Consultative
Committee for
Space Data Systems

**REPORT CONCERNING SPACE
DATA SYSTEM STANDARDS**

TELEMETRY

SUMMARY OF
CONCEPT AND RATIONALE

CCSDS 100.0-G-1

GREEN BOOK

DECEMBER 1987

AUTHORITY

Issue:	Green Book, Issue 1
Date:	January 1987
Location:	CCSDS Plenary Meeting November 1986 Frascati, Italy

This report reflects the consensus of the technical panel experts of the following member Agencies of the Consultative Committee for Space Data Systems (CCSDS):

- o British National Space Centre (BNSC)/United Kingdom.
- o Centre National D'Etudes Spatiales (CNES)/France.
- o Deutsche Forschungs-u. Versuchsanstalt fuer Luft und Raumfahrt e.V (DFVLR)/ West Germany.
- o European Space Agency (ESA)/Europe.
- o Indian Space Research Organization (ISRO)/India.
- o Instituto de Pesquisas Espaciais (INPE)/Brazil.
- o National Aeronautics and Space Administration (NASA)/USA.
- o National Space Development Agency of Japan (NASDA)/Japan.

The panel experts of the following observer Agencies also technically concur with this report:

- o Chinese Academy of Space Technology (CAST)/People's Republic of China.
- o Department of Communications, Communications Research Centre (DOC-CRC)/Canada.

This report is published and maintained by:

CCSDS Secretariat
Communications and Data Systems Division (Code-TS)
National Aeronautics and Space Administration
Washington, DC 20546, USA

FOREWORD

This CCSDS report presents the conceptual framework and rationale for the CCSDS Telemetry System. The background information provided here will be found helpful in understanding the two CCSDS technical Recommendations for Telemetry.

This report supports CCSDS Recommendations for "Packet Telemetry" (Reference [1]) and "Telemetry Channel Coding" (Reference [2]).

Through the process of normal evolution, it is expected that expansion, deletion or modification to this report may occur. This report is therefore subject to CCSDS document management and change control procedures which are defined in Reference [3].

Questions relative to the contents or status of this report should be addressed to the CCSDS Secretariat.

DOCUMENT CONTROL

Issue	Title	Date	Status/Remarks
CCSDS 100.0-G-1	Report Concerning Space Data System Standards, Telemetry: Summary of Concept and Rationale, Issue 1	December 1987	Current Issue

[This page intentionally left blank.]

CONTENTS

Sections Page

Annexes

Figures

Table

REFERENCES

[1] "Packet Telemetry", Recommendation CCSDS 102.0-B-2, Issue 2, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[2] "Telemetry Channel Coding", Recommendation CCSDS 101.0-B-2, Issue 2, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[3] "Reference Model of Open Systems Interconnection", International Organization for Standardization, Draft International Standard DIS-7498, February 1982 or later issue.

[4] Rice, R.F., and Hilbert, E., US Patent 3988677, October 26, 1976.

[5] Morakis, J.C., "Discussion of Synchronization Words", NASA Technical Memorandum 86222, NASA-Goddard Space Flight Center, Greenbelt, Maryland, May 15, 1985.

[6] "Procedures Manual for the Consultative Committee for Space Data Systems", Issue 1, Consultative Committee for Space Data Systems, August 1985 or later issue.

[7] Cager, R., "Spacecraft Identification Requirements Analysis", CCSDS Panel 1-C Telecommand Action Item 6.26, June 3-7, 1985.

[8] "Telecommmand: Summary of Concept and Service", Report CCSDS 200.0-G-6, Issue 6, Green Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[9] "Telecommand, Part 2: Data Routing Service, Architectural Specification", Recommendation CCSDS 202.0-B-1, Issue 1, Blue Book, Consultative Committee for Space Data Systems, January 1987 or later issue.

[10] Rice, R.F., *Channel Coding and Data Compression System Considerations for Efficient Communication of Planetary Imaging Data*, Technical Memorandum 33-695, NASA-Jet Propulsion Laboratory, Pasadena, California, June 15, 1974.

[11] Rice, R.F., *End-to-End Imaging Rate Advantages of Various Alternative Communication Systems*, JPL Publication 82-61, NASA-Jet Propulsion Laboratory, Pasadena, California, September 1, 1982

[12] Rice, R.F., *Mission Science Value/Cost Savings from the Advanced Imaging Communications System*, JPL Publication 84-33, NASA-Jet Propulsion Laboratory, Pasadena, California, July 15, 1984.

[13] Miller, R.L., et al., *On the Error Statistics of Viterbi Decoding and the Performance of Concatenated Codes*, JPL Publication 81-9, NASA-Jet Propulsion Laboratory, Pasadena, California, September 1, 1981.

[14] Odenwalder, J.P., *Concatenated Reed-Solomon/Viterbi Channel Coding for Advanced Planetary Missions, Final Report,* Contract 953866, December 1, 1974.

[15] Liu, K.Y., *The Effects of Receiver Tracking Phase Error on the Performance of Concatenated Reed-Solomon/Viterbi Channel Coding System*, JPL Publication 81-62, NASA-Jet Propulsion Laboratory, Pasadena, California, September 1, 1981.

[16] Odenwalder, J.P., et al., *Hybrid Coding Systems Study, Final Report,* NASA-Ames Research Center Contract NAS2-6722, Linkabit Corporation, San Diego, California, September 1972.

[17] Perlman, M., and Lee, J.J., *Reed-Solomon Encoders - Conventional vs Berlekamp's Architecture*, JPL Publication 82-71, NASA-Jet Propulsion Laboratory, Pasadena, California, December 1, 1982.

[18] *Tracking and Data Relay Satellite System (TDRSS) Users' Guide,* STDN 101.2, Rev. 5, NASA-Goddard Space Flight Center, Greenbelt, Maryland, September 1984.

The latest issues of CCSDS documents may be obtained from the CCSDS Secretariat at the address indicated on page i.

1 DOCUMENT PURPOSE, SCOPE AND ORGANIZATION

1.1 PURPOSE

This report contains the concept and supporting rationale for the Telemetry System developed by the Consultative Committee for Space Data Systems (CCSDS). It has been prepared to serve two major purposes:

(1) To provide an introduction and overview for the Telemetry System concept upon which the detailed CCSDS Telemetry Recommendations (References [1] and [2]) are based.

(2) To summarize the specific individual Recommendations and to supply the supporting rationale.

This document is a CCSDS informational **Report** and is therefore not to be taken as a CCSDS **Recommendation** for Data System Standards.

1.2 SCOPE

The concepts, protocols and data formats developed for the Telemetry System described herein are designed for flight and ground data systems supporting conventional, contemporary free flyer spacecraft. Data formats are designed with efficiency as a primary consideration, i.e., format overhead is minimized. The results reflect the consensus of experts from many space agencies.

1.3 ORGANIZATION

An overview of the CCSDS Telemetry System is presented in Section 2, which introduces the notion of architectural layering to achieve transparent and reliable delivery of scientific and engineering sensor data (generated aboard remote space vehicles) to the users located in space or on Earth.

Section 3 presents a more detailed description of the Telemetry System and rationale for the two specific CCSDS Telemetry Recommendations.

Annex A presents a Glossary in order to familiarize the reader with the terminology used throughout the CCSDS Telemetry System.

Annex B contains application notes which describe how a Project may implement complete or partial compatibility with the CCSDS Telemetry Recommendations [1] and [2].

Annex C summarizes the segmentation options available for segmenting very long Source Packets.

Annex D is a guideline for Transfer Frame error detection coding.

2 OVERVIEW OF CCSDS TELEMETRY SYSTEM

2.1 INTRODUCTION

The purpose of a telemetry system is to reliably and transparently convey measurement information from a remotely located data generating source to users located in space or on Earth. Typically, data generators are scientific sensors, science housekeeping sensors, engineering sensors and other subsystems on-board a spacecraft.

The advent of capable microprocessor based hardware will result in data systems with demands for greater throughput and a requirement for corresponding increases in spacecraft autonomy and mission complexity. These facts, along with the current technical and fiscal environments, create a need for greater telemetering capability and efficiency with reduced costs.

Traditionally, most of the telemetry resources used by a science mission have been wholly contained within a cognizant Project office and, with the exception of the tracking network, are completely dedicated to that mission. The lack of effective standardization among various missions forces the "multi-mission" tracking network to implement the lowest level of telemetry transport service, i.e., bit transport. Higher level data delivery services, oriented more toward computer-to-computer transfers and typical of modern day commercial and military networks, must be custom designed and implemented on a mission-to-mission basis.

The intent of the CCSDS Telemetry System is not only to ease the transition toward greater automation within individual space agencies, but also to ensure harmony among the agencies, thereby resulting in greater cross-support opportunities and services.

The CCSDS Telemetry System is broken down into two major conceptual categories: a **"Packet Telemetry"** concept (Reference [1]) and a **"Telemetry Channel Coding"** concept (Reference [2]).

Packet Telemetry is a concept which facilitates the transmission of space-acquired data from source to user in a standardized and highly automated manner. Packet Telemetry provides a mechanism for implementing common data structures and protocols which can enhance the development and operation of space mission systems. Packet Telemetry addresses the following two processes:

(1) The end-to-end transport of space mission data sets from source application processes located in space to distributed user application processes located in space or on Earth.

(2) The intermediate transfer of these data sets through space data networks; more specifically, those elements which contain spacecraft, radio links, tracking stations and mission control centers as some of their components.

The Packet Telemetry Recommendation contained in Reference [1] is primarily concerned with describing the telemetry formats which are generated by spacecraft in order to execute their roles in the above processes.

Telemetry Channel Coding is a method by which data can be sent from a source to a destination by processing it in such a way that distinct messages are created which are easily distinguishable from one another. This allows reconstruction of the data with low error probability, thus improving the performance of the channel. The Telemetry Channel Coding Recommendation contained in Reference [2] describes several space telemetry channel coding schemes. The characteristics of the codes are specified only to the extent necessary to ensure interoperability and cross-support.

Together, Packet Telemetry and Telemetry Channel Coding services provide to the user reliable and transparent delivery of telemetry information.

2.2 TELEMETRY SYSTEM CONCEPT

The system design technique known as layering was found to be a very useful tool for transforming the Telemetry System concept into sets of operational and formatting procedures. The layering approach is patterned after the International Organization for Standardization's Open Systems Interconnection layered network model (Reference [3]), which is a seven layer architecture that groups functions logically and provides conventions for connecting functions at each layer. Layering allows a complex procedure such as the telemetering of spacecraft data to the users to be decomposed into sets of peer functions residing in common architectural strata.

Within each layer, the functions exchange data according to established standard rules or "protocols". Each layer draws upon a well defined set of services provided by the layer below, and provides a similarly well defined set of services to the layer above. As long as these service interfaces are preserved, the internal operations within a layer are unconstrained and transparent to other layers. Therefore, an entire layer within a system may be removed and replaced as dictated by user or technological requirements without destroying the integrity of the rest of the system. Further, as long as the appropriate interface protocol is satisfied, a customer (user) can interact with the system/service at any of the component layers. Layering is therefore a powerful tool for designing structured systems which change due to the evolution of requirements or technology.

A companion standardization technique that is conceptually simple, yet very robust, is the encapsulation of data within an envelope or "header". The header contains the identifying information needed by the layer to provide its service while maintaining the integrity of the envelope contents.

Figure 2-1 illustrates the CCSDS Telemetry System in terms of a layered service model. It should be noted that the CCSDS Packet Telemetry and Telemetry Channel Coding Recommendations only address the five lower layers of this model.

2.2.1 PACKETIZATION LAYER

Within Packet Telemetry, spacecraft generated application data are formatted into end-to-end transportable data units called **TM Source Packets.** These data are encapsulated within a primary header which contains identification, sequence control and packet length information, and an optional trailing error control field. A TM Source Packet is the basic data unit telemetered to the user by the spacecraft and generally contains a meaningful quantity of related measurements from a particular source.

