



IRIG STANDARD 107-98

TELEMETRY GROUP

**DIGITAL DATA ACQUISITION
AND ON-BOARD RECORDING STANDARD**

**WHITE SANDS MISSILE RANGE
KWAJALEIN MISSILE RANGE
YUMA PROVING GROUND
DUGWAY PROVING GROUND
ABERDEEN TEST CENTER
NATIONAL TRAINING CENTER**

**ATLANTIC FLEET WEAPONS TRAINING FACILITY
NAVAL AIR WARFARE CENTER WEAPONS DIVISION
NAVAL AIR WARFARE CENTER AIRCRAFT DIVISION
NAVAL UNDERSEA WARFARE CENTER DIVISION, NEWPORT
PACIFIC MISSILE RANGE FACILITY
NAVAL UNDERSEA WARFARE CENTER DIVISION, KEYPORT**

**30TH SPACE WING
45TH SPACE WING
AIR FORCE FLIGHT TEST CENTER
AIR FORCE DEVELOPMENT TEST CENTER
AIR WARFARE CENTER
ARNOLD ENGINEERING DEVELOPMENT CENTER
GOLDWATER RANGE
UTAH TEST AND TRAINING RANGE**

**DISTRIBUTION A: APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED**

IRIG STANDARD 107-98

**DIGITAL DATA ACQUISITION
AND ON-BOARD RECORDING STANDARD**

JUNE 1998

Prepared by

**TELEMETRY GROUP
RANGE COMMANDERS COUNCIL**

Published by

**Secretariat
Range Commanders Council
U.S. Army White Sands Missile Range,
New Mexico 88002-5110**

TABLE OF CONTENTS

	<u>Page</u>
REFERENCES.....	v
STATEMENT OF INTENT.....	vii
FOREWARD	ix
1. INTRODUCTION	1-1
1.1 PURPOSE	1-1
1.2 SCOPE	1-1
1.3 APPLICABILITY	1-1
1.4 RATIONALE	1-1
1.5 STRUCTURE OF THE DOCUMENT	1-2
1.6 CONVENTIONS AND DEFINITIONS	1-2
2. OVERVIEW	2-1
2.1 THE PACKET TELEMETRY CONCEPT	2-1
2.2 SOURCE PACKET	2-1
2.3 TRANSFER FRAME	2-2
2.4 SHARING TRANSMISSION RESOURCES	2-2
2.5 APPLICATION NOTES	2-3
3. SOURCE PACKET	3-1
3.1 PACKET PRIMARY HEADER	3-1
3.2 PACKET DATA FIELD	3-5
4. [NO LONGER USED]	4-1
5. TRANSFER FRAME	5-1
5.1 TRANSFER FRAME PRIMARY HEADER	5-1
5.2 TRANSFER FRAME SECONDARY HEADER	5-7
5.3 TRANSFER FRAME DATA FIELD	5-8
5.4 OPERATIONAL ERROR CONTROL FIELD	5-9
5.5 FRAME ERROR CONTROL FIELD	5-9
INDEX	I-1
APPENDIX A - SUMMARY OF CONCEPT AND RATIONALE	A-1
ANNEX A-1 - GLOSSARY OF TERMINOLOGY	A-20
ANNEX A-2 - “APPLICATION NOTES”	A-21
ANNEX A-3 - TRANSFER FRAME ERROR DETECTION ENCODING/DECODING GUIDELINES	A-31
APPENDIX B - CHANNEL CODING	B-1
APPENDIX C - TIME CODE FORMATS	C-1

FIGURES

PAGE

FIGURE 1-1. Bit numbering convention 1-2
FIGURE 2-1. RCC packet telemetry data system 2-1
FIGURE 2-2. Example of on-board recorded data flow 2-4
FIGURE 2-3. Digital data acquisition and on-board recording
standard 2-5
FIGURE 3-1. Source packet format 3-2
FIGURE 5-1. Transfer frame format..... 5-3

REFERENCES

- [1] "Packet Telemetry," Blue Book, CCSDS 102.0-B-4, November 1995.
- [2] "Time Code Formats," Blue Book, CCSDS 301.0-B-2, April 90.
- [3] "Telemetry Channel Coding," Blue Book, CCSDS 101.0-B-3, May 93.
- [4] "Telemetry: Concept and Rationale," Green Book, CCSDS 100.0-G-1, December 87.
- [5] "Report on Digital data Acquisition Standard," Rothrock, M.E. and Rossetti, A.M., Prepared for Telemetry Group of the Range Commanders Council, July 18, 1996.
- [6] "Reference Model of Open Systems Interconnection," International Organization for Sization, Draft International Standard DIS-7498, February 1982 or later issue.
- [7] "Telemetry Standards," Range Commanders Council Telemetry Group, IRIG Standard 106-96, May 1996.

DOCUMENT CONTROL

A. FIRST ISSUE

DOCUMENT REFERENCE: IRIG 107-98

TITLE: Digital Data Acquisition and On-Board Recording
Standard; Packet Telemetry
Issue 1

DATE: 1998

STATEMENT OF INTENT

A large number of unique data structures have been developed lately for specific data recording applications that require unique decoding software programs. Writing unique decoding software, checking the software for accuracy, and decoding the data tapes is extremely time consuming and costly.

Therefore, the need exists for a digital data acquisition standard for digital data recording that supports the multiplexing of multiple data streams and maintains the accuracy of data correlation with time. Specifically, this digital data acquisition standard should be compatible with the multiplexing of both synchronous and asynchronous digital inputs such as Pulse Code Modulation (PCM) and MIL-STD-1553 asynchronous data bus, digital voice, time, discrete, and RS-232/422 communication data. In addition, the new standard should be aligned with current developments in layered communications architecture.

This digital data acquisition standard should allow the use of a common set of playback/data reduction software, take advantage of emerging random access recording media, provide an high efficiency multiplexing technique, and take advantage of the rapid improvement in commercial communication technology.

Packet telemetry represents an evolutionary step from the traditional Time-Division Multiplex (TDM) method of acquiring, recording, and playing back scientific applications and engineering data from instrumented vehicle sources to ground data systems sinks. It also relies on a layered architectural model to isolate independent interfaces. The packet telemetry process has the conceptual attributes of:

1. Defining a logical interface and protocol between an instrument and its associated on-board recorder/playback and ground support equipment which remains constant throughout the life cycle of the instrument (bench test, integration, flight, and possible re-use).
2. Simplifying overall system design by allowing microprocessor-based symmetric design of the instrument control and data paths compatible with commercially available components and interconnection protocol standards.
3. Facilitating interoperability of instrumented vehicle systems whose data acquisition and on-board recording systems interfaces conform to IRIG guidelines.
4. Enabling the delivery of high-quality data products in a mode, which is faster and less expensive than would be possible with conventional methods.

The Consultive Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation is intended primarily to support multi-point to multi-point data transfer over space based transmission links. This standard is primarily to support point to point data acquisition and recording and subsequent playback on a ground based data reduction system. While there are significant similarities, the differences required some minor deviations from the absolute adoption of the CCSDS Packet Telemetry Recommendation. It is therefore the intent of this standard to duplicate the CCSDS Packet Telemetry Recommendation deviating only where absolutely necessary to accommodate on-board recording/ground playback or point to point

considerations. The structure of the CCSDS Packet Telemetry Recommendation, both source packet and transfer frame, are unchanged. This minimum deviation from an existing international standard will allow for maximum use of existing and proposed CCSDS compatible hardware and software.

