ECSS-E-HB-50A
15 November 2008

EUROPEAN COOPERATION
ECSS
FOR SPACE STANDARDIZATION

Space engineering

Communications guidelines

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

This Handbook has been prepared by the ECSS-E-HB-50 Working Group, reviewed by the ECSS Executive Secretariat and approved by the ECSS Technical Authority.

Disclaimer

ECSS does not provide any warranty whatsoever, whether expressed, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item are error-free. In no respect shall ECSS incur any liability for any damages, including, but not limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or in any way connected to the use of this Handbook, whether or not based upon warranty, business agreement, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by ECSS.

Published by: ESA Requirements and Standards Division
ESTEC, P.O. Box 299,
2200 AG Noordwijk
The Netherlands

Change log

ECSS-E-HB-50A 15 November 2008	First issue

Table of contents

Figures

Tables

1 Introduction

1.1 Purpose

This ECSS handbook is intended to help implementers and users of data handling systems who are adhering to the ECSS E-50 series of standards. The handbook provides an overview of the E-50 standards and related CCSDS Recommended Standards and describes how the individual standards may be used together to form a coherent set of communications protocols. It also evaluates issues which could not be discussed in the Standards documents themselves, and provides guidance on option selection and implementation choices.

1.2 Scope

This handbook provides guidance to the ECSS E-50 series of standards including related CCSDS Recommendations. The information provided is informative and intended to be used as best practice; it is not binding on implementers.

The information contained in this handbook is not part of the ECSS Standards. In the event of any conflict between the ECSS Standards and the material presented in this handbook, the ECSS Standards prevail.

1.3 Document structure

This document is divided into sections and annexes as follows:

- Section 1 (this section) provides intentional and administrative information.
- Section 2 provides the definition and abbreviations of the terms used in the present document.
- Section 3 gives a list of the E-50 series of standards and describes their relationship to CCSDS and other standards bodies. It also provides an overall architectural framework.
- Section 4 provides detailed information on each of the individual ECSS and CCSDS standards covered by the handbook.
- Section 5 addresses individual technical topics related to the ECSS E-50 standards.

- Section 6 provides guidance on selecting appropriate ECSS and CCSDS standards for coherent mission and infrastructure scenarios.
- Section 7 provides a summary of supporting components and products.
- Annex A to Annex F contain draft proforma of the Protocol Implementation Conformance Statement (PICS) for three of the E-50 standards.
- Annex G provides additional performance data for one of the telemetry channel coding options.

2
Terms, definitions and abbreviations

2.1 Terms and definitions from other documents

For the purpose of this document, the terms and definitions from ECSS-S-ST-00-01 and ECSS-E-ST-50 apply.

2.2 Terms specific to the present document

3.2.1 delimited

<data unit> with a known and finite length.

3.2.2 mission phase

period of a mission during which specified communications characteristics are fixed

NOTE The transition between two consecutive mission phases can cause an interruption of the communications services.

3.2.3 octet

group of eight bits

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

3.2.4 physical channel

stream of bits transferred over a space link in a single direction

2.3 Conventions

2.3.1 Bit numbering and most significant bit

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 an N-bit field is used to express a binary value (such as a counter), the most significant bit (MSB) is the first bit of the field, i.e. bit 0 (see Figure 2-1).