2.2.2 SEGMENTATION LAYER

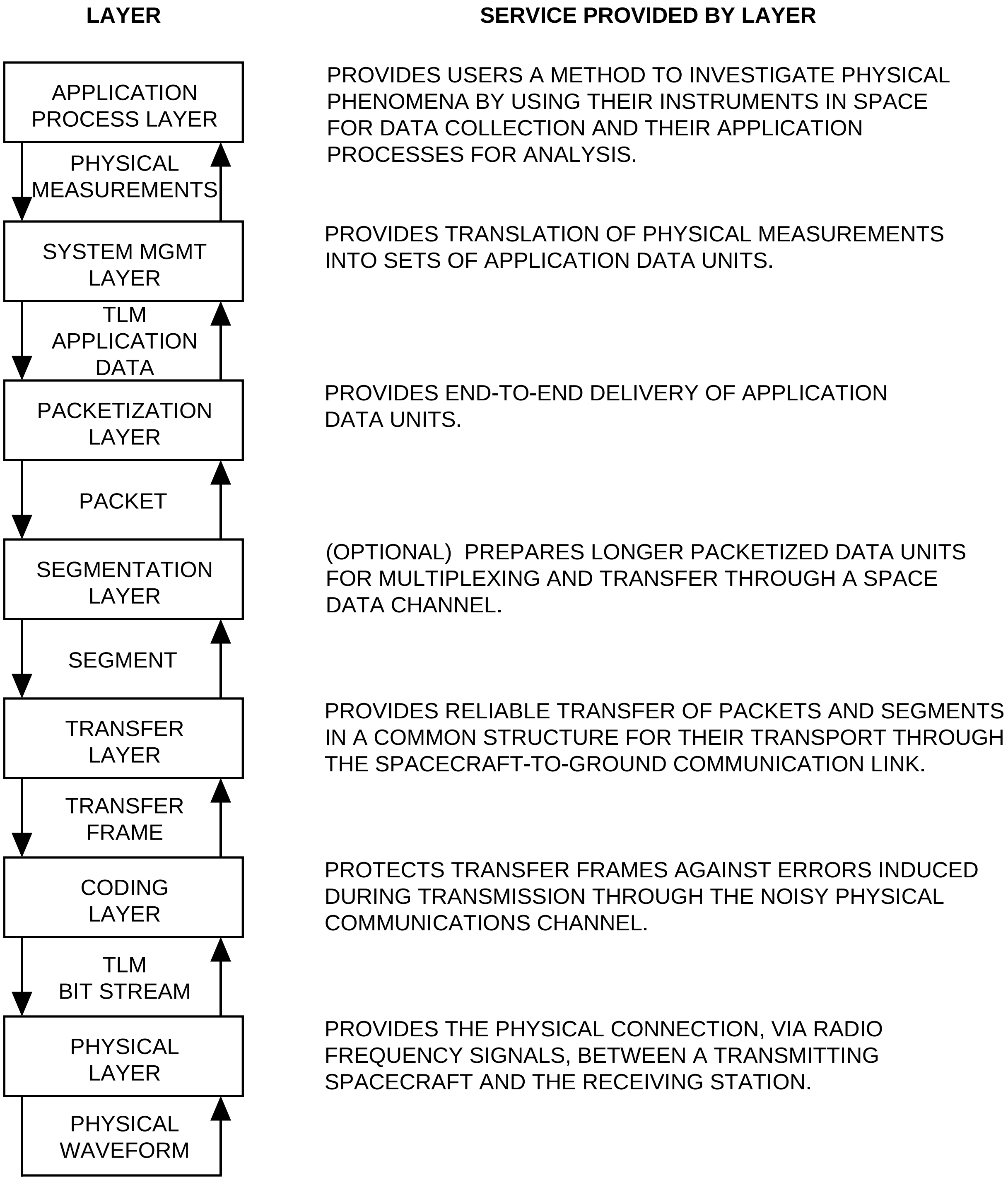
To provide assistance with data flow control, the Packet Telemetry Recommendation provides the capability to segment large packetized transportable data units into smaller communication oriented TM Source Packets (Version 1 format) or **TM Segments** (Version 2 format) for transfer through the space data channel. Consequently, the TM Source Packets and/or TM Segments are of proper size for placement into the data field of the data unit of the next lower layer.

2.2.3 TRANSFER FRAME LAYER

The **TM Transfer Frame** is used to reliably transport Source Packets and Segments through the telemetry channel to the receiving telecommunications network. As the heart of the CCSDS Telemetry System, the TM Transfer Frame protocols offer a range of delivery service options. An example of such a service option is the multiplexing of TM Transfer Frames into **"Virtual Channels"** (VCs).

The TM Transfer Frame begins with an attached frame synchronization marker and is followed by a primary header. The primary header contains frame identification, channel frame count information and frame data field status information.

The transfer frame data field may be followed by an optional trailer containing an operational control field and/or a frame error control field. The first of these fields provides a standard mechanism for incorporating a small number of real-time functions (e.g., telecommand verification or spacecraft clock calibration). The error control field provides the capability for



LAYER	SERVICE PROVIDED BY LAYER
APPLICATION PROCESS LAYER	PROVIDES USERS A METHOD TO INVESTIGATE PHYSICAL PHENOMENA BY USING THEIR INSTRUMENTS IN SPACE FOR DATA COLLECTION AND THEIR APPLICATION PROCESSES FOR ANALYSIS.
SYSTEM MGMT LAYER	PROVIDES TRANSLATION OF PHYSICAL MEASUREMENTS INTO SETS OF APPLICATION DATA UNITS.
PACKETIZATION LAYER	PROVIDES END-TO-END DELIVERY OF APPLICATION DATA UNITS.
SEGMENTATION LAYER	(OPTIONAL) PREPARES LONGER PACKETIZED DATA UNITS FOR MULTIPLEXING AND TRANSFER THROUGH A SPACE DATA CHANNEL.
TRANSFER LAYER	PROVIDES RELIABLE TRANSFER OF PACKETS AND SEGMENTS IN A COMMON STRUCTURE FOR THEIR TRANSPORT THROUGH THE SPACECRAFT-TO-GROUND COMMUNICATION LINK.
CODING LAYER	PROTECTS TRANSFER FRAMES AGAINST ERRORS INDUCED DURING TRANSMISSION THROUGH THE NOISY PHYSICAL COMMUNICATIONS CHANNEL.
PHYSICAL LAYER	PROVIDES THE PHYSICAL CONNECTION, VIA RADIO FREQUENCY SIGNALS, BETWEEN A TRANSMITTING SPACECRAFT AND THE RECEIVING STATION.

Figure 2-1: Layered Telemetry Service Model

detecting errors which may have been introduced into the frame during the data handling process.

The delivery of transfer frames requires the services provided by the lower layers (e.g., carrier, modulation/detection, and coding/decoding) to accomplish its role.

2.2.4 CHANNEL CODING LAYER

Since a basic system requirement is the error-free delivery of the TM Transfer Frames, Telemetry Channel Coding is used to protect the transfer frames against telemetry channel noise-induced errors. Reference [2] describes the CCSDS Recommendataion for Telemetry Channel Coding, including specification of a convolutionally encoded inner channel concatenated with a Reed-Solomon block-oriented outer code (Reference [4]). The basic data units of the CCSDS Telemetry Channel Coding which interface with the layer below are the **Channel Symbols** output by the convolutional encoder. These are the information bits representing one or more transfer frames as parity-protected channel symbols.

The RF channel physically modulates the channel symbols into RF signal patterns interpretable as bit representations. Within the error detecting and correcting capability of the channel code chosen, errors which occur as a result of the physical transmission process may be detected and corrected by the receiving entity.

Full advantage of all CCSDS Telemetry System services could be realized if a Project complied with all CCSDS Recommendations. Alternatively, Projects can interface with any layer of the Telemetry System as long as they meet the interface requirements as specified in the two Recommendations (References [1] and [2]).

Figure 2-2 illustrates how the various telemetry data structures map into one another. There is presently no attempt to define the data structures of the top two layers of the telemetry system; i.e., the Application Process layer and the System Management layer. Telemetry Source Packets may be segmented and placed into the data field of telemetry segments, which are preceded by a header. The Source Packets and/or the Segments are placed into the data field of the Transfer Frame which is preceded by a transfer frame header. If the specified Reed-Solomon code is used in the channel coding scheme, the transfer frame is placed into the Reed-Solomon data space of the Reed-Solomon codeblock, and the codeblock is preceded by an attached synchronization marker.

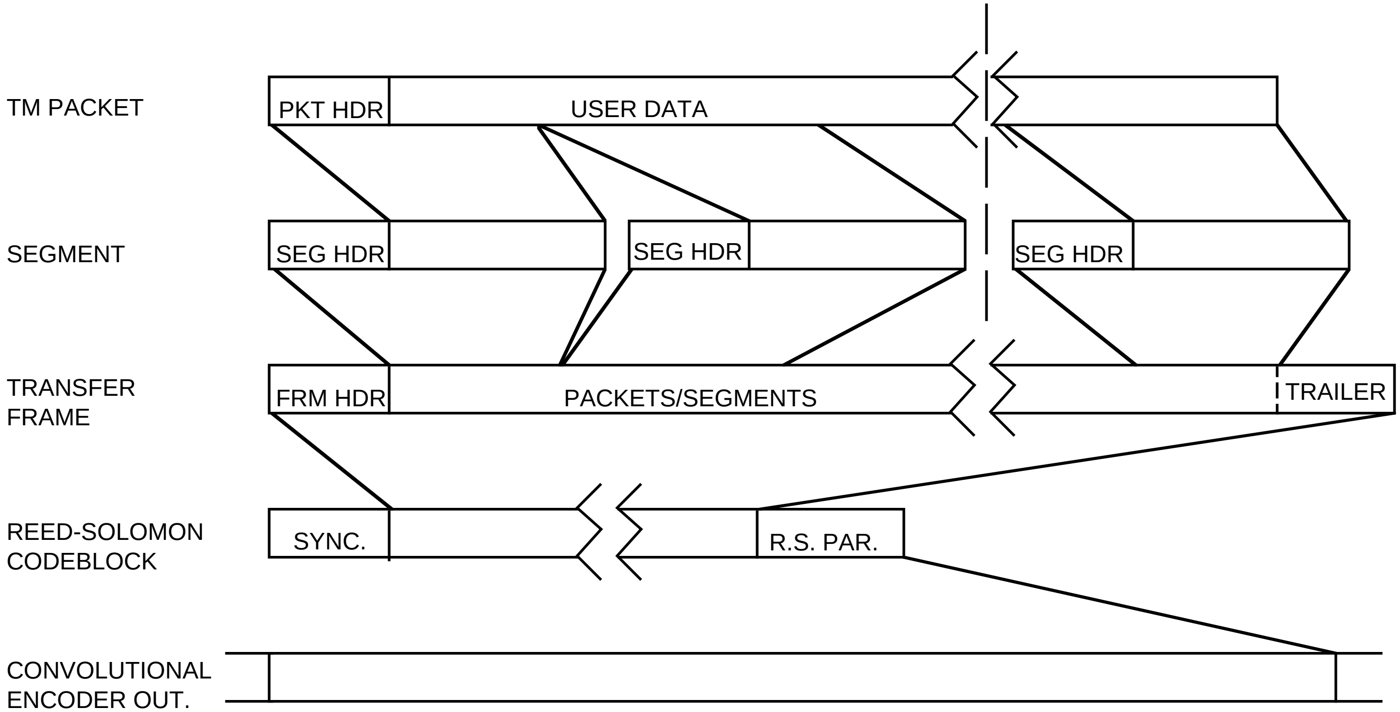


Figure 2-2: Telemetry Data Structures

2.2.5 RELATIONSHIP BETWEEN TELEMETRY AND TELECOMMAND SYSTEMS

A different level of understanding is revealed by considering interactions between the Telemetry System and other systems in the operational environment. In conceptual fashion, Figure 2-3 shows the balanced relationship between the Telemetry System and the uplink Telecommand System. The two systems work hand-in-hand to assure the transfer of user directives from the sending end (traditionally on the ground) to the receiving end (controlled process, device or instrument). Of course, the Telemetry System does a great deal more than simply returning command receipt status information to the sender: its usual function is to provide reliable, efficient transfer of all spacecraft data (housekeeping, sensor readings, etc.) back to users.