FOREWORD

This document is a technical standard for use in developing packetized on-board recording systems and has been prepared by the Range Commanders Council (RCC) Telemetry Group (TG). The packet telemetry concept described herein is the baseline for on-board recording of missions that require cross support between organizations. This standard establishes a common framework and provides a common basis for the data structures of on-board recording data.

This standard is an adaptation of the Consultative Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation contained in reference [1].

The CCSDS Packet Telemetry Recommendation is limited to the telemetry formats which are generated by the vehicle while the channel coding and synchronization mechanisms required to implement over-the-air transmission of acceptable quality are defined in reference [3]. Since this standard is for on-board recording versus over-the-air transmission, most of reference [3] is not directly applicable.

The CCSDS Packet Telemetry Recommendation also includes references to specific time code formats as defined in reference [3] and this standard adopts a portion of reference [3].

The CCSDS Packet Telemetry Recommendation incorporates a concept and rationale document, including "Application Notes," which is contained in reference [4].

For ease of use this standard has been prepared as a stand-alone document and contains only appropriate wording changes to the adopted CCSDS Packet Telemetry Recommendation of reference [1]. The data structure for both the source packet and transfer frame remains unchanged.

The concept and rationale for this standard, including "Application Notes" are contained in appendix A.

The specific portions of reference [3] appropriate to on-board recording channel coding are included in appendix B of this standard. The specific portions of reference [2] appropriate to time code formats are included in appendix C of this standard.

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to establish a common standard for the implementation of digital data acquisition and on-board recording “packet telemetry” systems by the organizations participating in the Range Commanders Council (RCC). This standard is an adoption of the Consultative Committee on Space Data Systems (CCSDS) Packet Telemetry Recommendation (see Reference [1]).

1.2 SCOPE

Packet telemetry is a concept that facilitates the transmission of vehicle-acquired data from source to user in a standardized manner. Packet telemetry provides a mechanism for implementing common data transport structures and protocols that may enhance the development and operation of mission systems.

This standard addresses end-to-end transport of mission data sets from source application processes located on a vehicle to an on-board recording device and then played back to user application processes located on the ground.

This standard is limited to describing the recording formats which are generated by the vehicle in order to execute its role in the above processes.

An overview of the packet telemetry concept is given in Chapter 2.

1.3 APPLICABILITY

The CCSDS Standard includes comprehensive specification of the structure of data streams that are generated for recording by on-board recorders and then played back through mission data processing facilities. The standard does not attempt to define the architecture or configuration of these data processing facilities, except to describe assumed ground data handling services which affect the selection of on-board formatting options.

The CCSDS Standard specifies a wide range of formatting capabilities which may facilitate a high degree of flexibility in the design of on-board data acquisition systems; however, compatibility with the packet telemetry concept may be realized by only implementing a narrow subset of these capabilities. Application notes for implementation of DDARS is included in appendix A.

1.4 RATIONALE

The CCSDS and the RCC-TG believes it is important to document the rationale underlying the recommendations chosen so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The concept and rationale for CCSDS packet telemetry may

be found in reference [4] and the concept and rationale for the RCC Digital Data Acquisition and On-Board Recording Standard may be found in appendix A.

1.5 STRUCTURE OF THE DOCUMENT

For the designation of text partitions the following conventions will be used.

Text designated by one number belongs to a chapter.

Text designated by two numbers belongs to a section.

Text designated by three numbers belongs to a sub-section.

Text designated by four numbers belongs to a paragraph.

Text designated by a lower case letter belongs to an item.

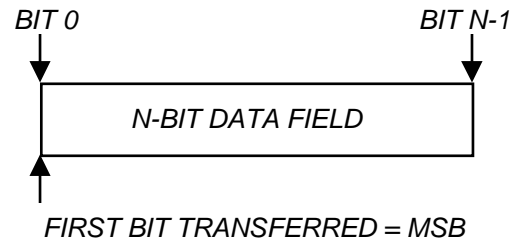


Figure 1-1. Bit Numbering Convention

All specifications are contained in Chapters 3, 4 and 5 of this standard. They are identified by an item number consisting of the number of the text partition as defined above, and a lower case letter. The conventions and definitions applied in these specifications are itemized in section 1.6.

All other text and all figures in these chapters represent comments to these specifications. All comments are printed in Italics. The contents of the specifications take precedence over those of the comments. All major terms used herein are referenced in the index.

1.6 CONVENTIONS AND DEFINITIONS

The following items contain the conventions, which have been used throughout this standard.

- a. To identify each bit in an N-bit field the first bit in the field to be transferred (i.e., the most left justified when drawing a figure) is defined to be "Bit 0"; the following bit is defined to be "Bit 1" and so on up to "Bit N-1." When the field is used to express a binary value (such as a counter), the most significant bit shall be the first bit of the field, i.e., "Bit 0" (see Figure 1-1).
- b. In accordance with modern data communication practice, vehicle data fields are often grouped into 8-bit words, which conform, to convention 1.6.a. Throughout this standard, such an 8-bit word is termed an "octet."
- c. The numbering for octets within a data structure starts with 0.
- d. The term "mission phase" designates a period of a mission during which specified on-board recording characteristics are fixed. The transition between two consecutive mission phases may cause an interruption of the on-board recording services.
- e. Certain characteristics of the data structures specified in this standard are required to remain unchanged throughout a mission phase or throughout all mission phases. In these cases the term "static" is used to specify characteristics which remain unchanged either with

respect to an application process identifier (for definition see paragraph 3.1.2.3), or within a specific virtual channel (for definition see item 5.e) or within a specific master channel (for definition see item 5.d).

f. Idle data is data that carries no information, but is sent to meet timing or synchronization requirements. The bit pattern of idle data is not specified.

2. OVERVIEW

This packet telemetry standard describes data structures used to transport data from data sources on board a vehicle to an on-board recording device, then played back through ground data systems to data sinks on the ground, as shown in Figure 2-1.

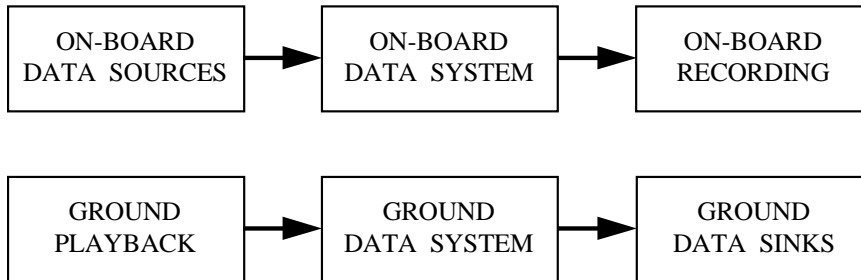


Figure 2-1. RCC Packet Telemetry Data System

2.1 PACKET TELEMETRY CONCEPT

The essence of the packet telemetry concept is to permit multiple application processes running in on-board sources to create units of data (packets) as best suits each data source. The concept permits the on-board data system to record these packets in a way that enables the ground playback system to recover the individual packets with high reliability and to provide them to the data sinks in sequence. These on-board sources are either instruments or sub-systems.