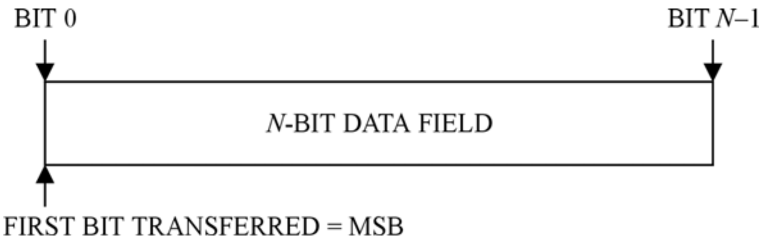

Figure 2-1: Bit numbering convention

2.4 Abbreviated terms

Abbreviation	Meaning
8PSK	phase shift keying of 8 states
AD	acceptance-check and data
AOS	advanced orbiting systems
API	application programming interface
APID	application process identifier
APP	*a posteriori* probability
ARP	address resolution protocol
ARQ	automatic repeat request
ASIC	application-specific integrated circuit
ASM	analogue signal monitor
ASM	attached sync marker
AWGN	additive white Gaussian noise
BC	bypass (of acceptance check) and control
BCH	Bose-Chaudhuri-Hocquenghem
BD	bypass (of acceptance check) and data
BDM	bi-level discrete monitor
BER	bit error rate
BPSK	binary phase shift keying
BSD	bi-directional serial digital
BSM	bi-level switch monitor
CADU	channel access data unit
CAN	controller area network
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS file delivery protocol
CLCW	communications link control word
CLTU	communications link transmission unit
COP	communications operation procedure

Abbreviation	Meaning
CPDU	command pulse distribution unit
CRC	cyclic redundancy code
dB	decibel
DHCP	dynamic host configuration protocol
DHS	data handling system
DML	data management library
DNS	domain name service
DRD	document requirements definition
DTN	delay tolerant networking
DWT	discrete wavelet transform
EDAC	error detection and correction
EMC	electromagnetic compatibility
EOF	end of file
ESA	European Space Agency
ESOC	European Space Operations Centre
FARM	frame acceptance and reporting mechanism
FDU	frame data unit
FER	frame error rate
FOP	frame operation procedure
FPGA	field-programmable gate array
FQPSK	Feher quadrature phase shift keying
GMSK	Gaussian minimum shift keying
GSTVi	ground systems test and validation infrastructure
HPC	high power command
HSRP	hot standby redundancy protocol
ID	identifier
IDC	image data compression
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISD	input serial digital
ISO	International Organization for Standardization
ISS	international space station
ITU	International Telecommunication Union
JAXA	Japan Aerospace Exploration Agency

Abbreviation	Meaning
JPEG	Joint Photographic Experts Group
LOS	loss of signal, or, line of sight
LPC	low power command
LVDS	low voltage data signalling
MAP	multiplexer access point
MER	Mars exploration rover
MCID	master channel identifier
MIL	military standard
MPE	multiprotocol encapsulation
MSB	most significant bit
NASA	National Aeronautics and Space Administration
OCF	operational control field
OID	only idle data
OQPSK	offset quadrature phase shift keying
OSD	output serial digital
OSI	open systems interconnection
OSPF	open shortest path first
PAC	packet assembly controller
PCM	pulse code modulation
PDU	protocol data unit
PEP	performance enhancing proxy
PICS	protocol implementation conformance statement
PLOP	physical layer operation procedure
PRL	PICS requirements list
PSNR	peak signal-to-noise ratio
PSS	procedures, specifications and standards
PUS	packet utilization standard
PVN	packet version number
QPSK	quadrature phase shift keying
RAF	return all frames
RCF	return channel frames
RIP	routing information protocol
RMAP	remote memory access protocol
RF	radio frequency
RFC	request for comments
R-S	Reed-Solomon

Abbreviation	Meaning
RTEMS	real-time executive for multiprocessor systems
SAR	synthetic aperture radar
SCID	spacecraft identifier *or* spacecraft identification
SCOS	spacecraft operating system
SCPS	space communications protocol specification
SEC	single error correction
SFO	store-and-forward overlay
SLE	space link extension
SOIS	spacecraft onboard interface services
SRD	system requirements document
SRRC	square root raised cosine
TC	telecommand
TCM	trellis-coded modulation
TCP	transmission control protocol
TDM	time division multiplexing
TED	triple error detection
TFVN	transfer frame version number
TM	telemetry
TP	transport protocol
TRL	technology readiness level
TSM	temperature sensors monitor
TT&C	telemetry, tracking and command
UDP	user datagram protocol
VCID	virtual channel identifier
VRRP	virtual router redundancy protocol

3
The E-50 series of standards

3.1 Background

3.1.1 Scope of the ECSS E-50 series

The E-50 series of standards are all related to the end-to-end communications between flight and ground infrastructures. The communications links covered by the E-50 series are the space-to-ground and space-to-space links used during spacecraft operations, and the communications links to the spacecraft used during the assembly, integration and test, and operational phases.

