reception progress at the time of the event that caused issuance of the NAK sequence.
N Segment Requests (64xN)*	Start offset	32	Variable	Data — Offset of start of requested segment Metadata — 00000000 (hex)
	End Offset	32	Variable	Data — Offset of first octet after end of requested segment Metadata — 00000000 (hex)

5.3.2.2 MONITOR AND CONTROL PDUs

Monitor and control PDUs basically help control the flow of the transactions between entities and identify when the data flow is not operating as intended. The monitor and control functions are not planned for LRO.

5.3.2.2.1 PROMPT PDU

No implementation planned for LRO.

5.3.2.2.2 KEEP ALIVE PDU

No implementation planned for LRO.

5.3.2.3 TERMINATION PDUs

Termination PDUs are associated with terminating a transaction, whether the transaction completes successfully or unsuccessfully.

5.3.2.3.1 END OF FILE (EOF) PDU

The end of file (EOF) PDU is initiated from the sender to the receiver and indicates that the sender has stopped sending the requested file; this could be a complete, incomplete, or an error condition which caused the cessation of transfer. In an acknowledged mode of operations the EOF is not necessarily the end of the transaction, simply the sender's indication that it does not intend to send more data unless requested to do so.

Fixed PDU Header	File Directive Code 04 Hex (8)	Condition Code (4)	Spare (4)	File Checksum (32)	File Size in Octets (32)	Fault Location (TBR)

Figure 5-9. End of File PDU

Table 5-7: End Of File PDU

Field	Length (bits)	Values	Comment
File Directive Code	8	04 Hex	Identifying this as a End of File PDU
Condition Code	4	Defined in Table 5-4.	Condition on which the transmission ended.
Spare	4	'0000'	Not used
File Checksum	32	Variable	Modulo 2^{32} word-wide addition (where .word. is defined as 4 octets) of all file segment data transmitted by the sender (regardless of the condition code, i.e., even if the condition code is other than .No error.), aligned with reference to the start of file.
File Size	32	Variable	Expressed in octets. This value will be the total number of file data octets transmitted by the sender, regardless of the condition code (i.e., it will be supplied even if the condition code is other than .No error.).
Fault Location		TBR	Omitted if condition code is 'No error'. Otherwise, entity ID in the TLV is the ID of the entity at which transaction cancellation was initiated.

5.3.2.3.2 FINISHED PDU

A Finished PDU is sent when all file data has been successfully assembled at the destination and indicates the final completion status, such as successful, cancelled, and/or fault. Delivery of this PDU is ensured through use of the finished acknowledgement (next section). The finished PDU follows the EOF sequence after a file directive or any NAKs and/or data retransmissions have been completed, since lost data may still be outstanding after the EOF sequence.

Fixed PDU Header	File Directive Code 05 Hex (8)	Condition Code (4)	End System Status (1)	Delivery Code (1)	File Status (2)

Figure 5-10. Finished PDU

Table 5-8: Finished PDU

Field	Length (bits)	Values	Comment
File Directive Code	8	05 Hex	Identifying this as a Finished PDU
Condition Code	4	Defined in Table 5-4.	Condition on which the transmission ended.
End System Status	1	0-1	Should always be '1' because no waypoints are being used for LRO '0' - Generated by Waypoint '1' - Generated by End System
Delivery Code	1	0-1	'0' - Data Complete '1' - Data Incomplete
File Status Codes	2	00,01,10,11	File status is meaningful only when the transaction includes the transmission of file data. '00' — Delivered file discarded deliberately. '01' — Delivered file discarded due to filestore rejection. '10' — Delivered file retained in filestore successfully. '11' — Delivered file status unreported.

5.3.2.3.3 POSITIVE ACKNOWLEDGMENT (ACK) PDU

Positive acknowledgement PDU is a response mechanism implemented to ensure that a critical PDU did in fact arrive at the destination and the appropriate decisions have been made. In the acknowledgement mode of transfer a positive acknowledgement PDU is generated in response to the receipt of both the EOF and Finished PDUs. The purpose of the Expected Response is to turn off the PDU retransmission timer at the sending end. This purpose must be served regardless of the status of the transaction: undefined, active, terminated, or unrecognized. If a preset limit is exceeded, the sending CFDP entity will declare a Positive ACK Limit Reached fault.

Fixed PDU Header	File Directive Code 06 Hex (8)	Directive Code (4)	Directory Subtype Code (4)	Condition Code (4)	Spare (2)	Transaction Status (2)

Figure 5-11. Positive Acknowledgement PDU

Table 5-9: Positive Acknowledgement PDU

Field	Length (bits)	Values	Comment
File Directive Code	8	06 Hex	Identifying this as a Positive Acknowledgement PDU
Directive code	4	04 Hex (EOF) or 05 Hex (Finished)	Directive code of the acknowledged PDU. Only EOF and Finished PDUs are acknowledged.
Directive subtype code	4		Values depend on directive code. For ACK of Finished PDU: binary 0001 if generated by end system. Binary 0000 for ACKs of all other file directives.
Condition code	4	Defined in Table 5-4.	Condition code of the acknowledged PDU.
Spare	2	'00'	Not used
Transaction status	2	00,01,10,11	Status of the transaction in the context of the entity that is issuing the acknowledgment. '00' Undefined: The transaction not currently active at this entity. The transaction might be one that was formerly active and has been terminated, or it might be one that has never been active at this entity.