To accomplish these functions, this standard defines two data structures: source packets and transfer frame and a multiplexing process to interleave source packets from various application processes into transfer frames.

2.2 SOURCE PACKET

The source packet, which in the following text may also be termed “packet,” is a data structure generated by an on-board application process in a manner responsive to the needs of that process. It can be generated at fixed or variable intervals and may be fixed or variable in length. Aside from a packet header that identifies the source and characteristics of the packet, the internal data content of the source packet is completely under the control of the application process.

The source packet allows each application process within a data source to optimize the size and structure of its data set with a minimum of constraints imposed by the on-board digital data recording system. Each data source is independent of other data sources and can adapt its data structure to the various modes of the instrument or sub-system.

The source packet primary header contains an application process identifier used to route the packet to its destination sink. The header also carries information about the length, sequence,

and other characteristics of the packet. An optional source packet secondary header is provided for standardized time tagging of source packets, and to carry application-unique ancillary data.

2.3 TRANSFER FRAME

The transfer frame is a data structure that provides an envelope for recording packetized data. It carries information in the transfer frame primary header that permits the ground system to route the transfer frames to their intended destination. The transfer frame is of fixed length (for a given physical channel during a mission phase).

The physical channel is the single bit stream that is recorded onto the physical media. It includes all multiplexed transfer frames as well as any implementor-unique coding algorithms.

Multiple, individual, asynchronous application processes on board a vehicle can generate variable-length source packets at different rates, and these source packets can then be multiplexed together into fixed-length coded transfer frames.

The transfer frame primary header provides the necessary elements to allow the variable-length source packets from a number of application processes on a vehicle to be multiplexed into a sequence of fixed-length frames. Short packets may be contained in a single frame, while longer ones may span two or more frames. Since a packet can begin or end at any place in a frame, the entire data field of every frame can be used to carry data; there is no need to tune the sizes of packets or their order of occurrence to fit the frames.

A mechanism (idle packets) is provided for cases where a frame must be released and insufficient packet data is available. Further, frames containing idle data are defined to keep the data capture element in synchronization in the absence of data, if required.

On the ground, the information in the frame and packet headers allows the data acquisition system to extract packets in a standardized way.

In addition to packets, the transfer frame can carry one optional field, the transfer frame secondary header. The transfer frame secondary header can be used to carry fixed-length mission specific data. *[Use of privately defined data is not supported by this standard.]*

2.4 SHARING TRANSMISSION RESOURCES

As most DOD recording and communication systems are capacity-limited, multiple data channels must share access to the recorded physical channel. Therefore, the on-board data system must be able to manage the data flow to the recording device in an orderly manner. In addition, different types of data may be handled differently on the vehicle or during playback on the ground. This standard provides the method of virtual channelization for controlling the data flow.

Virtual channelization is a mechanism that allows the various sources which generate packets to be “virtually” given exclusive access to this physical channel by assigning them recording/transmission capacity on a frame-by-frame basis. Each transfer frame is identified as belonging to one of the up to eight virtual channels. Virtual channelization is normally used to separate sources or destinations with different characteristics. For example, if a vehicle contains an imaging instrument which produces packets containing many thousands of octets, and a number of other instruments which generate smaller packets, a possible system architecture would be to assign the imaging instrument packets to one virtual channel and to handle the rest by multiplexing them onto a second virtual channel. Virtual channels may also be used to allow easy separation on the ground of data streams that are to be sent to different destinations.

Figure 2-2 shows an example of the flow of on-board recorded data from several on-board sources (instruments or sub-systems), through to the playback of the same data to sink processes on the ground. At the top of the figure, generation of source packets from application processes in several data sources is shown. These packets are multiplexed into the transfer frames of several virtual channels. These transfer frames are recorded on-board, using appropriate error protection and synchronization techniques. On the ground, during playback, they are demultiplexed into virtual channels, and the packets are extracted. Source packets are then delivered to sink processes, shown at the bottom of the figure, using the application process identifiers in the source packet headers for routing. Source packets with a given application process identifier may be delivered to one or more sink processes. Packets may be time-ordered prior to delivery using the information in the packet primary header and the packet secondary header.

An example of the implementation of a typical on-board recording data flow is also shown in Figure 2. Additional details of implementation options are contained in appendix A.

Figure 2-3 shows a reference model and the extent of coverage of this standard. Application processes have separately defined RCC/IRIG standards. At the coding layer, the coding and synchronization, if required, are not specified.

2.5 APPLICATION NOTES

Application Notes that describe how compatibility with these various data structures may be achieved are presented in reference [4] along with key elements of the rationale behind packet telemetry. Application notes specific to the Digital Data Acquisition and On-Board Recording Standard are presented in appendix A.