Spacecraft end-to-end communication systems comprise components in three distinct domains: the ground network, the space link and the space network. The E-50 series covers the components of the space link and space network in detail. However, they only cover those aspects of the ground network that are necessary for the provision of the end-to-end communication services. Other aspects of the ground network are covered by the ECSS E-70 series of standards.

Although some of the E-50 standards have been specifically developed under the ECSS authority, many are derived from other standardisation bodies such as the Consultative Committee for Space Data Systems (CCSDS) and the SpaceWire working group.

3.1.2 CCSDS standards

CCSDS was formed in 1982 by the major space agencies of the world to provide a forum for discussion of common problems in the development and operation of space data systems. It is currently composed of ten member agencies, more than 20 observer agencies, and over 100 industrial associates.

CCSDS produces Recommended Standards targeted at cross-support whereby the infrastructure owned by one agency may be utilised by another, for example for telemetry reception. Many CCSDS Recommended Standards are published as standards by ISO (International Organization for Standardization).

CCSDS Recommended Standards (Blue Books) define specific interfaces, technical capabilities or protocols, or provide prescriptive definitions of interfaces, protocols, or other controlling standards such as encoding approaches. The reference number for a Blue Book includes the letter B: for example, CCSDS 131.0-B-1. The last number within the reference number is the issue number.

The publication by CCSDS of a Recommended Standard is not considered binding on any CCSDS member agency. A CCSDS member can choose to endorse or adopt a Recommended Standard or to establish a related standard that is in accord with it. The CCSDS Recommended Standards are referred to in this handbook as "standards" but that should not be taken to imply any adoption by ECSS. Recommended Standards carry on their cover page the words "Recommendation for Space Data System Standards" and are frequently referred to as "CCSDS recommendations" or simply "recommendations".

Informational Reports (Green Books) form another important category of CCSDS publications. They provide a very broad range of general and background information. They are often published in support of a standard and include high-level overviews, performance data, implementation and testing experience and tutorials. They offer useful informal information, some of which has been used in the preparation of this handbook. The reference number for a Green Book includes the letter G: for example, CCSDS 130.1-G-1.

CCSDS also publishes documents in other categories:

- CCSDS Recommended Practices (Magenta Books) are the consensus results of CCSDS community deliberations and provide a way to capture "best" or "state of the art" approaches for applying or using standards.
- CCSDS Experimental Specifications (Orange Books) are published as part of a research or development effort based on prospective requirements, and as such they are not considered as standards documents. An Experimental Specification is intended to demonstrate technical feasibility in anticipation of a possible future requirement.
- The CCSDS Record (Yellow Books) category includes CCSDS administrative documents such as charters and procedures.
- The CCSDS Historical (Silver Books) designation is applied to approved CCSDS documents that have been superseded by more recent versions or are otherwise considered obsolete. Most of the documents in this category were originally Blue Books but it also includes documents from other categories. When a document is given Silver status, its reference number is extended by appending "-S" at the end.

3.1.3 Relationship to PSS standards

In the past, Recommended Standards developed by CCSDS have generally not been directly adopted by ESA. Rather, minor modifications have been introduced to suit the specific needs of European missions and the modified Recommended Standard has been released under a separate cover. Previously these modified recommendations were released in the ESA Procedures, Specifications and Standards (PSS) series of documents.

The PSS documents related to space communication were typically derived from CCSDS recommendations and included or excluded specific options depending on European preferences.

All PSS documents have now been withdrawn in favour of ECSS standards or CCSDS Recommended Standards.