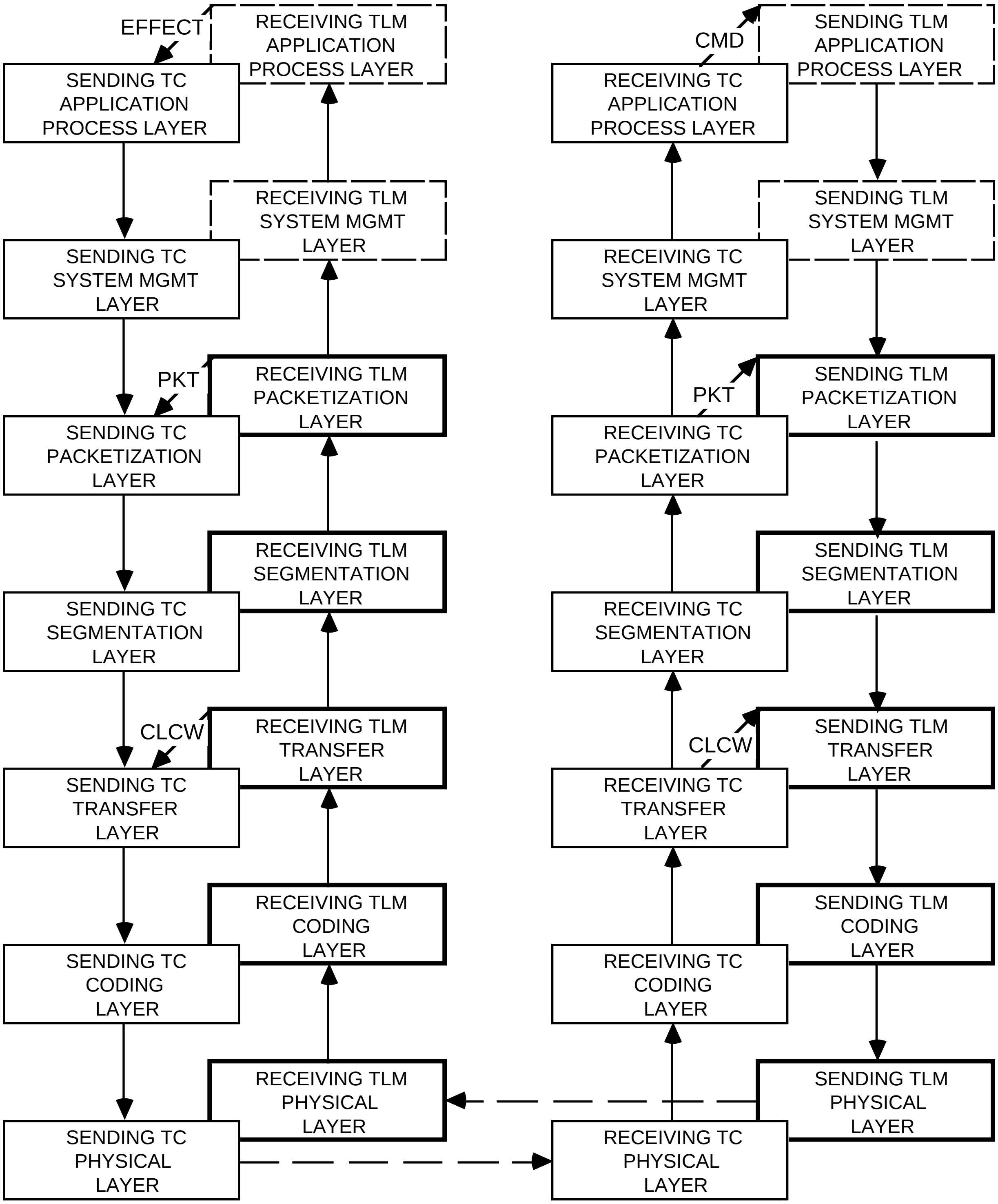

EFFECT
RECEIVING TLM APPLICATION PROCESS LAYER
SENDING TC APPLICATION PROCESS LAYER
RECEIVING TLM SYSTEM MGMT LAYER
SENDING TC SYSTEM MGMT LAYER
PKT
RECEIVING TLM PACKETIZATION LAYER
SENDING TC PACKETIZATION LAYER
RECEIVING TLM SEGMENTATION LAYER
SENDING TC SEGMENTATION LAYER
CLCW
RECEIVING TLM TRANSFER LAYER
SENDING TC TRANSFER LAYER
RECEIVING TLM CODING LAYER
SENDING TC CODING LAYER
RECEIVING TLM PHYSICAL LAYER
SENDING TC PHYSICAL LAYER
CMD
SENDING TLM APPLICATION PROCESS LAYER
RECEIVING TC APPLICATION PROCESS LAYER
SENDING TLM SYSTEM MGMT LAYER
RECEIVING TC SYSTEM MGMT LAYER
PKT
SENDING TLM PACKETIZATION LAYER
RECEIVING TC PACKETIZATION LAYER
SENDING TLM SEGMENTATION LAYER
RECEIVING TC SEGMENTATION LAYER
CLCW
SENDING TLM TRANSFER LAYER
RECEIVING TC TRANSFER LAYER
SENDING TLM CODING LAYER
RECEIVING TC CODING LAYER
SENDING TLM PHYSICAL LAYER
RECEIVING TC PHYSICAL LAYER

Figure 2-3: Telemetry/Telecommand Relationships

3 TELEMETRY SYSTEM DESCRIPTION AND RATIONALE

This section describes the services and protocols characterizing the Telemetry System and presents the rationale for detailed structure of the data units. The section is partitioned into the two major parts of the CCSDS Telemetry System: Packet Telemetry and Telemetry Channel Coding. Within the Packet Telemetry section, discussion is organized according to three main protocol and format areas: 1) TM Source Packet, 2) Source Packet Segmentation, and 3) TM Transfer Frame. The CCSDS Telemetry Channel Coding section is divided into the three main subject coding methods: 1) Convolutional Code, 2) Periodic Convolutional Interleaving, and 3) Reed-Solomon Code.

3.1 PACKET TELEMETRY

3.1.1 INTRODUCTION

Packet Telemetry represents an evolutionary step from the traditional Time-Division Multiplex (TDM) method of transmitting scientific, applications and engineering data from spacecraft sources to users located in space or on Earth. The Packet Telemetry process conceptually involves:

(1) Encapsulating, at the source, observational data (to which may be added ancillary data to subsequently interpret the observational data), thus forming an autonomous "packet of information in real time on the spacecraft.

(2) Providing a standardized mechanism whereby autonomous packets from multiple data sources on the spacecraft can be inserted into a common "frame" structure for transfer to another space vehicle or to Earth through noisy data channels, and delivered to facilities where the packets may be extracted for delivery to the user.

The Packet Telemetry process has the conceptual attributes of:

(1) Facilitating the acquisition and transmission of instrument data at a rate appropriate for the phenomenon being observed.

(2) Defining a logical interface and protocol between an instrument and its associated ground support equipment which remains constant throughout the life cycle of the instrument (bench test, integration, flight, and possible re-use).

(3) Simplifying overall system design by allowing microprocessor-based symmetric design of the instrument control and data paths ("Telecommand Packets in,

Telemetry Packets out") compatible with commercially available components and interconnection protocol standards.

(4) Eliminating the need for mission-dependent hardware and/or software at intermediate points within the distribution networks through which space data flows; in particular, enabling the multi-mission components of these networks to be designed and operated in highly automated fashion, with consequent cost and performance advantages.

(5) Facilitating interoperability of spacecraft whose telemetry interfaces conform to CCSDS guidelines, i.e., allowing very simple cross-strapping of spacecraft and network capabilities between space agencies.

(6) Enabling the delivery of high-quality data products to the user community in a mode which is faster and less expensive than would be possible with conventional telemetry.

Figure 3-1 is a functional diagram of the telemetry data flow from the creation of a data set by an application process operating within a spacecraft "source" (instrument or subsystem), through to the delivery of the same data to a user "sink" (application process) on the ground. Since many of the elements of this flow are presently mission-unique, a primary objective of Packet Telemetry is to define stable, mission-independent interface standards for the communications path within the flow.

3.1.2 TELEMETRY SOURCE PACKET

A Telemetry Source Packet is a data unit which encapsulates a block of observational data which may include ancillary data and which may be directly interpreted by the receiving end application process. Detailed discussion of the format specification for the Telemetry Source Packet is specified in Reference [1]. The Source Packet Format (Version 1), with the addition of a secondary header and packet error control field, is reproduced in Figure 3-2 below for the convenience of the reader.

From the viewpoint of data processing efficiency, the CCSDS strongly recommends that all major fields of all telemetry formats should be an even number of octets. This facilitates efficient internal processing within 16- or 32-bit computers, which are anticipated to be widely used in application processes.

User application data are encapsulated within a packet by prefacing them with a standard label or "primary header", which is used by the data transport system to route the data through the system and to allow the user to reconstruct the original data set. The primary header consists of three main fields: packet identification, packet sequence control and packet length.

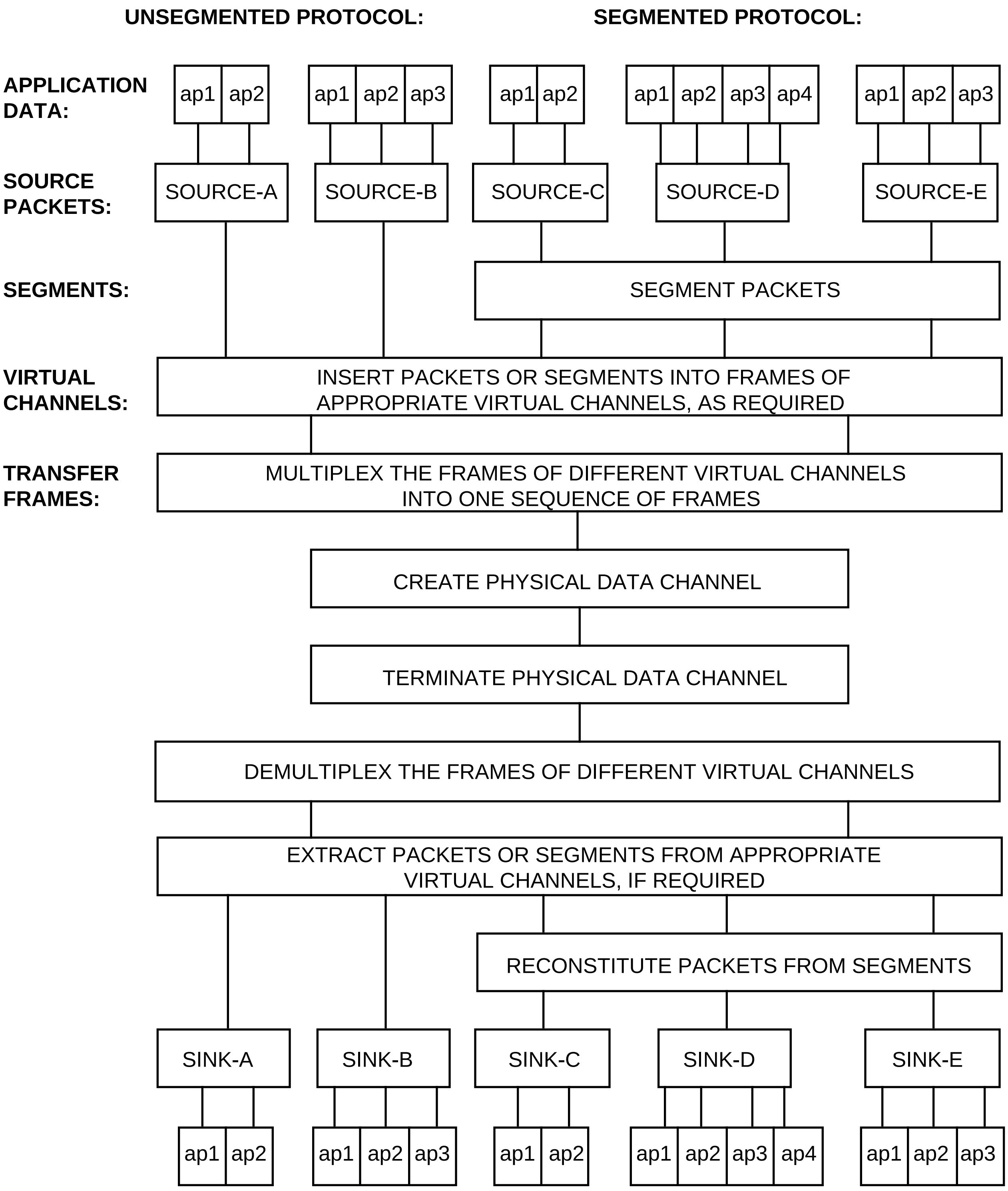
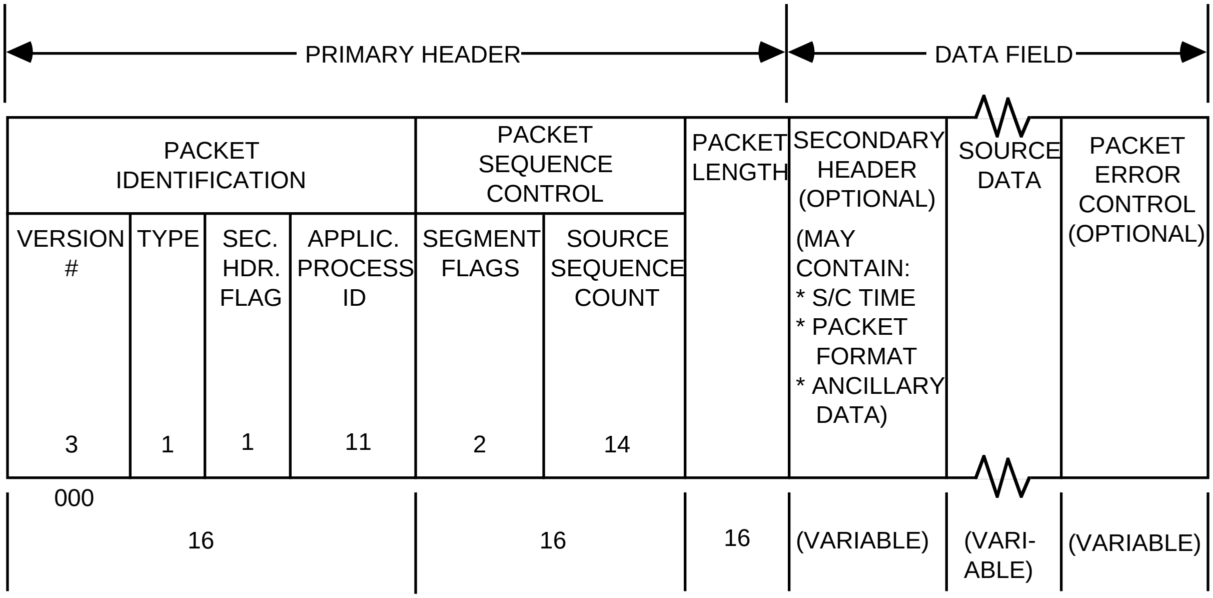