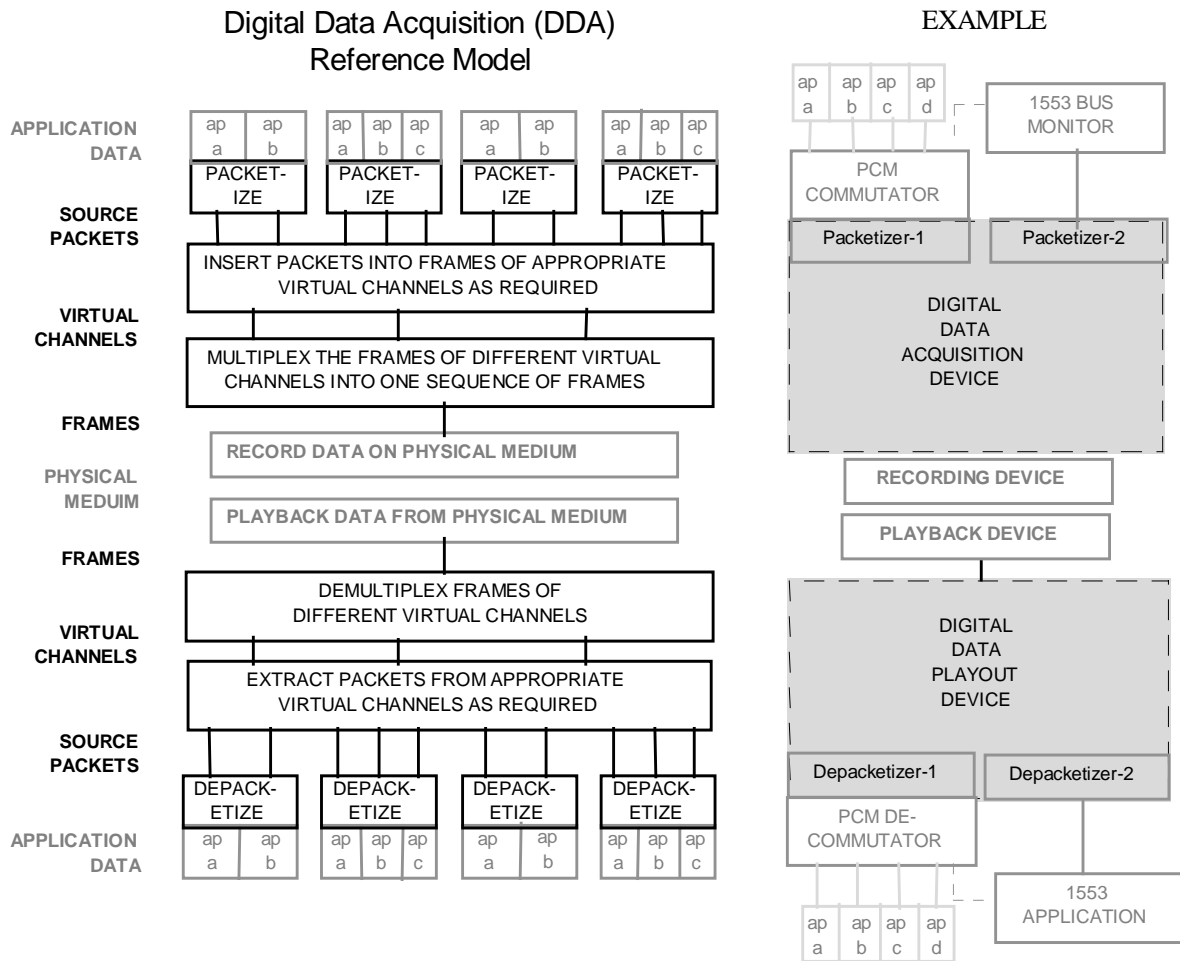


Figure 2-2. Example of on-board recorded data flow.

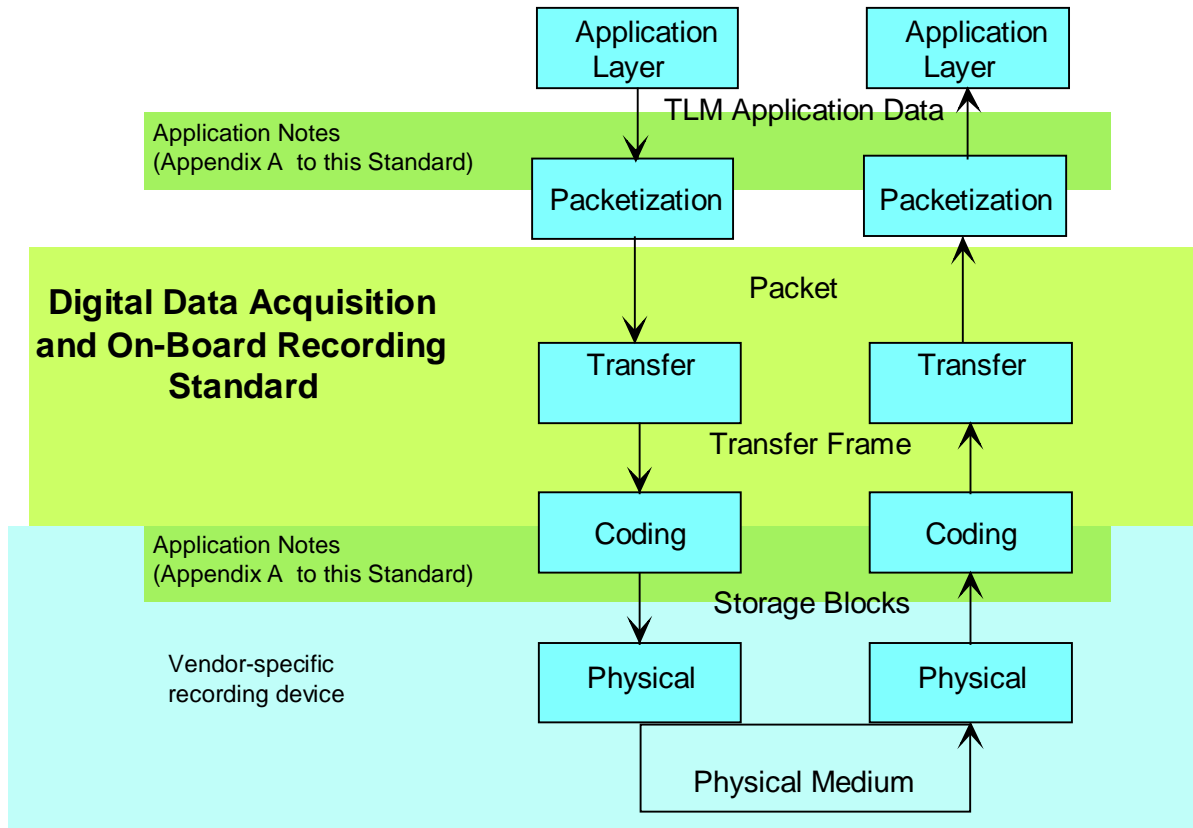


Figure 2-3. Digital data acquisition and on-board recording standard reference model.

3. SOURCE PACKET

a. A source packet, which in the following text may also be termed packet, shall encapsulate a block of observational and ancillary application data which is to be recorded from an application process on-board a vehicle; then played back to one or several sink processes on the ground.

b. The source packet shall consist of two major fields, positioned contiguously, in the following sequence:

	Length in bits
Packet primary header (mandatory)	48
Packet data field (mandatory)	variable

c. The source packet shall consist of at least 7 and at most 65542 octets.

d. A source packet which contains idle data in its packet data field is called an idle packet. Idle packets may be generated by the on-board data system when needed to maintain synchronization of the data transport and the packet extraction processes.

e. A series of source packets generated consecutively by a single application process may be designated as a group of source packets.

Figure 3-1 shows the format of the source packet as specified above including the sub-formats to be specified in the following sections.

3.1 PACKET PRIMARY HEADER

The packet primary header is mandatory and shall consist of the four fields, positioned contiguously, in the following sequence:

	Length in bits
Version number	3
Packet identification	13
Packet sequence control	16
Packet data length	16

3.1.1 Version Number.

a. The version number shall be contained within the bits 0–2 of the packet primary header.

b. This 3-bit field shall identify the data unit as a source packet and shall be set to “000”.

The version number is used to reserve the possibility of introducing other data structures.

3.1.2 Packet Identification Field.

- a. The packet identification field shall be contained within the bits 3–15 of the packet primary header.
- b. This 13-bit field shall be separated into three sub-fields:

	Length in bits
Type indicator	1
Packet secondary header flag	1
Application process identifier	11

The packet identification verifies the type of the packet, indicates whether the packet carries a secondary header or not, and provides information on the source of the data, i.e., the application process.

3.1.2.1 Type Indicator.

Bit 3 of the packet primary header shall contain the type indicator indicating the type of data unit. The type indicator shall be set to “0.”

3.1.2.2 Packet Secondary Header Flag.

- a. Bit 4 of the packet primary header shall contain the packet secondary header flag.
- b. The packet secondary header flag shall indicate the presence or absence of the packet secondary header within this source packet. It shall be “1”, if a packet secondary header is present; it shall be “0”, if a packet secondary header is not present.
- c. The packet secondary header flag shall be static with respect to the application process identifier throughout a mission phase.
- d. The packet secondary header flag shall be set to “0” for idle packets.