3.1.4 Relationship of ECSS standards to CCSDS standards

The policy of modifying CCSDS Recommended Standards to suit the specific needs of European missions and publishing them as ECSS standards has continued within ECSS for the core telemetry and telecommand standards. The ECSS standards in this area are generally aligned with the equivalent CCSDS approach with two minor differences:

- The ECSS series of standards includes the *Telemetry and telecommand packet utilization standard*, ECSS-E-ST-70-41, as an application layer protocol. This standard has been developed independently by ESA and ECSS and has not been adopted by CCSDS.
- The ECSS series combines the telecommand protocols into a single standards document, ECSS-E-ST-50-04, whereas CCSDS uses three separate documents. This difference is for historical reasons and the protocols are intended to be equivalent. Some minor differences are described in the section 4.7, about ECSS-E-ST-50-04.

This practice of modifying CCSDS Recommended Standards is now being phased out and in the future they will be directly reused by ECSS thorough adoption notices. The adoption notices may include minor modifications such as the removal of options. It should be noted that not all CCSDS Recommended Standards have been adopted by ECSS.

3.2 Standards covered in this handbook

Table 3-1 lists the ECSS E-50 standards covered by this handbook. Section 4 describes each E-50 standard in some detail and discusses optional features and recommended practices. Where applicable, the differences from equivalent CCSDS Recommended Standards are identified and discussed.

Note that Table 3-1 lists the standards in the numerical order of their reference numbers. This does not reflect the order used within Section 4, so the table shows the appropriate section.

Table 3-2 lists the other standards discussed in Section 4. Most of these are CCSDS Recommended Standards, which can be combined with the ECSS E-50 standards to cover end-to-end communications across the space link and space network.

Table 3-2 also includes ECSS-E-ST-70-41, the telemetry and telecommand packet utilization standard (PUS), which belongs to the ECSS E-70 ground systems and operations series of standards. It is included in Section 4 because it defines the use of fields in CCSDS Space Packets. Other features of the PUS, such as its services, are not discussed in any detail.

Other standards for the ground network are not covered in this handbook, though they are occasionally mentioned for compatibility reasons. For example, these include the Space Link Extension (SLE) services. For a useful introduction to the SLE services, see the informational report CCSDS 910.0-G.

Figure 3-1 shows the standards from Table 3-1 and Table 3-2 and their relationship to the layers of the OSI reference model: these relationships are discussed in the next section. Within this handbook, the standards from Table 3-1 and Table 3-2 are referred to collectively as "E-50 standards", although this name strictly applies only those in Table 3-1. However, they all come within the scope described in 3.1.1 and can be used together to achieve end-to-end communications between flight and ground infrastructures.

Table 3-1: ECSS E-ST-50 standards covered in this handbook

Reference	Title	Issue / date		Section
E-ST-50	Space engineering - Communications	C	31 July 2008	4.2
E-ST-50-01	Space engineering - Space data links - Telemetry synchronization and channel coding	C	31 July 2008	4.5
E-ST-50-02	Space engineering - Ranging and Doppler tracking	C	31 July 2008	4.3
E-ST-50-03	Space engineering - Space data links - Telemetry transfer frame protocol	C	31 July 2008	4.6
E-ST-50-04	Space engineering - Space data links - Telecommand protocols, synchronization and channel coding	C	31 July 2008	4.7
E-ST-50-05	Space engineering - Radio frequency and modulation	C	31 July 2008	4.4
E-ST-50-11	Space engineering - SpaceWire protocols		To be published	4.18
E-ST-50-12	Space engineering - SpaceWire - Links, nodes, routers and networks	C	31 July 2008	4.18
E-ST-50-13	Space engineering - MIL 1553 Standard extension	C	15 Nov 2008	4.19
E-ST-50-14	Space engineering - Spacecraft discrete interfaces	C	31 July 2008	4.20