UNSEGMENTED PROTOCOL:
SEGMENTED PROTOCOL:
APPLICATION DATA:
ap1
ap2
ap3
ap4
SOURCE PACKETS:
SOURCE-A
SOURCE-B
SOURCE-C
SOURCE-D
SOURCE-E
SEGMENTS:
SEGMENT PACKETS
VIRTUAL CHANNELS:
INSERT PACKETS OR SEGMENTS INTO FRAMES OF APPROPRIATE VIRTUAL CHANNELS, AS REQUIRED
TRANSFER FRAMES:
MULTIPLEX THE FRAMES OF DIFFERENT VIRTUAL CHANNELS INTO ONE SEQUENCE OF FRAMES
CREATE PHYSICAL DATA CHANNEL
TERMINATE PHYSICAL DATA CHANNEL
DEMULTIPLEX THE FRAMES OF DIFFERENT VIRTUAL CHANNELS
EXTRACT PACKETS OR SEGMENTS FROM APPROPRIATE VIRTUAL CHANNELS, IF REQUIRED
RECONSTITUTE PACKETS FROM SEGMENTS
SINK-A
SINK-B
SINK-C
SINK-D
SINK-E

Figure 3-1: Telemetry Data Flow



PRIMARY HEADER							DATA FIELD		
PACKET IDENTIFICATION				PACKET SEQUENCE CONTROL		PACKET LENGTH	SECONDARY HEADER (OPTIONAL)	SOURCE DATA	PACKET ERROR CONTROL (OPTIONAL)
VERSION #	TYPE	SEC. HDR. FLAG	APPLIC. PROCESS ID	SEGMENT FLAGS	SOURCE SEQUENCE COUNT		(MAY CONTAIN: * S/C TIME * PACKET FORMAT * ANCILLARY DATA)		
3	1	1	11	2	14				
000									
16				16		16	(VARIABLE)	(VARI-ABLE)	(VARIABLE)

Figure 3-2: "Source Packet" (Version 1) Format

3.1.2.1. Packet Identification

Version Number. The version number is the first of four sub-fields of packet identification. This sub-field explicitly indicates the version of the formatted packet, and its length of three bits allows eight different versions to be identified. While only two versions are currently defined, this arrangement allows a reasonable growth capability to support future needs. However, in the interest of constraining the proliferation of standards, additional versions will be discouraged unless it can be demonstrated that the current versions are truly inadequate.

Type. The second sub-field is a one-bit identifier to signal that this packet is a "Telemetry" packet and not a "Telecommand" packet. It is always set to "zero" for Telemetry packets.[1]

Secondary Header Flag. The third sub-field is a one-bit secondary header flag. The CCSDS recognizes that users may need a means of encapsulating ancillary data (such as time, internal data field format, spacecraft position/attitude, etc.) which may be necessary for the interpretation of the information contained within the packet. Therefore, this flag, when set to one, indicates that a secondary header follows the primary header.

[1] In the first issue of Reference [1] (May 1984) this field was described as a "reserved spare" and was, by convention, set to zero for Telemetry. In Issue 2 (January 1987), the value of the field has not changed, but its function has been established.

Application Process ID. The last sub-field in the packet identification field is used to uniquely identify the originating source packet application process. In conventional free flyer spacecraft, source data (packets) are traditionally routed to the corresponding user application process on Earth; this field could then also be used as a "destination ID"[2] Eleven bits are allocated to the Application Process ID, permitting identification of up to 2048 separate application processes per spacecraft, sufficient for any envisioned free flyer spacecraft. For positive identification, one can consider this sub-field an extension of the spacecraft ID, which is in the Transfer Frame primary header (see Figure 3-4).

3.1.2.2 Packet Sequence Control

Segmentation Flags. The first sub-field of the packet sequence control field is called "Segmentation Flags", and provides for a logical representation of four types of segmentation status. These flags identify whether the source data field contains the first, continuing or last segment of a source packet, or if it contains no segment (meaning it contains a complete set of source application data). Refer to Section 3.1.3 for an explanation of segmentation.

Source Sequence Count. This second sub-field provides for each packet to be numbered in a sequential manner, thus providing a method of checking the order of source application data at the receiving end of the system. It is normally used for ground accounting purposes to measure the quantity, continuity and completeness of the data received from the source. The field provides a straight sequential count to modulo 16,384. Longer-term unambiguous ordering (beyond 16,384 packets) may be accomplished by associating the measurement time code contained within the packet with the Source Sequence Count.

3.1.2.3 Packet Length. The last major field of the primary header delimits the boundaries of the packet. It is a count of the number of octets in the packet beginning with the first octet after the 48-bit primary header and ending with the last octet of the packet. The 16-bit field allows packet lengths up to 65,536 octets (not counting the 48-bit primary header). This packet limit was a compromise between the majority of users (who produce medium-size packets) and the few users who may produce exceptionally long packets. Placing a reasonable limit on packet size helps avoid the flow control problems associated with very long packets, and eliminates the overhead penalty of a larger length field for the great majority of packet producers.

[2] As such, the need for separate destination ID does not seem apparent. However, if users require one or more different destination IDs, these could be placed in the secondary header.

3.1.2.4 Data Field. The remainder of the packet may consist of any data desired, although some suggestions are provided by the Recommendation. The total length of all subsequent data should be an even number of octets (a multiple of 16 bits) for efficiency in computer processing. In addition, Figure 3-2 indicates three possible sub-fields: secondary header, source application data and a packet error control field.

Secondary Header. A secondary header may be desirable for providing any ancillary data generated by another application process (time, spacecraft position/attitude) or for providing an internal data field format. The CCSDS has not developed a recommendation for the format, but in order to allow for the future standardization of the secondary header, the most significant bit (bit 0) of the first octet of each secondary header shall be set to "0" to signify a non-CCSDS-defined secondary header.

Source Data. Following the secondary header, the source data sub-field contains source application data generated by the application process identified in the primary header. For efficiency in computer processing, this sub-field should be a multiple of 16 bits.

Packet Error Control. At the discretion of the user, an optional error detection code may be included at the end of the packet in order to verify that the overall integrity of the message has been preserved during the transport process. The particular implementation of such an error detection code, including the selection of the encoding polynomial and the length of the field, is left to the user or to the local agency.

3.1.3 FLOW CONTROL MECHANISMS

Space telecommunications systems are usually constrained by the capacity or the bandwidth of the telecommunications channel which connects the spacecraft to the data capture element located in space or on Earth. Flow control becomes crucial when multiple users must share the same telecommunications channel. The Telemetry System must ensure that all sources have proper access to this common resource frequently enough to ensure timely delivery as well as to control the need to buffer data while other sources are being serviced. Long source packets may present flow control problem if they monopolize the data channel for unacceptable periods of time while forcing other sources to implement unreasonably large local buffering of their data. Several alternative solutions to the problem of flow control are presented in the Recommendation. These are discussed below, and are summarized in Annex C of this report.

3.1.3.1 Virtual Channelization. One solution to the flow control problem is to assign each source (which generates long packets) its own "Virtual Channel". This is accomplished by inserting these packets into specially identified Transfer Frames. These dedicated frames form a "Virtual Channel" and may be interleaved with other frames containing data from other users. Detailed discussion of Virtual Channelization occurs in Section 3.1.4.

3.1.3.2 Source-Internal Segmentation: Source Packet (Version 1). Another solution to the flow control problem is accomplished entirely within the source, whereby it manipulates its own "segmentation" flags when producing packets. That is, if the source is producing a very long message, or data unit, it breaks the unit into segments that can fit into working-size Version 1 packets. This way, the spacecraft data system and ground see and handle normal packets whose data fields actually contain segments of a long message whose reassembly by the application can be assured by use of the packet sequence control described below.

Packet Identification. Except for the Secondary Header Flag[3], the Packet Identification fields of each of the source packets created from the original very long message are identical.

Packet Sequence Control. The packet containing the first segment of the original very long message is identified by setting the segmentation flags in the primary header to 0,1. The source sequence count value is incremented by one for each packet of the sequence. The actual value for the first segment depends on the running count at the time the first segment is to appear.

Packets containing continuation segments are identified by setting the segmentation flags to 0,0. The sequence of packets is identified by incrementing the source sequence count for each packet.

The packet containing the last segment of the original very long message is identified by setting its segmentation flags to 1,0.

Packet Length. Since the packet length field is used to point to the beginning of the next packet for purposes of extraction from the transfer frame, the packet length must always refer to the length of the source packet being handled. The total length of the original very long message can be provided by the user through private, internal message labeling.

3.1.3.3 Spacecraft Segmentation: Source Packet (Version 1). Instead of source-internal segmentation, another alternative is a more centralized approach to data flow control wherein the spacecraft data system performs the segmentation. Spacecraft segmentation is accomplished by breaking up a completely formed original long source packet and inserting the pieces into newly generated, shorter Version 1 source packets; but in this case the shorter source packets are created by the spacecraft data system instead of the source itself, and carry "S/C data system" Application Process ID.

[3] For example, the Secondary Header Flag may indicate a secondary header present in the first packet of the sequence and not in subsequent packets of the sequence.

Packet Identification. The application process ID in the packet identification field indicates that the spacecraft data system is generating the source packets containing the segments.

Packet Sequence Control. The segmentation flags are set as described in the previous section. The source sequence count sub-field contains the count value generated by the spacecraft data system and is incremented for each segment produced. (Note: the original long packet sequence count value remains hidden in the data field of the first packet generated by the spacecraft data system.)

Packet Length. As in the previous section, the packet length field indicates the length of the newly generated packet.

3.1.3.4 Spacecraft Segmentation: Telemetry Segment (Version 2). The segmentation options discussed above utilize the source packet (version 1) format, in which the length is always based on the length, in octets, of the data field (packet or segment) which is transmitted, and the sequence count increments once per packet generated by a given application process. When a long packet (version 1) requires segmentation, the monotonically increasing nature of the source sequence count, during the source packet generation process, may be disrupted.

For those missions which require the source sequence count for a given application process to increase without any gaps in the sequence, another formatting option exists. "Version 2 of the packet format, called a "Telemetry Segment", is a format within which the length field in the data unit defines the **length of an original packet that remains to be transmitted,** and the sequence count field remains static because it refers to the numbering of the original source packet generated by its application process. The length and sequence count of the data unit being transmitted are, therefore, semantically different between the two versions.

It is assumed that Telemetry Segments (Version 2) are always generated by an application process other than the original application process. In most cases, such Telemetry Segments will be generated by the spacecraft data system.

The Telemetry Segment (Version 2) structure is shown in Figure 3-3.

Segment Identification. When a long source packet (version 1) is segmented using the Telemetry Segment protocol, the packet ID field is modified only by changing the version number sub-field to indicate "version 2". This implies that a separate application process is doing the segmentation and, therefore, the application process ID sub-field contains the value of the original application process.

Segment Sequence Control. The protocol for the segmentation flags sub-field is the same as for the version 1 format except that the sequence count sub-field indicates the count of the

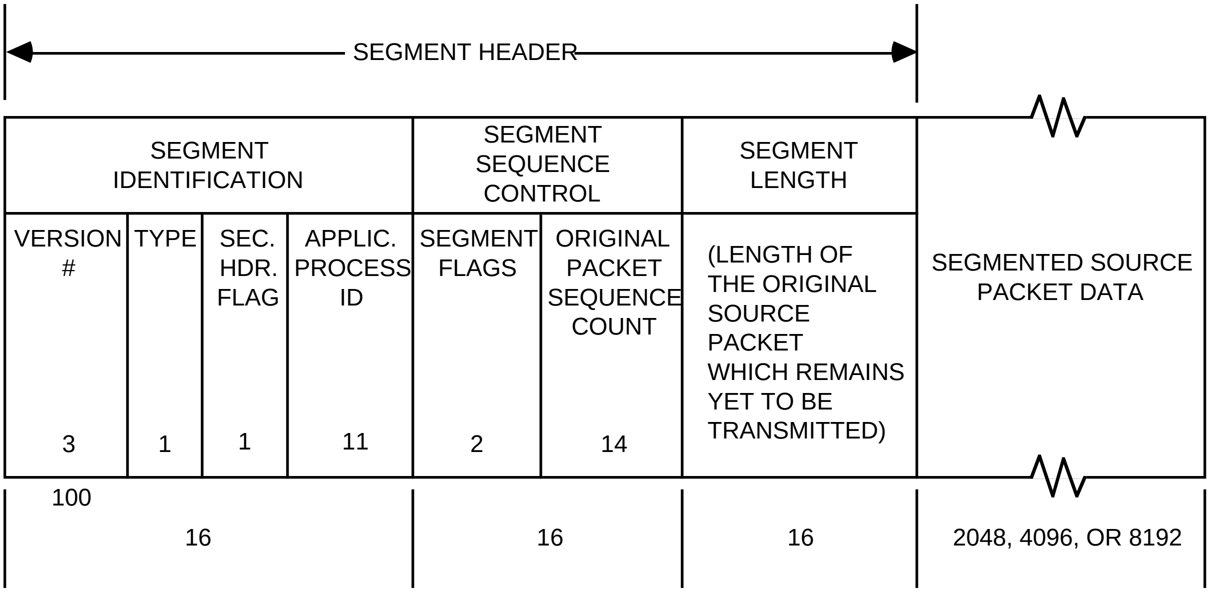


Figure 3-3: Telemetry Segment (Version 2) Format

original long packet being segmented and is not incremented for each segment generated. As such, it would seem as though each segment cannot be uniquely identified, but in fact the following fields do provide mechanism for assigning a "serial number" to each segment. The serial number may then be used to recombine segments should their natural order be disturbed during transmission or the data handling process.