3.1.2.3 Application Process Identifier.

- a. Bits 5–15 of the packet primary header shall contain the application process identifier.
- b. The application process identifier shall be different for different application processes on the same master channel (for the definition of the master channel see item 5.d).
- c. For idle packets the application process identifier shall be “111111111”, i.e., “all ones”.

This identifier is tailored to local mission needs and is therefore assigned by mission management. Users should note that ground data accounting considerations might limit the number of different application processes, which may be active simultaneously. Certain application process Identifiers have been reserved for specific purposes as shown on the next page.

<i>Application ID (Decimal Equivalent)</i>	<i>Utilization</i>
2047	<i>RESERVED TO IDENTIFY A “idle” PACKET</i>
2046	<i>RESERVED BY CCSDS TO IDENTIFY A FLOW OF ENCAPSULATED ISO 8473 PACKETS</i>
2032-2045	<i>RESERVED BY CCSDS FOR POSSIBLE FUTURE USE</i>
0-2031	<i>AVAILABLE FOR USER DOMAIN ASSIGNMENT BY PROJECT ORGANIZATIONS</i>

3.1.3 Packet Sequence Control Field.

- a. The packet sequence control field shall be contained within bits 16–31 of the packet primary header.
- b. This 16-bit field shall be sub-divided into two sub-fields as follows:

	Length in bits
Grouping flags	2
Source sequence count	14

The packet sequence control field provides a sequential count of the packets generated with the same application process identifier, and if the grouping feature is applied, provides information on the position of a source packet in a group.

3.1.3.1 Grouping Flags.

- a. Bits 16 and 17 of the packet primary header shall contain the grouping flags.
- b. The grouping flags shall be set as follows:
 - “01” for the first source packet of a group;
 - “00” for a continuing source packet of a group;
 - “10” for a last source packet of a group.
- c. For a source packet not belonging to a group of source packets the grouping flags shall be set to “11”.
- d. All source packets belonging to a specific group of source packets shall originate from the same application process identified by a unique application process identifier.

The use of a group of source packets is outside the scope of this recommendation.

3.1.3.2 Source Sequence Count.

- a. Bits 18–31 of the packet primary header shall contain the source sequence count.
- b. The source sequence count shall provide the sequential binary count of each source packet generated by an application process identified by a unique application process identifier.
- c. The source sequence count shall be continuous, modulo 16384.
- d. Idle packets are not required to increment the source sequence count.
- e. A re-setting of the source sequence count before reaching 16383 shall not take place unless it is unavoidable.

The purpose of the field is to order this packet with other packets generated by the same application process, even though their natural order may have been disturbed during transport to the on-board recording device and subsequent playback to the user's processor on the ground.

The field will normally be used in conjunction with a time code (see paragraph 3.2.1.1) to provide unambiguous ordering; it is therefore essential that the resolution of the time code is sufficient for this code to increment at least once between successive recycling of the source sequence count.

If the source sequence count is re-set due to an unavoidable re-initialization of a process the completeness of a sequence of source packets cannot be determined.

3.1.4 Packet Data Length Field.

- a. The packet data length field shall be contained within bits 32–47 of the packet primary header.
- b. This 16-bit field shall contain a binary number equal to the number of octets in the packet data field minus 1.
- c. The value contained in the packet data length field may be variable and shall be in the range of 1 to 65536 octets.

Users should recognize that although very long packets are permissible, these may present special problems in terms of recording system monopolization and source data buffering and may add complexity to ground processing. The standard therefore provides the means to assign these packets to individual virtual channels (see Chapter 5). An additional measure could be to limit the maximum length of the source packets for a specific mission or mission phase.

3.2 Packet Data Field

- a. The packet data field shall follow, without gap, the packet primary header.
- b. The packet data field is mandatory and shall consist of at least one of the two fields, positioned contiguously, in the following sequence:

	Length in octets
Packet secondary header	variable
Source data field	variable

- c. The packet data field shall contain at least one octet.

3.2.1 Packet Secondary Header.

- a. If present, the packet secondary header shall follow, without gap, the packet data length field.
- b. The packet secondary header is mandatory if no source data field is present; otherwise it is optional. The presence or absence of a packet secondary header shall be signaled by the packet secondary header flag within the packet identification field (see paragraph 3.1.2.2).
- c. If present, the packet secondary header field shall consist of:

Packet secondary header identification field,*
 Packet secondary header time code preamble field,**
 Packet secondary header time code field (optional), and
 Packet secondary header data field (optional).

*Mandatory

**Mandatory if time code field is present.

The chosen options shall remain static for a specific application process identifier throughout all mission phases.

- a. If present, the packet secondary header shall contain a packet secondary header identification field. The packet secondary header identification field shall be separated into two sub-fields:

	Length in bits location (bit #)	
Time code flag	1	0
Packet secondary header field length	7	1-7

- b. The time code flag shall be a value of “1” to indicate a time code format is present.
- c. The packet secondary header field length indicates the number of octets in the packet secondary header minus 1.

1

The purpose of the secondary header is to allow (but not require) a means for placing ancillary data (time, internal data field format.) within a source packet. The maximum length of the packet secondary header shall be 127 octets.

3.2.1.1 Packet Secondary Header Time Field.

- a. The “time code format” shall consist of a preamble field (P-field) and a time field (T-field). The time code format shall consist of an integral number of octets.
- b. The preamble field shall consist of one of the CCSDS P-fields for segmented binary time codes specified in appendix C. The P-field extension flag (bit-0) shall be set to “0” indicating that there is not a second P-field octet present.
- c. The packet secondary header time code field shall consist of one of the CCSDS segmented binary time codes specified in appendix C.

The time codes defined in appendix C consist of a P-field, which identifies the time code and its characteristics, and a T-field.

- d. The time code selected shall be static for a given application process identifier throughout all mission phases.

For services such as archiving, sorting, processing and correlation with other data sets, the source sequence count may have to be concatenated with a time field in order to identify a packet unambiguously.

See also the comment concerning time code under paragraph 3.1.3.2.

3.2.1.2 Packet Secondary Header Data Field.

The packet secondary header data field shall consist of an integral number of octets.

The data field may contain any ancillary data necessary for the interpretation of the information contained within the source data field of the packet. The content and the format of this data are not specified by this standard.

3.2.2 Source Data Field.

- a. If present, the source data field shall follow, without gap, either the packet secondary header (if a packet secondary header is present) or the packet data length field (if a packet secondary header is not present).
- b. The source data field is mandatory if no packet secondary header is present, otherwise it is optional.
- c. The source data field shall contain either source data from an application process or idle data.
- d. The length of the source data field may be variable. It shall contain an integral number of octets. See also the specifications in sub-section 3.1.4.

4. [NO LONGER USED]

[This chapter defined the source packet segment, which is no longer defined in this issue.]

5. TRANSFER FRAME

- a. The transfer frame shall provide the data structure for the transmission of (1) source packets, and (2) idle data across the data recording channel.
- b. The transfer frame shall encompass the major fields, positioned contiguously, in the following sequence:

	Length in bits
Transfer frame primary header (mandatory)	48
Transfer frame secondary header (optional)	16, 24, ... or 512
Transfer frame data field (mandatory)	variable
Operational control field (not used)	32
Frame error control field (mandatory)	16

- c. The transfer frame shall be of constant length throughout a specific mission phase.