Table 3-2: Other standards covered in this handbook

Reference	Title	Issue / date		Section
CCSDS 121.0-B	Lossless Data Compression	1	May 1997	4.16.2
CCSDS 122.0-B	Image Data Compression	1	Nov 2005	4.16.3
CCSDS 133.0-B	Space Packet Protocol	1	Sep 2003	4.10.2
CCSDS 133.1-B	Encapsulation Service	1	Jun 2006	4.10.3
CCSDS 135.0-B	Space Link Identifiers	3	Oct 2006	4.14
CCSDS 211.0-B	Proximity-1 Space Link Protocol - Data Link Layer	4	Jul 2006	4.9
CCSDS 211.1-B	Proximity-1 Space Link Protocol - Physical Layer	3	Mar 2006	4.9
CCSDS 211.2-B	Proximity-1 Space Link Protocol - Coding and Synchronization Sublayer	1	Apr 2003	4.9
CCSDS 301.0-B	Time Code Formats	3	Jan 2002	4.17
CCSDS 320.0-B	CCSDS Global Spacecraft Identification Field: Code Assignment Control Procedures	5	Sep 2007	4.15
CCSDS 727.0-B	CCSDS File Delivery Protocol (CFDP)	4	Jan 2007	4.13
CCSDS 732.0-B	AOS Space Data Link Protocol	2	Jul 2006	4.8
ECSS-E-ST-70-41	Ground systems and operations - Telemetry and telecommand packet utilization	A	30 Jan 2003	4.12
RFC 791	Internet Protocol, Protocol Specification *[1]*		Sep 1981	4.11.1
RFC 1883	Internet Protocol, Version 6 (IPv6) Specification		Dec 1995	4.11.3

[1] now known as IPv4

Section 3.3.2 discusses the Space Data Link Protocols in the Data Link Protocol Sublayer and the related Synchronization and Channel Coding Sublayer. Table 3-3 provides an overview of the associated ECSS and CCSDS standards documents.

Table 3-3: Standards for the TM, TC and AOS Space Data Link Protocols

<table>
<tr><th rowspan="2">Space Data Link Protocol</th><th colspan="2">Standards documents</th><th rowspan="2">Synchronization and Channel Coding</th><th colspan="2">Standards documents</th></tr>
<tr><th>ECSS</th><th>CCSDS</th><th>ECSS</th><th>CCSDS</th></tr>
<tr><td>TM</td><td>ECSS-E-ST-50-03</td><td>CCSDS 132.0-B</td><td rowspan="2">TM</td><td rowspan="2">ECSS-E-ST-50-01</td><td rowspan="2">CCSDS 131.0-B</td></tr>
<tr><td>AOS</td><td>-</td><td>CCSDS 732.0-B</td></tr>
<tr><td>TC</td><td>ECSS-E-ST-50-04</td><td>CCSDS 232.0-B</td><td>TC</td><td>ECSS-E-ST-50-04</td><td>CCSDS 231.0-B</td></tr>
</table>

3.3 ECSS E-50 Protocols

Figure 3-1 shows the ECSS E-50 standards as well as standards from CCSDS and IETF which address areas within the ECSS E-50 purview. It should be noted that not all standards may be applicable to a single project and some standards make reference to other standards.

A communications protocol is usually associated with one of the seven layers of the OSI (Open Systems Interconnection) reference model defined in ISO/IEC 7498-1. Although some space link protocols do not fit well with the OSI seven-layer model, ECSS E-50 uses this model for categorizing the space link protocols. The space link protocols are defined for the following five layers of the OSI model:

- Physical Layer
- Data Link Layer
- Network Layer
- Transport Layer
- Application Layer.

As in most terrestrial networks, protocols of the Session and Presentation Layers of the OSI model are rarely or never used over space links. Figure 3-1 shows the space link protocols categorized into the five layers listed above. Figure 3-2 shows some options for combining the protocols.

In Figure 3-1, there are two protocols that do not correspond to a single layer. The CCSDS File Delivery Protocol (CFDP) has the functionality of the Transport and Application Layers, and the Proximity-1 Space Link Protocol has the functionality of the Data Link and Physical Layers.

Although CCSDS and ECSS do not formally define application programming interfaces (APIs) for the space link protocols, most CCSDS standards provide service definitions in the form of primitives following the conventions established by ISO. A primitive is an abstract representation of an API that does not depend on any implementation technology and is suitable for use as the baseline for an API.