Segment Length. Instead of indicating the length of the segment, the version 2 format segment length field is based on the length of data (in octets) from the original long packet (including that contained within the segment) **which remains yet to be transmitted.** The length of the segment is a fixed value (256, 512 or 1024 octets) for each Virtual Channel and is specified in the Transfer Frame header (rationale in Section 3.1.4).

Since the fixed segment lengths are defined to be binary values of octets, by utilizing the decrementing length approach, the value of the segment length field will decrease in binary countdown fashion as successive segments are transmitted. This information provides a "serial number" for the segment which may be used to recombine segments should their natural order be disturbed during transmission. An example of this process is presented in Section 4.3.1 of Reference [1].

3.1.4 TELEMETRY TRANSFER FRAME

The source packet data structures described in the previous sections are unsuitable for transmission directly through the communication links which interconnect the spacecraft and data capture element in space or on Earth. They must be embedded within a data transfer structure which provides reliable, error-controlled transfer through the media. The CCSDS has developed such a data structure, the telemetry "Transfer Frame", which has a fixed length for a given mission or spacecraft. The attributes of the Transfer Frame and its supporting rationale will follow during the discussion of the Transfer Frame format. Figure 3-4 illustrates the telemetry Transfer Frame format.

3.1.4.1 Synchronization Marker. Attached to the beginning of the Transfer Frame primary header is a 32-bit frame synchronization marker which is used by the receiving network to acquire synchronization with the frame boundaries after transmission through the data channel. A 32-bit synchronization pattern is selected because it provides very good synchronization qualities in a noisy channel environment. The 32-bit pattern is also double-octet compatible with 32-bit computers. The particular bit pattern and its performance characteristics are found in References [1] and [5].

In conjunction with the selection of the 32-bit marker, the Recommendations currently require that all Transfer Frames in a single physical data channel in a given mission be of constant length. When the frame is of fixed length, conventional "flywheeling" techniques may be used to maintain frame synchronization in a noisy environment.

The maximum distance from one attached sync marker to the next when using the maximum-length Transfer Frame (8920 bits), Reed-Solomon check symbols (1280 bits), and sync marker (32 bits) is 10,232 bits.

3.1.4.2 Frame Identification. The first major field of the Transfer Frame primary header is the frame identification field.

Version Number. Only one version of the Transfer Frame has been defined by the CCSDS, although this 2-bit field allows growth to four. The "version" refers to the frame structuring principles which are described in this section. Given the small number of tracking networks, as opposed to the number of end users (packet creators), and the flexibility built into this version to meet future needs, the size of the field is considered adequate.

Spacecraft ID. The spacecraft identification field provides for positive identification of the spacecraft which generated the Transfer Frame. The 10 bits assigned to spacecraft identification allows up to 1024 separate positive IDs. Spacecraft IDs are assigned per the procedures in Reference [6] by the CCSDS, and analysis (Reference [7]) has shown that under those procedures 1024 is an adequate number for future needs.

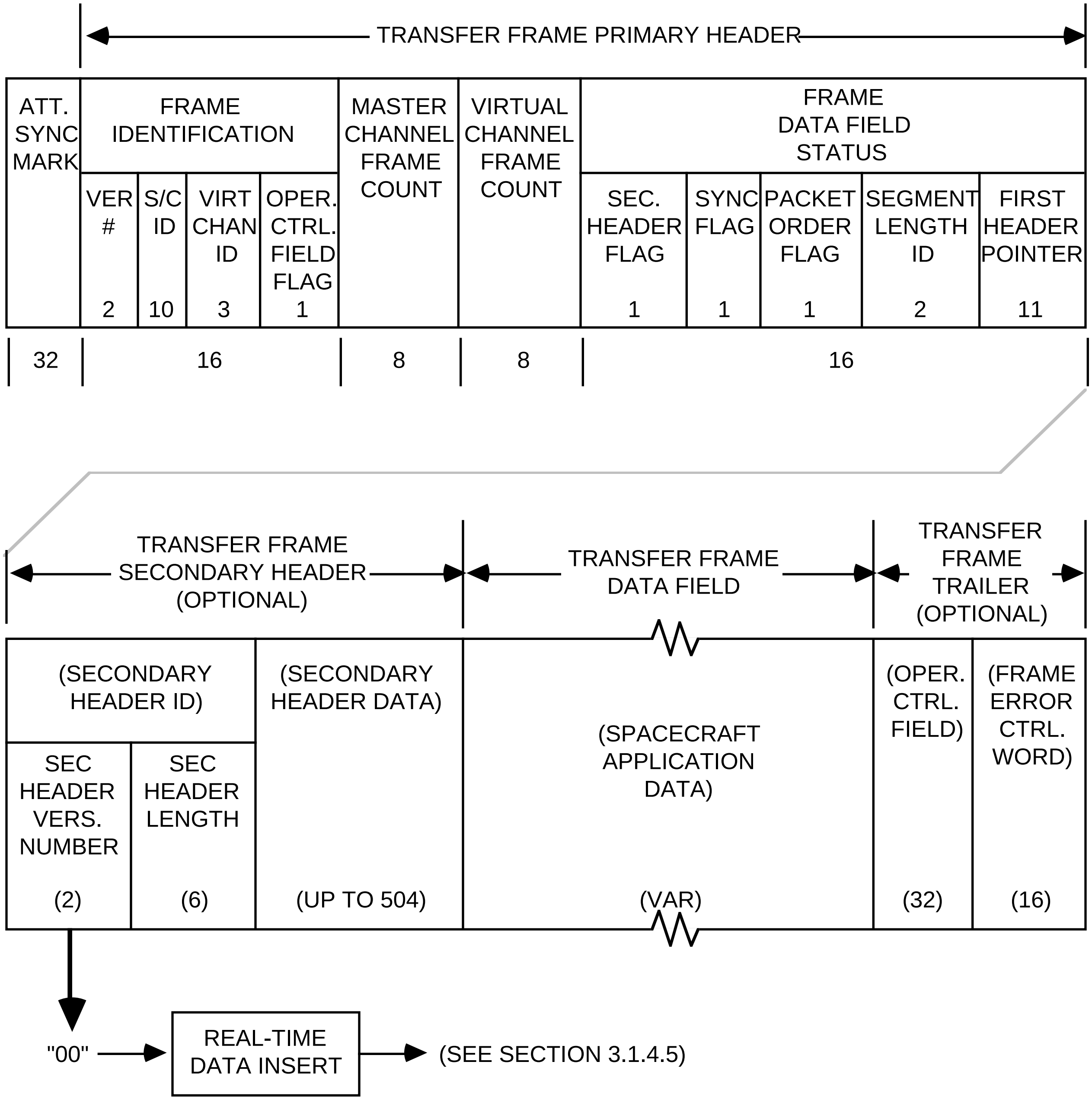


Figure 3-4: Telemetry Transfer Frame Format

Virtual Channel ID. This three-bit sub-field allows up to eight "Virtual Channels" to be run concurrently on a particular physical data channel. Frames from different Virtual Channels are multiplexed together on the telecommunications channel, and, with this identifier in each frame, can be easily split apart after receipt at the ground. Virtual Channels can be used for a variety of purposes such as flow control to prevent long packets from "hogging" the channel; selecting out different types of data for stream splitting at the ground (e.g., when low-rate engineering data must be split out from multiplexed high-rate science data upon receipt so it can be forwarded over a capacity-constrained real-time ground data link) or when different levels of data quality are to be accommodated for different types of data (in which case error protection may be applied to certain Virtual Channels but not others). Eight Virtual Channels are considered sufficient to provide adequate flexibility for envisioned future free flyer spacecraft.

Operational Control Field Flag. The last bit of the frame identification field, when set to one, signals the presence of the 32-bit operational control field, which is contained within the frame trailer. The information in this field, discussed later in Section 3.1.4.7, is defined to provide a standardized spacecraft reporting mechanism for spacecraft telecommanding.

3.1.4.3 Master Channel and Virtual Channel Frame Count. The next two fields provide a running count of the number of frames transmitted. These counters provide a degree of data accountability (for short duration data outages), the ambiguity level being defined by the field lengths.

Master Channel Frame Count. This 8-bit field provides sequential count (modulo 256) of the number of frames transmitted by a single physical spacecraft data channel. The counter is long enough to provide a reasonable probability of detecting a discontinuity, in a sequence of frames, when the physical channel is briefly interrupted. If such a discontinuity does occur, the Virtual Channel accounting process can provide a greater probability of detecting the number of missing frames.

Virtual Channel Frame Count. The following 8-bit field provides accountability for each of the eight independent "Virtual Channels". This field is used with the "Virtual Channel ID" sub-field to provide accountability via a sequential count (modulo 256). The rationale for the counter ambiguity level is the same as for the master channel frame counter. If only one "Virtual Channel" is incorporated for a given mission, both the Virtual Channel Frame Counter and the Master Channel Frame Counter must increment once per generated Transfer Frame (i.e., the two fields should not be concatenated into a master frame counter). This is because the ground facilities would normally be designed to handle the general case of spacecraft with multiple Virtual Channels.

3.1.4.4 Frame Data Field Status. The "frame data field status" field provides control information which allows the receiving end to extract and reconstitute packets and/or segments.

Secondary Header Flag. The first sub-field indicates the presence or absence of the optional secondary header. If its presence is so indicated, the secondary header must appear in every frame transmitted through a physical data channel, and its length must also be fixed. Rationale for this requirement is provided later in the discussion (Section 3.1.4.5) about the secondary header.

Synchronization Flag. This flag indicates whether or not the packet or segment data units are inserted into the Transfer Frame data field on octet boundaries. If they are, then they are said to be "synchronously inserted" (packet octet boundaries align with frame octet boundaries) and the extraction technique (pointing to specific octet) is valid. If the flag indicates "asynchronous" data insertion (i.e., an unstructured (non-packetized) data contents or packets inserted without regard to octet boundaries), then the Transfer Frame layer at the receiving end will not be able to reconstitute the original data sets without additional knowledge.

Packet Order Flag. This flag indicates whether the sequence count order of the contained packet or segment is increasing (forward) or decreasing (reverse). This has important implications when tape recorded data are played back opposite to their recorded direction. When this is the case, the spacecraft electronics re-justifies the BIT DIRECTION of each packet/segment so each packet or segment individually flows in the forward direction and its header can be read to allow proper packet extraction from the Transfer Frame. Even though the playback packets appear individually to flow the same as the rest of the data, the sequence of packets will be running backwards in time, as indicated by the decreasing sequence counter. A discussion of various options for handling tape recorded data is contained in Annex B.

Segment Length ID. The segment length identifier sub-field identifies which of three fixed segment lengths are contained within the data field of the standard Version 2 Telemetry Segment. The lengths are fixed in order to provide a method of serializing each Telemetry Segment, as explained in Section 4.3.1 in Reference [1]. The 2-bit flag allows for indication of three different lengths (2048, 4096 or 8192 bits) or an indication that the Version 2 Telemetry Segment is not being used on this Virtual Channel. Three lengths provide efficient flow control for the types of data and missions envisioned. Shorter lengths are not considered because the overhead becomes unacceptably large, while higher values are not considered because virtual channelization becomes a more effective flow control method.

First Header Pointer. The first header pointer sub-field points directly to the location of the starting octet of the first packet or segment header structure within the frame data field. It counts from the end of the primary header (secondary header if present) and effectively delimits the beginning of the first packet/segment. The packet/segment length field, in turn, delimits the beginning of the next packet/segment, and so on. Since the pointer counts octets, this feature works only when the headers are aligned with octet boundaries, i.e., when the packet/segment data are synchronously inserted (data field synchronization flag set to zero). The eleven bits allocated to the pointer allow for a count to 2048 octets, which exceeds the count required to point to an octet at the end of the data field. Special pointer values are used to denote:

(1) No packet/segment header is contained in this frame, but there is valid data; or
(2) No valid data is contained in this frame ("idle channel").

3.1.4.5 Frame Secondary Header (Optional). An optional secondary header is provided for users who desire a means for deterministically inserting real-time data (e.g., Time-Division-Multiplexed data) which may be required for spacecraft monitoring and control applications.