The maximum length of a transfer frame data field is limited by reference [3] to 8,920 bits. See appendix B for excerpts of the appropriate sections of reference [3] being adopted by this standard.

- d. All transfer frames with the same transfer frame version number (see sub-section 5.1.1) and the same vehicle identifier (see paragraph 5.1.2.1) on the same physical channel constitute a master channel.

In most cases the master channel will be identical with the physical channel. However, when the physical channel also carries transfer frames with other vehicle identifiers a distinction between master channel and physical channel is necessary, i.e., multiplexing of transfer frames with different vehicle identifiers will be performed by the multiplexing of different master channels on the same physical channel.

- e. A master channel shall consist of between one and eight virtual channels.

Although packet telemetry systems may be designed to tolerate channel noise, full benefit from packet telemetry will require that a high-quality data channel be provided so that packetized data may be adaptively inserted into the frame. The synchronization bits (Attached Sync Marker (ASM)) shall adhere to the requirements of reference [3]. See appendix B for excerpts of the appropriate sections of reference [3] being adopted by this standard.

Figure 5-1 illustrates the detailed format of the transfer frame.

5.1 TRANSFER FRAME PRIMARY HEADER

The transfer frame primary header is mandatory and shall consist of five fields, positioned contiguously, in the following sequence:

	Length in bits
Transfer frame version number	2
Transfer frame identification	14
Master channel frame count	8
Virtual channel frame count	8
Transfer frame data field status	16

The primary header covers five principal functions:

*Identification of the data unit as a transfer frame,
 Identification of the vehicle (and possibly of the link, if applicable) that recorded the data,
 Multiplexing of the virtual channels into one master channel,
 Providing a counting mechanism for the virtual channels and the master channel, and
 Providing pointers and other control information so that a variable length source packet may be extracted from the transfer frame data field.*

5.1.1 Transfer Frame Version Number.

- a. The transfer frame version number shall be contained within bits 0–1 of the transfer frame primary header.
- b. This 2-bit field shall identify the data unit as a transfer frame; it shall be set to “00”.

This standard defines Version 1 of the transfer frame.

5.1.2 Transfer Frame Identification Field.

- a. The transfer frame identification field shall be contained within bits 2–15 of the transfer frame primary header.
- b. This 14-bit field shall be sub-divided into three sub-fields as follows:

	Length in bits
Vehicle identifier	10
Virtual channel identifier	3
Operational control field flag	1

This field identifies the generator of the transfer frame, it specifies the virtual channel to which it belongs, and it provides information on the format of the transfer frame.

5.1.2.1 Vehicle Identifier.

- a. Bits 2–11 of the transfer frame primary header shall contain the vehicle identifier.
- b. The vehicle identifier shall provide the identification of the vehicle that created the frame of data.
- c. The vehicle identifier shall be static throughout all mission phases.

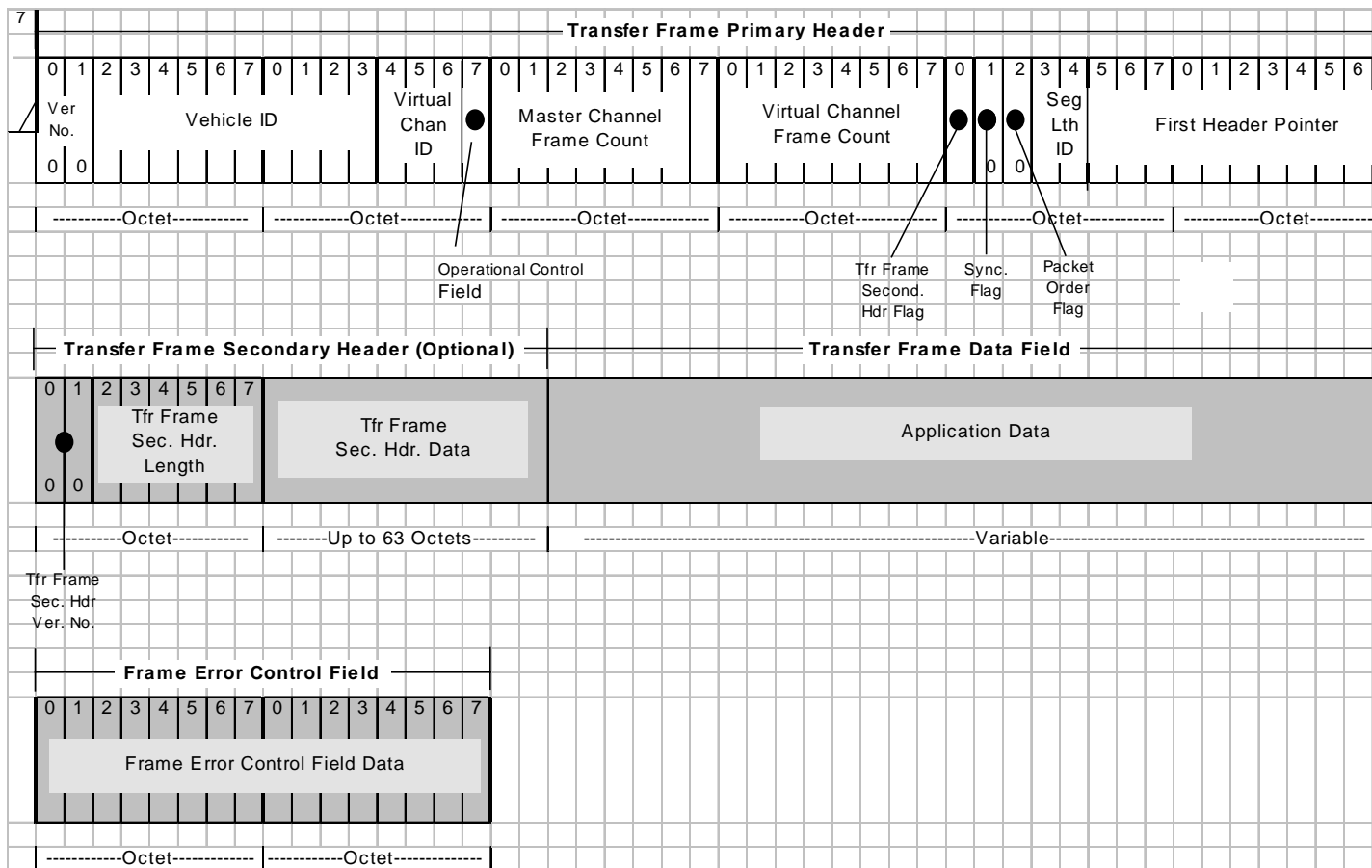


Figure 5-1. Transfer frame format.

5.1.2.2 Virtual Channel Identifier.

- a. Bits 12–14 of the transfer frame primary header shall contain the virtual channel identifier.
- b. The virtual channel identifier provides the identification of the virtual channel.

The order of occurrence of different virtual channels on a master channel may vary.

5.1.2.3 Operational Control Field Flag.

Bit 15 of the transfer frame primary header shall contain the operational control field flag.

(For purposes of this standard, the operational control field will not be used and the operational control field flag shall be set to “0” (operational control field, not present)).