			‘01’ Active: The transaction is currently active at this entity. ‘10’ Terminated: The transaction is not currently active at this entity, and the transaction is known to be one that was formerly active is now terminated. ‘11’ Unrecognized: The transaction is not currently active at this entity, and the transaction is known to be one that has never been active at this entity.

Appendix A

Abbreviation/ Acronym	DEFINITION
ACK	Acknowledgement
Acq.	Acquisition
ACS	Attitude Control System
AOS	Advanced Orbiting Systems
APID	Application Identifier
ASCII	American Standard Code for Information Interchange
BCH	Bose-Chaudhuri-Hocquenghem
Bps	bytes per second
bps	bits per second
C&DH	Command & Data Handling
CADU	Channel Access Data Unit
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
CLCW	Command Link Control Word
CLTU	command link transmission unit
CM	Configuration Management
CMD	Command
CMO	Configuration Management Office
COP	Communications Operating Procedures
CRaTER	Cosmic Ray Telescope for Effects of Radiation
CRC	Cyclic Redundancy Check
CUC	CCSDS Unsegmented Time Code
CVCDU	Coded Virtual Channel Data Unit
DCN	Document Change Notice
DDD	DSN Data Delivery
DFP	Double Floating Point
DMR	Detailed Mission Requirements
DPS	Data Processing System
DSN	Deep Space Network
EOF	End of File
EOS	Earth Observatory System
FARM	Frame Acceptance and Rejection Mechanism
FARMB	Frame Acceptance and Rejection Mechanism Type B
F-CLTU	Forward Command Link Transmission Unit
FDU	File Delivery Unit
FMEA	Failure Mode and Effect Analysis
FSW	Flight Software
FSMF	Flight Software Maintenance Facility

Abbreviation/ Acronym	DEFINITION
FSWM	Flight Software Maintenance
GMT	Greenwich Mean Time
GS	Ground System
GSE	Ground Support Equipment
GSFC	Goddard Space Flight Center
GSCID	Global Spacecraft Identifier
HDR	Header
Hex	Hexadecimal
I&T	Integration and Test
ICD	Interface Control Document
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IGSE	Instrument Ground Support Equipment
IP	Internet Protocol
ISO	International Organization for Standardization
ITOS	Integrated Test and Operations System
JPL	Jet Propulsion Laboratory
kbps	kilobits per second
km	kilometer
L&EO	Launch and Early Orbit
LAMP	Lyman-Alpha Mapping Project
LEND	Lunar Exploration Neutron Detector
LEOT	Low Earth Orbiting Tracking
LOLA	Lunar Orbiter Laser Altimeter
LPRP	Lunar Precursor and Robotic Program
LRO	Lunar Reconnaissance Orbiter
LROC	Lunar Reconnaissance Orbiter Camera
LSB	least significant bit
LV	Length-Value
Mb	megabits
MB	megabytes
MC	Mission Critical
MCO	Mission Concept of Operations
MCS	master channel sequence
MO&DSD	Mission Operations and Data Systems Directorate
MOC	Mission Operations Center
M_PDU	Multiplexing Protocol Data Unit
MRD	Mission Requirements Document
MSB	most significant bit
NAK	Negative Acknowledgement
NASA	National Aeronautics and Space Administration

Abbreviation/ Acronym	DEFINITION
NISN	NASA Integrated Services Network
NPD	NASA Policy Directive
OBC	onboard computer
PB	playback
PDB	project database
PDS	Planetary Data System
PDU	Protocol Data Unit
PLOPs	Physical Link Operation Procedures
psk	phase shift key
PTP	programmable telemetry processor
RF	Radio Frequency
RS	Reed-Solomon
RTMC	Real-Time Mission Critical
S/C	Spacecraft
SB	Signed Byte
SBC	Single Board Computer
SCID	spacecraft identifier
SCN	Space Communications Network
SCP	Stored Command Processor
SCS	Sequence and Compression System
SFP	Single Floating Point
SI	Signed Integer
SLI	Signed Long Integer
SLE	Service Link Extension
SMEX	Small Explorer
SOC	Science Operations Center
STOL	Systems Test and Operations Language
T&C	Telemetry, & Command
TBD	To Be Determined
TBR	To Be Resolved
TBS	To Be Supplied
TC	Telecommand
TCP	Transmission Control Protocol
TF	Transfer Frame
TLM	Telemetry
TLV	Type-Length-Value
UB	Unsigned Byte
UI	Unsigned Integer
ULI	Unsigned Long Integer
URL	Uniform Resource Locator
USN	Universal Space Network

Abbreviation/ Acronym	DEFINITION
UTC	universal time code
VC	Virtual Channel
VCDU	Virtual Channel Data Unit
VCID	virtual channel identifier
WS	workstation