When the secondary header presence is indicated by the secondary header flag, its length must be of a fixed value and must appear in every frame transmitted through a physical channel. Given the requirement for fixed Transfer Frame length, a fixed secondary header length simplifies data processing and packet extraction at the receiving end.

Secondary Header ID. The first part of the secondary header has two sub-fields. The first is the Secondary Header Version Number, a 2-bit field allowing four versions (or structuring rules). Only one version is currently defined by the CCSDS. This provides for a reasonable future growth capability.

The second sub-field, Secondary Header Length, indicates what length has been selected for the secondary header. This 6-bit sub-field provides a binary count of the total number of octets contained within the entire Transfer Frame secondary header (including the ID field itself, which is one octet in length). This limits the total secondary header length to 64 octets (512 bits) which is considered adequate for currently understood applications.

Secondary Header Data. This sub-field contains up to 504 bits of user specified data.

3.1.4.6 Transfer Frame Data Field. The Transfer Frame data field contains an integral number of octets of data (e.g., Source Packets and/or Telemetry Segments) to be transmitted from the spacecraft to the receiving element. The maximum length of this field depends on which optional fields are implemented. As discussed in Reference [2], if frame lengths shorter than the 8920-bit maximum are implemented and the frame is encoded using the recommended Reed-Solomon algorithm, then the length of the frame data field must be selected, bearing in mind the constraint that "Virtual Fill" (see Annex A) must occur in fixed increments. This is necessary in order to simplify data processing at the receiving end. This field may also accommodate an unstructured bit stream (not necessarily packetized) as its data contents. In such a case, standard data extraction services would not be provided.

3.1.4.7 Transfer Frame Trailer (Optional). An optional Transfer Frame trailer is provided and is divided into two main fields, each of which is optional.

Operational Control Field. The presence or absence of the operational control field is indicated by a flag located in the frame identification field of the primary header. When present, this field facilitates closed-loop reporting of standardized real-time functions. The first bit (bit 0) of this field indicates the type of report and is currently set to zero. This signifies that this field contains a "Command Link Control Word" which is used for acceptance reporting of spacecraft command activity and certain other front-end telecommunication status. This reporting mechanism is fundamental to the automated Telecommand System which is summarized in Reference [8]. The standardized internal format of the Command Link Control Word is fully defined in Reference [9].

Frame Error Control Word. When present, this field occupies the two trailing octets of the Transfer Frame. Its presence or absence is implicitly defined from the spacecraft identifier, and thus must or must not appear in all frames of a given spacecraft ID. It provides the capability for detecting errors which may have been introduced into the frame during the data handling processes. Its presence is mandatory if the Transfer Frame is NOT Reed-Solomon encoded but is optional if the frame is synchronously contained within the data space of a Reed-Solomon codeblock.

A Cyclic Redundancy Code (CRC) has been selected for this purpose because of its effectiveness and simplicity, and is defined and specified in Reference [1], Section 5.5.2. Parity is generated over the entire Transfer Frame (less the final 16 bits), and the 16 bits of parity checks are then appended to complete the frame. It should be noted that in the 1984 issue of the Packet Telemetry Recommendation, the frame was defined to include the "attached sync marker"; in the 1987 issue, the frame definition was changed to exclude the marker, but it was still considered to be "attached". To maintain compatibility with already-built systems, it was necessary to allow for two options over which the CRC is applied: that is, it may include the sync marker or it may exclude it. Since the marker pattern is always known, the preferred choice is to omit the marker when encoding. This is explained in Reference [1], Section 5.5.2, and details of the encoding and decoding process are contained in Annex D of this book.

3.2 TELEMETRY CHANNEL CODING

3.2.1 INTRODUCTION

Channel coding is a method by which data can be sent from a source to a destination by processing data so that distinct messages are easily distinguishable from one another. This allows reconstruction of the data with low error probability.

In spacecraft, the data source is usually digital, with the data represented as a string of zeroes and ones. A channel encoder (or simply "encoder") is then a device that takes this string of binary data and produces a modulating waveform as output. If the channel code is chosen correctly for the particular channel in question, then a properly designed decoder will be able to reconstruct the original binary data even if the waveforms have been corrupted by channel noise. If the characteristics of the channel are well understood, and an appropriate coding scheme is chosen, then channel coding provides higher overall data throughput at the same overall quality (bit error rate) as uncoded transmission - but with less energy expended per information bit. Equivalently, channel coding allows a lower overall bit error rate than the uncoded system using the same energy per information bit.

There are other benefits that may be expected from coding. First, the resulting "clean" channel can benefit the transmission of compressed data. The purpose of data compression schemes is to map a large amount of data into a smaller number of bits. Adaptive compressors will continually send information to direct a ground decompressor how to treat the data that follows. An error in these bits could result in improper handling of subsequent data. Consequently, compressed data is generally far more sensitive to communication errors than uncompressed data. The combination of efficient low error rate channel coding and sophisticated adaptive data compression can result in significant improvement in overall performance (References [10],[11] and [12]).

Second, a low bit error rate is also required when adaptive telemetry is used. Adaptive telemetry is much like adaptive data compression in that information on how various ground processors should treat the transmitted data is included as part of the data. An error in these instructions could cause improper handling of subsequent data and the possible loss of much information.

Third, low error probability telemetry may allow a certain amount of unattended mission operations. This is principally because the operations systems will know that any anomalies detected in the downlink data are extremely likely to be real and not caused by channel errors. Thus, operators may not be required to try to distinguish erroneous data from genuine spacecraft anomalies.

In a typical space channel, the principal signal degradations are due to the loss of signal energy with distance, and to the thermal noise in the receiving system. The codes described in Reference [2] can usually provide good communication over this channel. An additional degradation, caused by interference from Earth-based pulse radars, may occur for users of the Tracking and Data Relay Satellite System (TDRSS). Such users may consider adding periodic convolutional interleaving (PCI) to their coding system; in this case, they should carefully analyze the effects of the PCI on their systems.

If interagency cross support requires one agency to decode the telemetry of another, then the codes recommended in Reference [2] should be used. A block diagram of the recommended coding system appears in Figure 3-5.

The relative performance of the various codes in a Gaussian channel is shown in Figure 3-6 (from Reference [13]). Here, the input is constrained to be chosen from between two levels, because biphase modulation is assumed throughout this recommendation. These performance data were obtained by software simulation and assume that there are no synchronization losses. The channel symbol errors were assumed to be independent. This is a good assumption for the deep space channel. Also, infinite interleaving was assumed in the Reed-Solomon code. It is clear from the figure that the convolutional code offers a coding gain of about 5.5 dB over an uncoded system at decoded bit error rate of 10^{-5}. The use of the outer Reed-Solomon code results in an additonal 2.0 dB of coding gain. Note that Figure 3-6 does not necessarily represent the performance of the TDRSS channel.

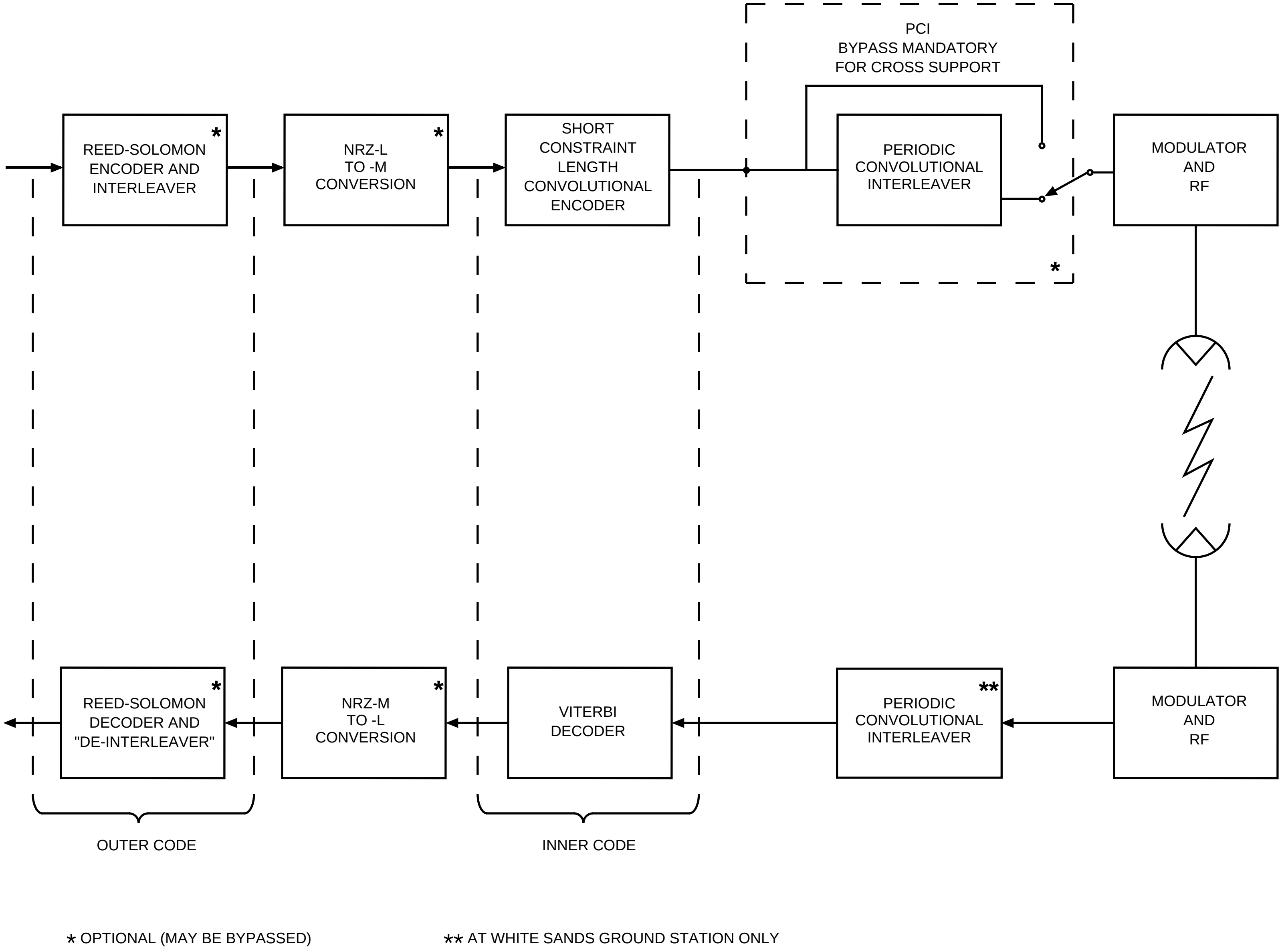

PCI
BYPASS MANDATORY
FOR CROSS SUPPORT
REED-SOLOMON ENCODER AND INTERLEAVER *
NRZ-L TO -M CONVERSION *
SHORT CONSTRAINT LENGTH CONVOLUTIONAL ENCODER
PERIODIC CONVOLUTIONAL INTERLEAVER
*
MODULATOR AND RF
REED-SOLOMON DECODER AND "DE-INTERLEAVER" *
NRZ-M TO -L CONVERSION *
VITERBI DECODER
PERIODIC CONVOLUTIONAL INTERLEAVER **
MODULATOR AND RF
OUTER CODE
INNER CODE
* OPTIONAL (MAY BE BYPASSED)
** AT WHITE SANDS GROUND STATION ONLY

Figure 3-5: Coding System Block Diagram

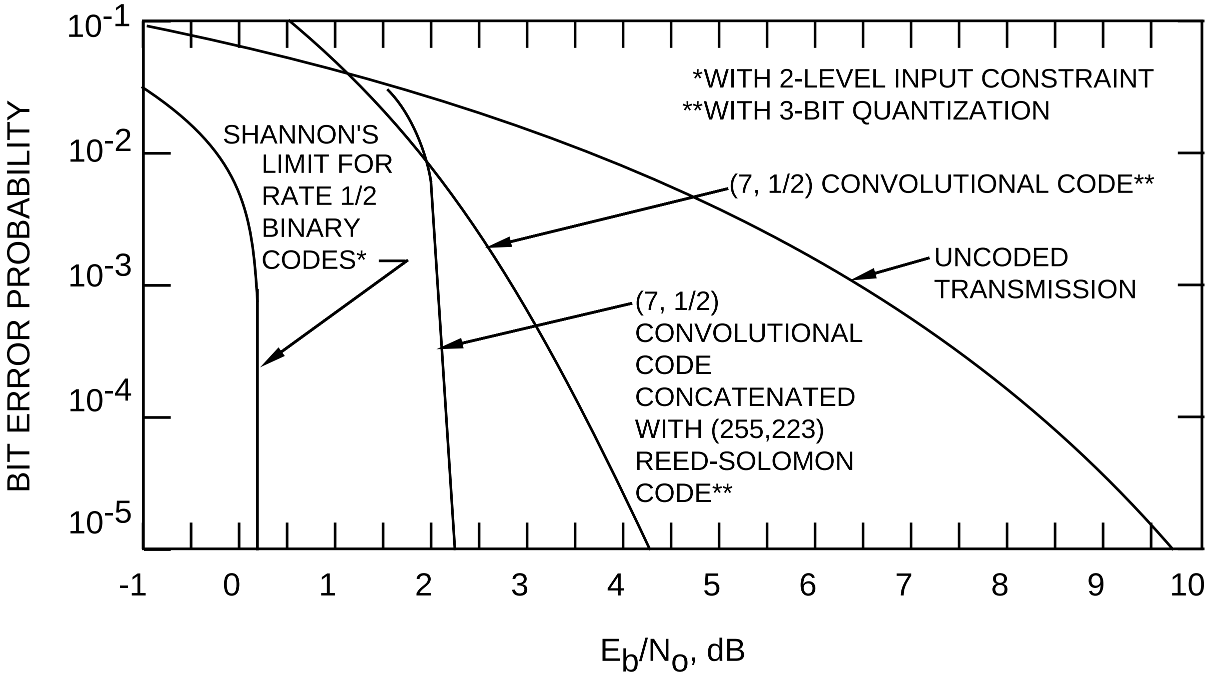


Figure 3-6: Performance of Various Codes in a Gaussian Channel

Performance gains higher than 2.0 dB over the convolutional code alone are provided by the concatenated channel for error rates lower than 10^{-5} and if receiver tracking losses are accounted for (References [10],[14] and [15]). The net throughput improvement provided by the combination of data compression and the concatenated channel is dramatic[4] (Reference [11]).