5.1.3 Master Channel Frame Count Field.

- a. The master channel frame count field shall be contained within bits 16–23 of the transfer frame primary header.
- b. This 8-bit field shall contain a sequential binary count (modulo 256) of each transfer frame transmitted within a specific master channel.
- c. A re-setting of the master channel frame count before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide a running count of the frames that have been transmitted through the same master channel.

If the master channel frame count is re-set due to an unavoidable re-initialization, the completeness of a sequence of transfer frames cannot be determined.

5.1.4 Virtual Channel Frame Count Field.

- a. The virtual channel frame count field shall be contained within bits 24–31 of the transfer frame primary header.
- b. This 8-bit field shall contain a sequential binary count (modulo 256) of each transfer frame transmitted through a specific virtual channel of a master channel.
- c. A re-setting of the virtual channel frame count before reaching 255 shall not take place unless it is unavoidable.

The purpose of this field is to provide individual accountability for each of the maximum eight virtual channels, primarily to enable systematic source packet extraction from the transfer frame data field.

If the virtual channel frame count is re-set due to an unavoidable re-initialization the completeness of a sequence of transfer frames in the related virtual channel can not be determined.

5.1.5 Transfer Frame Data Field Status Field.

The transfer frame data field status field shall be contained within bits 32–47 of the transfer frame primary header.

This 16-bit field shall be sub-divided into five sub-fields as follows:

	Length in bits
Transfer frame secondary header flag	1
Synchronization flag	1
Packet order flag	1
Segment length identifier	2
First header pointer	11

This field indicates whether a secondary header is present. Further, it provides information on the type of data contained in the frame and provides, together with the virtual channel frame count, the control information necessary to enable source packets to be extracted from the transfer frame data field.

5.1.5.1 Transfer Frame Secondary Header Flag.

- a. Bit 32 of the transfer frame primary header shall contain the transfer frame secondary header flag.
- b. The transfer frame secondary header flag shall signal the presence or absence of the transfer frame secondary header. It shall be “1”, if a transfer frame secondary header is present; it shall be “0”, if a transfer frame secondary header is not present.
- c. The transfer frame secondary header flag shall be static within a specific master channel throughout a mission phase when the transfer frame secondary header is associated with a master channel.
- d. The transfer frame secondary header flag shall be static within a specific virtual channel throughout a mission phase when the transfer frame secondary header is associated with a virtual channel.

For the significance of the above mentioned associations see item 5.2.d.

5.1.5.2 Synchronization Flag.

- a. Bit 33 of the transfer frame primary header shall contain the synchronization flag.
- b. The synchronization flag shall signal the type of data, which is inserted into the transfer frame data field. It shall be “0” if octet-synchronized and forward-ordered source packets or idle data are inserted; it shall be “1” if privately defined data is inserted. For purposes of this standard privately defined data shall not be used and the synchronization flag shall always be set to “0.”
- c. The synchronization flag shall be static within a specific virtual channel throughout a mission phase.

Source packet data units are normally inserted into the transfer frame data field synchronously on octet boundaries one following directly after another. Generally, the source packets “spill over” into the next frame for the same virtual channel; therefore, source packets do not usually begin at the first octet of the transfer frame data field. The location of the first source packet header in a particular transfer frame is identified by the first header pointer field. (See comment under paragraph 5.1.5.5 also.)

5.1.5.3 Packet Order Flag.

- a. Bit 34 of the transfer frame primary header shall contain the packet order flag.
- b. The packet order flag is reserved for future use and shall be set to “0.”

5.1.5.4 Segment Length Identifier.

- a. Bits 35 and 36 of the transfer frame primary header shall contain the segment length identifier.
- b. The segment length identifier shall be set to “11.”

5.1.5.5 First Header Pointer.

- a. Bits 37–47 of the transfer frame primary header shall contain the first header pointer.
- b. If the synchronization flag is set to “0,” the first header pointer shall contain information on the position of the first source packet within the transfer frame data field.
- c. The locations of the octets in the transfer frame data field shall be numbered in ascending order. The first octet in this field is assigned the number 0. The first header pointer shall contain the binary representation of the location of the first octet of the first packet primary header.

The locations of any subsequent headers within the same transfer frame data field will be determined by calculating these locations using the packet data length field. (See subsection 3.1.4.)

The specification also covers the following two special cases:

(1) If a first source packet primary header starts at the end of the transfer frame data field within frame N and spills over into frame M of the same virtual channel, the first header pointer in frame N indicates the start of this header.

(2) If a source packet header is split between frames N and M ($M > N$), the first header pointer in frame M ignores the residue of the split header and only indicates the start of any subsequent new source packet header within frame M.

In both cases (1) and (2) above, one or more frames with idle data may occur between frame N and frame M ($M > N$).

- d. If no packet primary header starts in the transfer frame data field, the first header pointer shall be set to “1111111111.”
- e. If a transfer frame contains idle data in its transfer frame data field, the first header pointer shall be set to “1111111110.”

5.2 TRANSFER FRAME SECONDARY HEADER

- a. If present, the transfer frame secondary header shall follow, without gap, the transfer frame primary header.
- b. The transfer frame secondary header is optional; its presence or absence shall be signaled by the transfer frame secondary header flag in the transfer frame primary header. (See paragraph 5.1.5.1.)

c. The transfer frame secondary header shall consist of an integral number of octets as follows:

	Length in bits
Transfer frame secondary header identification field	8
Transfer frame secondary header data field	8, 16, or 504

d. The transfer frame secondary header shall be associated with either a master channel or a virtual channel.

The association of a secondary header with a master channel allows data to be transferred frame-synchronously with respect to this master channel.

e. The transfer frame secondary header shall be of fixed length within the associated master channel or within the associated virtual channel throughout a mission phase.

5.2.1 Transfer Frame Secondary Header Identification Field.

- a. The transfer frame secondary header identification field shall be contained within bits 0–7 of the transfer frame secondary header.
- b. The transfer frame secondary header identification field shall be sub-divided into two sub-fields as follows:

	Length in bits
Transfer frame secondary header version number field	2
Transfer frame secondary header length field	6

5.2.1.1 Transfer Frame Secondary Header Version Number.

- a. The transfer frame secondary header version number shall be contained within bits 0–1 of the transfer frame secondary header.
- b. The transfer frame secondary header version number shall be set to “00.”

This sub-field shall indicate which of up to four secondary header versions is used. The present standard recognizes only one version.

5.2.1.2 Transfer Frame Secondary Header Length.

- a. The transfer frame secondary header length shall be contained within bits 2–7 of the transfer frame secondary header.
- b. This sub-field shall contain the total length of the transfer frame secondary header in octets minus one, represented as a binary number.

- c. The transfer frame secondary header length shall be static either within a specific master channel or a specific virtual channel throughout a mission phase.

When a secondary header is present, this length may be used to compute the location of the start of the frame data field.

5.2.2 Transfer Frame Secondary Header Data Field.

- a. The transfer frame secondary header data field shall follow, without gap, the transfer frame secondary header identification field.
- b. The transfer frame secondary header data field shall contain the transfer frame secondary header data.

The field could be used for a time code (see paragraph 3.2.1.1; its insertion is, however, not mandatory) to facilitate seeking start/stop times during ground playback. See appendix A for possible uses for this field.