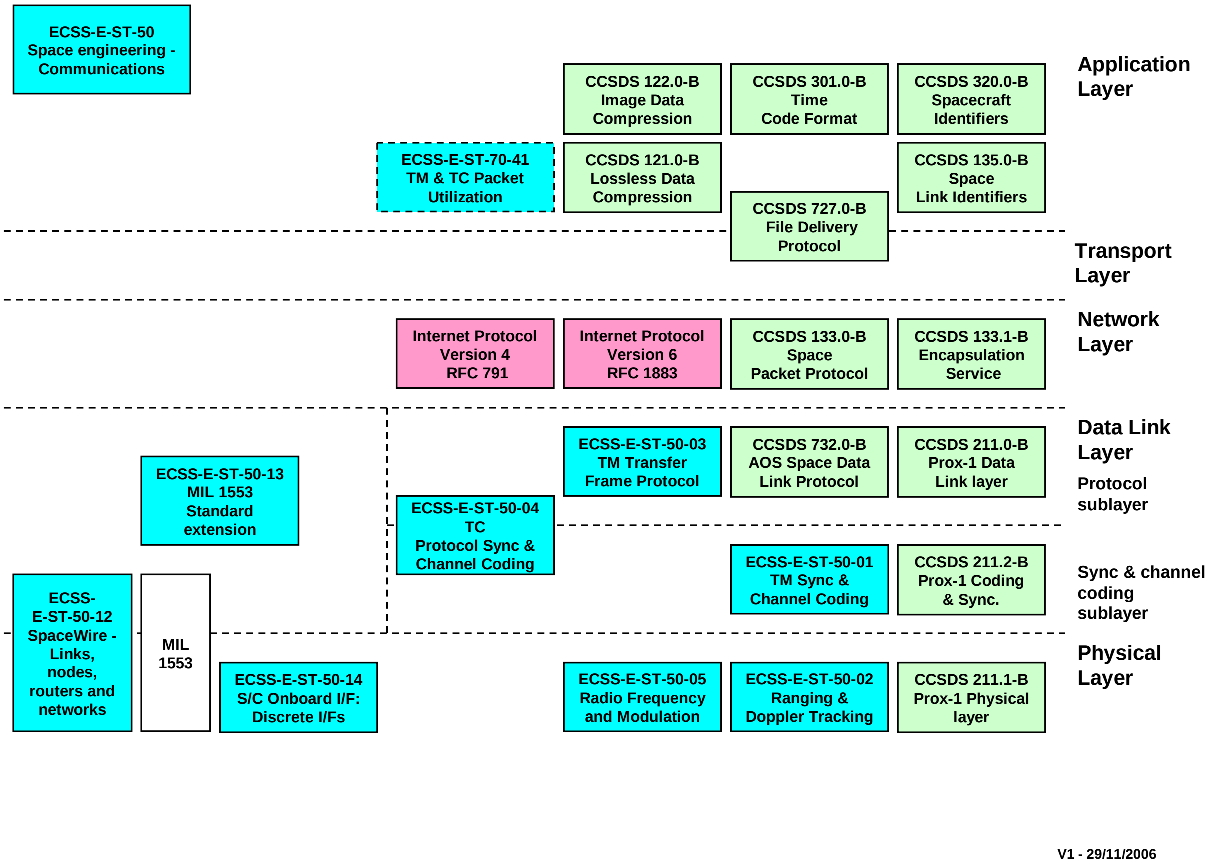

Figure 3-1: ECSS E-50 standards and associated layers

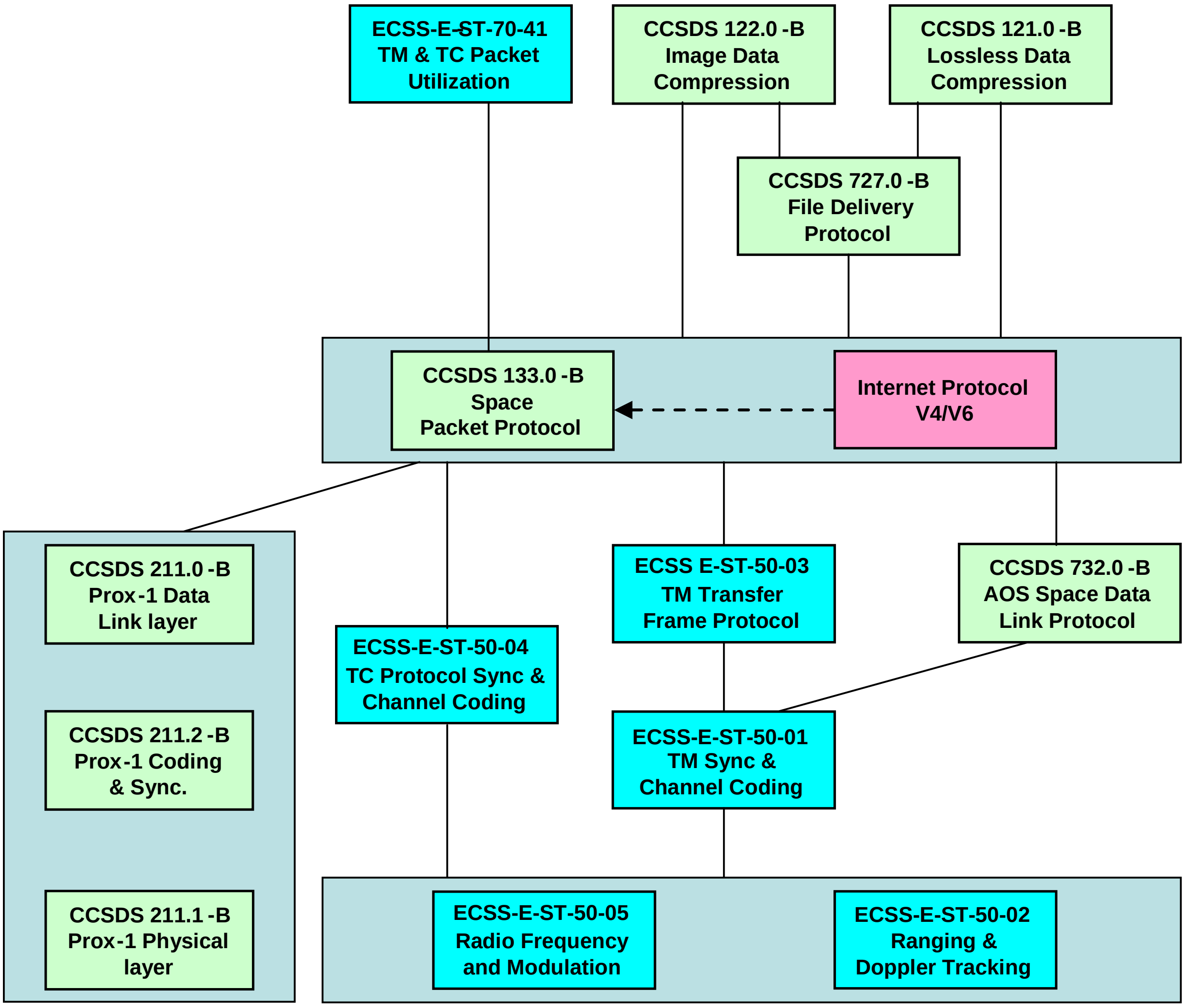

Figure 3-2: Some options for combining the protocols

3.3.1 Physical Layer

ECSS has the standard ECSS-E-ST-50-05, *Radio frequency and modulation*, for the Physical Layer of the space links between spacecraft and ground stations. In addition, CCSDS has a Recommended Standard, CCSDS 211.1-B, for the Physical Layer of the Proximity-1 Space Link Protocol.

3.3.2 Data Link Layer

The Data Link Layer has two sublayers: the Data Link Protocol Sublayer and the Synchronization and Channel Coding Sublayer.