These codes are included in the CCSDS Recommendation because they represent state-of-the-art coding technology and provide substantial coding gain over an uncoded system. They have already been incorporated, or are planned to be incorporated, into missions of member agencies of the CCSDS.

The next three sections explain the choice of the codes and the parameters of each code in more detail.

[4] This was demonstrated by the Voyager 2 spacecraft in 1986 during the Uranus encounter.

3.2.2 CONVOLUTIONAL CODE

A rate 1/2, constraint length 7 convolutional code with Viterbi (maximum likelihood) decoding is already a standard for both NASA and ESA. It has been used in several missions and has demonstrated the expected coding gain.

The encoder for this code is extremely simple. It consists of a shift register of length six and some exclusive OR gates that implement the two parity checks. The two checks are then multiplexed into one line. This means that the encoder can be made small and that it dissipates very little power. These are good attributes for spacecraft hardware.

It has been customary to invert one or the other parity check in the encoder. This is to ensure that there are sufficient transitions in the channel stream for the symbol synchronizer to work in the case of a steady state (all zeroes or all ones) input to the encoder.

Historically, ESA, NASA-GSFC and NASA-JPL have each used a different ordering of the two parity checks or has inverted a different parity check. Performance is not affected by these minor differences. While interim cross support of these different conventions may require minor differences in ground station equipment, all agencies are encouraged to adopt for all facilities the single convention described in Reference [2], which is the NASA-GSFC convention.

3.2.3 PERIODIC CONVOLUTIONAL INTERLEAVING

Low Earth-orbiting spacecraft sending telemetry to the ground using the services of the TDRSS S-band Single Access (SSA) channel when symbol rates exceed 300 ks/s may experience pulsed radio interference which is expected to severely degrade the link performance during certain portions of the user orbit. In order to be able to maintain specified performance on this link at all times, the user satellite must employ an interleaving technique in conjunction with the convolutional coding and must increase the effective isotropic radiated power (EIRP). These techniques will ensure that no more than one of the dependent symbol errors due to a single radio frequency interference (RFI) pulse is within the path memory length of the decoder at any given time, and that the signal energy has been increased sufficiently to offset the increased symbol error probability (Reference [18]).

The interleaving parameters have been selected to achieve this goal for a particular worst-case pulse interference signature and the maximum symbol rate (6 Ms/s) of the SSA channel. De-interleaving must take place before convolutional decoding, and therefore is accomplished at the White Sands Ground Terminal.

3.2.4 REED-SOLOMON CODE

Due to the nature of Viterbi decoding, the decoded bit errors of the (7, 1/2) convolutional code tend to clump together in bursts. For this reason, in a concatenated coding system that uses a convolutional inner code, the outer code should be tailored to a burst error environment.

The code that is recommended as the outer code is a (255, 223) Reed-Solomon code. This code is a non-binary code. Each member of its coding alphabet is one of 256 elements of a finite field rather than zero or one. A string of eight bits is used to represent elements in the field so that the output of the encoder still looks like binary data.

Reed-Solomon codes are block codes. This means that a fixed block of input data is processed into a fixed block of output data. In the case of the (255, 223) code, 223 Reed-Solomon input symbols (each eight bits long) are encoded into 255 output symbols. The Reed-Solomon code in the Recommendation is systematic. This means that some portion of the codeword contains the input data in unalterable form. In this case, the first 223 symbols are the input data. The Reed-Solomon decoder almost always knows when there are too many errors to correct a word. In the event this happens, the decoder can inform the user of this fact.

A Reed-Solomon symbol size of eight bits was chosen because the decoders for larger symbol sizes would be difficult to implement with current technology. This choice forces the longest codeword length to be 255 symbols. A 16 Reed-Solomon symbol error correction capability was chosen as this was shown to have the best performance when concatenated with the (7, 1/2) convolutional inner code (References [10], [14] and [16]). Since two check symbols are required for each error to be corrected, this results in a total of 32 check symbols and 223 information symbols per codeword.

The (255, 223) Reed-Solomon code is capable of correcting up to 16 Reed-Solomon symbol errors in each codeword. Since each symbol is actually eight bits, this means that the code can correct up to 16 short bursts of error due to the inner convolutional decoder.

In addition, the Reed-Solomon codewords can be interleaved on a symbol basis before being convolutionally encoded. Since this separates the symbols in a codeword, it becomes less likely that a burst from the Viterbi decoder disturbs more than one Reed-Solomon symbol in any one codeword. This improves the performance of the Reed-Solomon code. An interleaving depth of five was chosen for two reasons (Reference [14]). A depth of five results in performance that is virtually indistinguishable from a depth of infinity. Also, a depth of five results in a frame length (a set of five codewords which, together with the check symbol field, constitutes a codeblock) that is a good compromise considering ease of handling, data outages (quality, quantity and continuity) and frame synchronization rate.

The same encoding and decoding hardware can implement a shortened (n, n-32) Reed-Solomon code, where n = 33, 34, ... , 254. This is accomplished by assuming that the remaining symbols are fixed: in the case of the Recommendataion, they are assumed to be all zero. This virtual zero fill allows the frame length to be tailored, if necessary, to suit a particular mission or situation.

The method currently recommended for synchronizing the codeblock is by synchronization of the Transfer Frame which contains a frame synchronization marker of 32 bits. However, advanced approaches being studied (e.g., self-synchronizing Reed-Solomon codes) may enable these two functions to be separately synchronized in the future.

The Reed-Solomon code, like the convolutional code, is a transparent code. This means that if the channel symbols have been inverted somewhere along the line, the decoders will still operate. The result will be the complement of the original data. However, the Reed-Solomon code loses its transparency if virtual zero fill is used. For this reason it is mandatory that the sense of the data (i.e., true or complemented) be resolved before Reed-Solomon decoding.

The two polynomials that define the Reed-Solomon code (Section 4.2(4) and (5) in Reference [2], and Reference [17]) were chosen to minimize the encoder hardware. The code generator polynomial is a palindrome (self-reciprocal polynomial) so that only half as many multipliers are required in the encoder circuit. The particular primitive element "α" (and hence the field generator polynomial) was chosen to make these multipliers as simple as possible. An encoder using the "dual basis" representation requires for implementation only a small number of integrated circuits or a single VLSI chip.

The False Sync Problem

Issue 1 of the Telemetry Channel Coding Blue Book (May 1984) made reference to a "False Sync" problem in footnote 5. As defined by the Recommendation at that time, the codeblock "attached sync marker" was included as a part of the Reed-Solomon data space. It was discovered that under certain repeating data values (e.g., test patterns of "01010101...") the R-S encoding algorithm regenerates the pattern of the leading data bytes in the leading bytes of the check symbol field. If the leading bytes happen to be the codeblock sync marker, two sync markers will appear in each R-S codeblock, leading to confusion in determining which is the correct starting point for the codeblock. The Recommendation indicated that a solution would be sought.

Various solutions were studied and it was finally decided to adopt the cleanest technical solution: to remove the attached sync marker from the encoding process. In addition, by steering the 32-bit sync marker away from the R-S encoder, the R-S codeblock now has space for an additional 32 bits of data. This solution was incorporated into Issue 2, References [1] and [2], which redefined the "Codeblock" (and "Transfer Frame", for consistency) to exclude their respective "attached sync markers". Of course, an attached sync marker must still precede each uncoded Transfer Frame, or each R-S codeblock.

ANNEX A

GLOSSARY OF TELEMETRY TERMINOLOGY

Block Encoding:

A one-to-one transformation of sequences of length k of elements of a source alphabet to sequences of length n of elements of a code alphabet, n>k.

Channel Symbol:

The unit of output of the innermost encoder which is a serial representation of bits, or binary digits, which have been encoded to protect against transmission induced errors.

Clean Data (Bits):

Data (bits) which are error free within the error detection and optional error correction capabilities of the TM System.

Codeblock:

A codeblock of an (n,k) block code is a sequence of n channel symbols which were produced as a unit by encoding a sequence of k information symbols, and will be decoded as a unit.

Code Rate:

The average ratio of the number of binary digits at the input of an encoder to the number binary digits at its output.

Codeword:

In a block code, one of the sequences in the range of the one-to-one transformation (see **Block Encoding**).

Command Link Control Word:

The Telecommand System Transfer Layer protocol data unit for Telecommand reporting via the TM Transfer Frame Operational Control Field.

Concatenation:

The use of two or more codes to process data sequentially with the output of one encoder used as the input of the next.

Constraint Length:

In convolutional coding, the number of consecutive input bits that are needed to determine the value of the output symbols at any time.

Convolutional Code:

As used in this document, a code in which a number of output symbols are produced for each input information bit. Each output symbol is a linear combination of the current input bit as well as some or all of the previous k-1 bits, where k is the constraint length of the code.

Fill Bit(s):

Additional bit(s) appended to enable a "data entity" to exactly fit an integer number of octets or symbols.

Inner Code:

In a concatenated coding system, the last encoding algorithm that is applied to the data stream. The data stream here consists of the codewords generated by the outer decoder.

Modulating Waveform:

A way of representing data bits ("1" and "0") by a particular waveform.

NRZ-L:

A modulating waveform in which a data "one" is represented by one of two levels, and a data "zero" is represented by the other level.

NRZ-M:

A modulating waveform in which a data "one" is represented by a change in level and a data "zero" is represented by no change in level.

Octet:

An 8-bit word consisting of eight contiguous bits.

Outer Code:

In a concatenated coding system, the first encoding algorithm that is applied to the data stream.

Packet:

An efficient application-oriented protocol data unit that facilitates the transfer of source data to users located in space or on Earth.

Protocol:

A set of procedures and their enabling format conventions that define the orderly exchange of information between entities within a given layer of the TM System.

Reed-Solomon ("R-S") Symbol:

A set of J bits that represents an element in the Galois field GF(2^J), the code alphabet of a J-bit Reed-Solomon code.

Reliable:

Meets the quality, quantity, continuity and completeness criteria which are specified by the TM System.

Segment:

A protocol data unit which facilitates telemetry flow control through the breaking of long source packets into communications-oriented data structures.

Systematic Code:

A code in which the input information sequence appears in unaltered form as part of the output codeword.

Telemetry System:

The end-to-end system of layered data handling services which exist to enable a spacecraft to send measurement information, in an error-controlled environment, to receiving elements (application processes) in space or on Earth.

Transfer Frame:

A communication oriented protocol data unit that facilitates the transfer of application oriented protocol data units through the space-to-ground link.

Transparent:

The invisible and seemingly direct (virtual) transfer of measurement information from the spacecraft source application process to the user (receiving application process).

Transparent Code:

A code that has the property that complementing the input of the encoder or decoder results in complementing the output.

User:

A human or machine-intelligent process which directs and analyzes the progress of a space mission.

Virtual Channel:

A given sequence of Transfer Frames, which are assigned a common identification code (in the Transfer Frame header), enabling all Transfer Frames who are members of that sequence to be uniquely identified. It allows a technique for multiple source application processes to share the finite capacity of the physical link (i.e., through multiplexing).

Virtual Fill:

In a systematic block code, a codeword can be divided into an information part and a parity (check) part. Suppose that the information part is N symbols long (symbol is defined here to be an element of the code's alphabet) and that the parity part is M symbols long. A "shortened" code is created by taking only S (S < N) information symbols as input, appending a fixed string of length N-S and then encoding in the normal way. This fixed string is called "fill". Since the fill is a predetermined sequence of symbols, it need not be transmitted over the channel. Instead, the decoder appends the same fill sequence before decoding. In this case, the fill is called "Virtual Fill".