5.3 TRANSFER FRAME DATA FIELD

- a. The transfer frame data field shall follow, without gap, the transfer frame primary header or the transfer frame secondary header if present.
- b. The transfer frame data field shall contain the data to be recorded on-board the vehicle and shall consist of an integral number of octets. Transfer frame data may be any of the three types of data specified in item 5.a.
- c. Source packets shall be inserted contiguously and in forward order into the transfer frame data field.
- d. The length of the transfer frame data field shall be constrained by the length of the total transfer frame. For this constraint see item 5.c.
- e. Source packets may either be transmitted on separate virtual channels or may be mixed on the same virtual channel.
- f. In the case where not sufficient source packets (including idle packets) are available to fill a transfer frame data field, a transfer frame with a data field containing only idle data shall be transmitted.

Transfer frames containing idle data in their data fields may have to be sent to maintain synchronization with the recording device and also because the secondary header may still be needed to transmit valid data.

Transfer frames carrying idle data may be sent on a virtual channel that also carries packets, but a separate virtual channel dedicated to idle data is preferred.

Idle data in a transfer frame data field must not be confused with idle packets specified in item 3.d.

Packets with different application process identifiers may be multiplexed in the frame data field in any combination.

5.4 OPERATIONAL CONTROL FIELD

(The operational control field is not supported by this issue of the standard.)

5.5 FRAME ERROR CONTROL FIELD

- a. The frame error control field shall occupy the two octets following, without gap, the transfer frame data field.
- b. The frame error control field is mandatory.
- c. The frame error control field shall occur within every transfer frame transmitted within the same master channel throughout a mission phase. If the frame error control field is not utilized for error detection purposes, it is recommended to fill the field with all ones or zeros.

The purpose of this field is to provide a capability for detecting errors that have been introduced into the frame during the data handling process.

5.5.1 Encoding Procedure.

- a. The encoding procedure accepts an (n-16)-bit transfer frame, excluding the frame error control field, and generates a systematic binary (n, n-16) block code by appending a 16-bit frame error control field as the final 16 bits of the codeblock.
- b. The equation for the contents of the frame error control field is:
$$\text{FEFC} = [(X^{16} \cdot M(X)) + (X^{(n-16)} \cdot L(X))] \text{ modulo } G(X)$$

where

- all arithmetic is modulo 2;
- n is the number of bits in the encoded message;
- M(X) is the (n-16)-bit message to be encoded expressed as a polynomial with binary coefficients;
- L(X) is the presetting polynomial given by

$$L(X) = \sum_{i=0}^{15} X^i ;$$

- G(X) is the generating polynomial given by:

$$G(X) = X^{16} + X^{12} + X^5 + 1.$$

The $X^{(n-16)} \cdot L(X)$ term has the effect of presetting the shift register to all "1" state prior to encoding.

5.5.2 Decoding Procedure.

The error detection syndrome, $S(X)$, is given by

$$S(X) = [(X^{16} \cdot C^*(X)) + (X^n \cdot L(X))] \text{ modulo } G(X)$$

where

$C^*(X)$ is the received block, including the frame error control field, in polynomial form
 $S(X)$ is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected.

INDEX

- application notes, 2-4
- application process, 2-1, 2-2, 2-3, 3-1, 3-6, 3-7, 3-10
- application process identifier, 1-3, 2-2, 2-3, 3-4, 3-6, 3-7, 3-8, 3-10
- coding layer, 2-3
- decoding procedure, 5-12
- Digital Data Acquisition and On-Board Recording Standard, 1-2, 2-4
- encoding procedure, 5-11
- first header pointer, 5-6, 5-7, 5-8
- frame error control field, 5-1, 5-11, 5-12
- group of source packets, 3-1, 3-6
- grouping flags, 3-5, 3-6
- idle data, 1-3, 2-2, 3-1, 3-10, 5-1, 5-7, 5-8, 5-10
- idle packet, 2-2, 3-1, 3-4, 3-7, 5-10
- master channel, 1-3, 3-4, 5-1, 5-5, 5-6, 5-8, 5-9, 5-11
- master channel frame count, 5-2, 5-5
- master channel frame count field, 5-5
- mission phase, 1-3, 2-2, 3-4, 3-8, 3-10, 5-1, 5-3, 5-6, 5-7, 5-9, 5-11
- most significant bit, 1-2
- N-bit field, 1-2
- octet, 1-3
- operational control field, 5-1
- operational control field flag, 5-3, 5-5
- packet, 3-1
- packet data field, 3-1, 3-7, 3-8
- packet data length, 3-1
- packet data length field, 3-7, 3-8
- packet identification, 3-1
- packet identification field, 3-4, 3-8
- packet order flag, 5-6, 5-7
- packet primary header, 2-3, 3-1, 3-4, 3-5, 3-7, 5-7, 5-8
- packet primary header., 3-5
- packet secondary header, 2-3, 3-4, 3-8, 3-10
- packet secondary header data field, 3-8, 3-10
- packet secondary header flag, 3-4, 3-8
- packet secondary header time code field, 3-8, 3-10
- packet sequence control, 3-1
- packet sequence control field, 3-5
- packet telemetry, 1-1, 2-4
- packet telemetry standard, 2-1
- physical channel, 2-2, 2-3, 5-1
- privately defined data, 5-7
- Range Commanders Council, 1-1
- reference model, 2-3
- SEGMENT length identifier, 5-6, 5-7
- sink process, 2-3, 3-1
- source data, 3-10
- source data field, 3-8, 3-10
- source packet, 2-1, 2-2, 2-3, 3-1, 3-2, 3-4, 3-5, 3-7, 5-1, 5-7, 5-10
- source packet primary header, 2-2
- source packet secondary header, 2-2
- source packets, 5-10
- source sequence count, 3-5, 3-7
- standard. See Digital Data Acquisition and On-Board Recording Standard.
- synchronization flag, 5-6, 5-7
- transfer frame, 2-1, 2-2, 2-3, 5-1, 5-2, 5-5, 5-8, 5-10, 5-11
- transfer frame data, 5-10

transfer frame data field, 5-1, 5-6, 5-7, 5-8, 5-10, 5-11
transfer frame data field status, 5-2
transfer frame identification, 5-2
transfer frame identification field, 5-2
transfer frame primary header, 2-2, 5-1, 5-2, 5-3, 5-4, 5-5, 5-6, 5-7, 5-8, 5-10
transfer frame secondary header, 2-2, 5-1, 5-6, 5-8, 5-9, 5-10
transfer frame secondary header data field, 5-8, 5-10
transfer frame secondary header flag, 5-6, 5-8
transfer frame secondary header identification field, 5-8, 5-9, 5-10
transfer frame secondary header length, 5-9
transfer frame secondary header length field, 5-9
transfer frame secondary header version number field, 5-9
transfer frame version number, 5-1, 5-2
type indicator, 3-4

vehicle identifier, 5-1, 5-3
version number, 3-1
virtual channel, 1-3, 2-3, 5-1, 5-4, 5-5, 5-6, 5-7, 5-8, 5-9, 5-10
virtual channel frame count, 5-2, 5-5
virtual channel frame count field, 5-5
virtual channel identifier, 5-3, 5-4
virtual channelization, 2-3