ANNEX B

"APPLICATION NOTES" FOR PACKET TELEMETRY

Purpose:

The CCSDS Telemetry System architecture discussed in this report is layered so that various levels of interface compatibility are possible by the judicious selection of available options. This Annex describes how some of these options may be selected.

Status:

This Annex is currently under development by the CCSDS.

B-1 HANDLING PLAYBACK DATA IN REVERSE DIRECTION

Under some situations it may be desired to play back stored data in the reverse direction. The "Packet Order Flag" (Reference [1], Section 5.2.4 (c)) signals this condition. There are three recognized options for implementing reverse playback on the spacecraft:

(1) The complete telemetry stream may be recorded as a series of telemetry frames. This entire stream may then later be replayed in reverse direction and dumped to the receiving element OVER A PHYSICAL DATA CHANNEL WHICH IS SEPARATE FROM THAT USED TO TRANSMIT REAL-TIME DATA. In this case, the Packet Order Flag shall indicate the status of the packets or segments WHEN THE FRAMES WERE ORIGINALLY RECORDED.

(2) The complete telemetry stream may be recorded as a series of telemetry frames, each having their Packet Order Flag set as appropriate during recording. This entire stream may then later be replayed in reverse direction as a pure bit-stream for insertion within the data field of new frames which form a separate playback Virtual Channel. These playback frames may then be interleaved with other frames which form Virtual Channels that contain real-time packets or segments. In this case, the replayed bit-stream will be inserted into the playback Virtual Channel asynchronously, with the "Data Field Synchronization Flag" for this channel set to a "1" and the Packet Order Flag consequently ignored. (Note: precautions must be taken to ensure that the replayed synchronization marker occurring periodically within the frame data field does not interfere with the overall frame synchronization strategy. As an example, the reverse-justified synchronization marker should be distinguishable from the forward-justified pattern.)

(3) Packets or segments may be recorded with or without first encapsulating them within Transfer Frames. These packets or segments may later be replayed in reverse direction, and re-synchronized on-board the spacecraft for normal insertion into the Data Field of new real-time transfer frames.

B-2 REAL TIME DATA INSERT

The Real Time Data Insert is described in Reference [1], Section 5.3.2. The format, utilization and operational procedures associated with the Real Time Data Insert field are at this time all mission-dependent and shall be the subject of detailed cross-support agreements between the agencies involved.

B-3 TAILORING TELEMETRY TRANSFER FRAME LENGTHS FOR 32-BIT PROCESSORS

The CCSDS Recommendation for Packet Telemetry is organized around the use of "octets" (8-bit bytes) for both Source Packets and Transfer Frames, and 8-bit symbols for the corresponding Reed-Solomon code. Thus an attempt has been made to maintain all fields including the total length to be a multiple of 8 bits. However, some users may find it advantageous to organize frame lengths in multiples of 32 bits for more efficient manipulation during very high speed operations (e.g., frame synchronization) using 32-bit based microprocessors. The Recommendations are designed to permit such organization under the following conditions:

If Reed-Solomon coding is NOT used, then the preferred transfer frame length is 8896 bits, because this is the longest frame (of length 8920 bits or less) which is evenly divisible by 32. When the sync marker is attached to the Transfer Frame, the frame synchronizer on the ground will see 8896 + 32 or 8928 bits, which still maintains divisibility by 32. It should be noted that other lengths, such as the 8800-bit length described below, can also be chosen for the non-RS-encoded case.

If Reed-Solomon Coding IS used, then an additional coding constraint must be satisfied: the codeblock must, in addition, be integrally divisible by 8I, where I is the interleaving depth used. Using the preferred interleaving depth of I=5, this means any shortening of the transmitted codeblock must be achieved by adding virtual fill in multiples of 40 bits. (A transmitted codeblock consists of the Transfer Frame plus the appended Reed-Solomon check symbols.) The largest Transfer Frame size (of 8920 bits or less) that meets BOTH criteria (i.e., a multiple of both 32 and 40 bits) is 8800 bits. To this value we add the fixed 1280 bits consisting of the R-S check symbols to yield a transmitted codeblock length of 10080 bits. It can be seen that such a length is divisible by 32 (for processing efficiency) as well as 40 (for the interleaving process). With this length, each codeblock must be configured for 120 bits of virtual fill to make the logical codeblock always equal to 10200 bits. The sync marker is then attached to the transmitted codeblock, and the total length seen by the ground frame synchronizer is 10080 + 32 = 10112 bits, which is also divisible by 32. Thus the parameters selected would be:

Transfer Frame		8800 bits
Virtual Fill	(for I=5)	120 bits
Transmitted Codeblock	(8800 + 1280)	10080 bits
Logical Codeblock		10200 bits
Ground frame sync set to	(10080 + 32)	10112 bits

The requirements and principles described above can also be applied when optimizing for other than 32-bit processors.

ANNEX C

SUMMARY OF SEGMENTATION OPTIONS

Purpose:

This Annex provides a summary of the various options which exist for segmenting very long TM Source Packets in order to achieve flow control through the space data channel.

C-1 SEGMENTATION SUMMARY

Several options for segmenting long source packets are specified in Sections 4 and 5 of Reference [1]. In selecting the segmentation method to be used for a particular mission, the following system considerations may be important:

(1) Segmentation should not introduce extra overhead into short packets which have no need to be segmented.

(2) It should be possible to mix short unsegmented packets on the same virtual channel with long source packets which have been divided into segments.

(3) It is highly desirable to implement a solution which uses a single protocol for both segmented and unsegmented packets, in which data fields are interpreted in singular, consistent ways.

(4) For a given mission, a fixed maximum segment length should be selected. When long packets are broken into segments, the segment lengths may be equal to the mission-fixed maximum, except for the last segment which may contain the residue of the original packet.

(5) The segmentation solution should involve the simplest possible algorithms for extracting the packets and segments from the Transfer Frame, and for reconstituting the packets, since these algorithms must operate at full incoming telemetry bit rate.

Table C-1 presents a summary of the major attributes of the various alternative methods on segmentation.

Table C-1: Summary of Segmentation Options

Option in Ref. [1]	Principle	Segments Formed By	Ground Processor Uses	User Receives	Total Overhead
Source - internal using Version 1 Source Packet (4.1)	Use source sequence counter to identify each segment within the packet	Source	Packet sequence count and length fields to extract segments	Segments	6 octets/ segment + 4 octets (length & count) in the first segment
Spacecraft segmentat-ion using Version 1 Source Packet (4.2)	Nest the user source packet within an "outer" spacecraft source packet	Spacecraft data system	Outer spacecraft source packet to extract and recombine segments	Source packet	6 octets/ segment + 6 octets (original header) in the first segment
Spacecraft segmentat-ion using Version 2 Telemetry Segment (4.3)	Source packet length field decremented by fixed binary segment length	Spacecraft data system	Known fixed length to extract each segment; inferred sequence to recombine them	Source packet	6 octets/ segment
Virtual Channel-ization (5.2.2 b)	Long packets assigned to their own dedicated Transfer Frame	Spacecraft data system	Virtual Channel ID in Transfer Frame header	Source packet	Depends on design: may be zero

ANNEX D

TELEMETRY TRANSFER FRAME

ERROR DETECTION

ENCODING/DECODING GUIDELINE

Purpose:

This Annex provides a description of the error detection encoding and decoding procedures recommended for use in conjunction with the Frame Error Control field of the Telemetry Transfer Frame.

D-1 CODING FOR ERROR DETECTION IN TRANSFER FRAMES

This Annex describes the error detection encoding/decoding procedure that is recommended for Transfer Frame coding.

The code specifies the same generator polynomial used by HDLC (ISO), ADCCP (ANSI), V.41 (CCITT), etc. It has the following capabilities when applied to an encoded block of less than 32,768 (2^{15}) bits:

(1) All error sequences composed of an odd number of bit errors are detected.

(2) All error sequences containing at most two bit errors anywhere in the encoded block will be detected.

(3) If a random error sequence containing an even number of bit errors (greater than or equal to 4) occurs within the block, the probability that the error will be undetected is approximately 2^{-15} (or approximately 3 x 10^{-5}).

(4) All single error bursts spanning 16 bits or less will be detected provided no other errors occur within the block.

D-1.1 Encoding Procedure

The encoding procedure accepts an (n-16)-bit data block and generates a systematic binary (n,n-16) block code by appending a 16-bit Frame Check Sequence (FCS) as the final 16 bits of the codeblock. This FCS is inserted into the Frame Error Control Word of the Transfer Frame Trailer. The equation for the FCS is:

$$\text{FCS} = [X^{16} \cdot M(X) \oplus X^{(n-16)} \cdot L(X)] \text{ modulo } G(X)$$

where

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:

$$L(X) = \sum_{i=0}^{15} x^i \quad \text{(all "1" polynomial of order 15)}$$

G(X) is the generating polynomial given by:

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

n is the number of bits in the encoded message

$\oplus$ is the modulo 2 addition operator (Exclusive OR)

Note that the encoding procedure differs from that of a conventional cyclic block encoding operation in that:

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

D-1.2 Decoding Procedure

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

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

where C*(X) is the received block in polynomial form and S(X) is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected.

D-2 POSSIBLE IMPLEMENTATION

A possible implementation of the above-defined encoding/decoding procedure is described below.

D-2.1 Encoding

Figure D-1 shows an arrangement for encoding using the shift register. To encode, the storage stages are set to "one", gates A and B are enabled (closed), gate C is inhibited (open), and (n-16) message bits are clocked into the input. They will appear simultaneously at the output. After the bits have been entered, the output of gate A is clamped to "zero", gate B is inhibited, gate C is enabled, and the register is clocked a further 16 counts. During these counts the required check bits will appear in succession at the output.

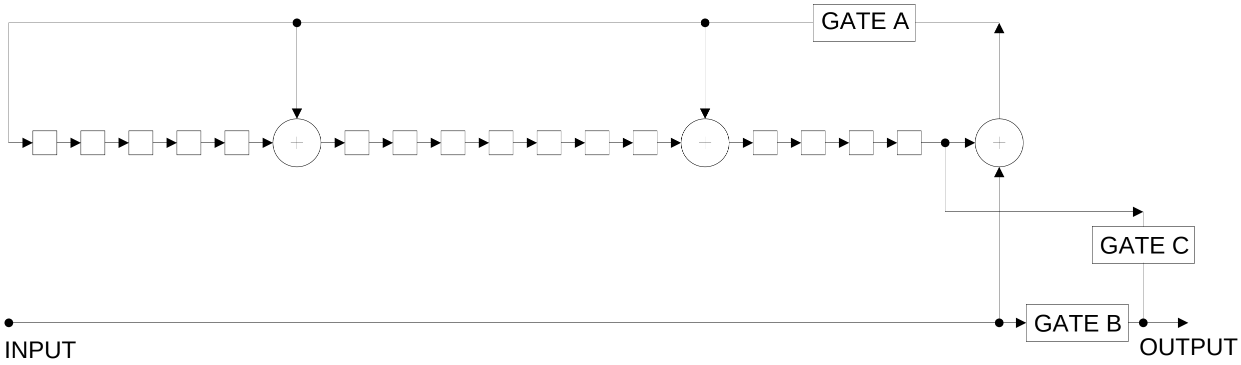


Figure D-1: Encoder

D-2.2 Decoding

Figure D-2 shows an arrangement for decoding using the shift register. To decode, the storage stages are set to "one" and gate B is enabled. The received n-bits [the (n-16) message bits plus the 16 bits of the FCS] are then clocked into the input. After n-16 counts, gate B is inhibited, the 16 check bits are then clocked into the input, and the contents of the storage stages are then examined. For an error-free block, the contents will be zero. A non-zero content indicates an erroneous block.

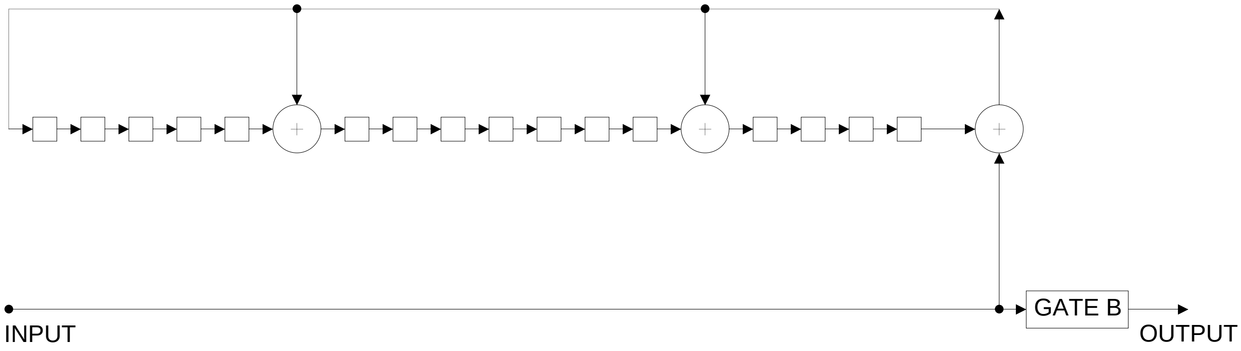


Figure D-2: Decoder