The Data Link Protocol Sublayer specifies methods of transferring data units provided by the higher layer over a space link using data units known as transfer frames. There are three Space Data Link Protocols for the Data Link Protocol Sublayer:

- TM Space Data Link Protocol

- TC Space Data Link Protocol
- AOS Space Data Link Protocol

Each of these protocols provides the capability to send data over a single space link.

The Synchronization and Channel Coding Sublayer specifies methods of synchronization and channel coding for transferring Transfer Frames over a space link. There are two standards for the Synchronization and Channel Coding Sublayer of the Data Link Layer:

- TM Synchronization and Channel Coding
- TC Synchronization and Channel Coding

TM Synchronization and Channel Coding is used with the TM or AOS Space Data Link Protocol, and TC Synchronization and Channel Coding is used with the TC Space Data Link protocol.

In addition, there is the Proximity-1 Space Link Protocol for the Data Link Layer. It also provides a method for transferring data units over a space link using Transfer Frames, but has a more complex subdivision of the Data Link Layer into sublayers. Proximity-1 has its own standard for its Synchronization and Channel Coding Sublayer.

The TM, TC and AOS Space Data Link Protocols and the transfer frame protocol of the Proximity-1 Space Link protocol are called the Space Data Link Protocols in this document.

3.3.3 Network Layer

Space link protocols of the Network Layer provide the function of routing higher-layer data through the entire data system that includes both onboard and ground subnetworks. Three protocols of the Network Layer are discussed in this handbook:

- Space Packet Protocol
- Internet Protocol version 4 (IPv4)
- Internet Protocol version 6 (IPv6)

The packets of a Network Layer protocol are transferred over a space link with the Space Data Link Protocols of the Data Link Layer. The Space Packets of the Space Packet Protocol and the datagrams of IPv4 can be directly transferred as packets by the Space Data Link Protocols. The datagrams of IPv6 must first be encapsulated, for example into a Space Packet.

In some cases, Space Packets are generated and consumed by application processes, and in these cases this protocol is used both as a Network Layer protocol and as an Application Layer protocol.

3.3.4 Transport Layer

Although the transport layer is identified within the architecture, the CCSDS File Delivery Protocol (CFDP) is currently the only protocol defined by ECSS for use over the space link. Other protocols such as UDP may be applicable but generally the unidirectional nature of the TM/TC links coupled with inherent transmission delays preclude the use of conventional transport layer protocols such as TCP/IP.

Protocol Data Units (PDUs) of a Transport Layer protocol are usually transferred using a protocol of the Network Layer over a space link.

3.3.5 Application Layer

Space link protocols of the Application Layer provide users with end-to-end application services such as file transfer and data compression. Three protocols of the Application Layer are discussed in this handbook:

- Lossless Data Compression
- Image Data Compression
- Telemetry and telecommand packet utilization

The lossless and lossy data compression standards may be used to reduce the volume of data transmitted over the space link. The telemetry and telecommand packet utilization standard defines a set of Application Layer services for spacecraft monitoring and control.

The CCSDS File Delivery Protocol (CFDP) provides the functionality of the Application Layer (i.e. functions for file management), but it also provides functions of the Transport Layer.

Other Application Layer protocols are outside the scope of this handbook. A project can choose to use application-specific protocols to fulfil their mission requirements in the Application Layer over CCSDS space link protocols.

3.4 End to End Aspects

Having listed the various E-50 standards we now examine how these can be assembled as an end to end communication system between space and ground, and space to space. It is not possible to cover all feasible system configurations, either in the elements involved or the protocols utilised but the configurations presented here are generally applicable to any mission.

Figure 3-3 identifies the four elements typically involved in end-to-end communication:

- The onboard end system such as a payload. The onboard end system can be any onboard application, including command and control.
- The onboard relay system represented by the onboard data handling system which includes the TT&C elements and the onboard data buses (subnets).
- The ground relay system composed of the ground station with its telecommand transmission and telemetry acquisition elements, along with the networks used to communicate with the control centre.

- The ground end system represented by the control centre. The ground end system can also be a user application which receives, for example, science telemetry data.

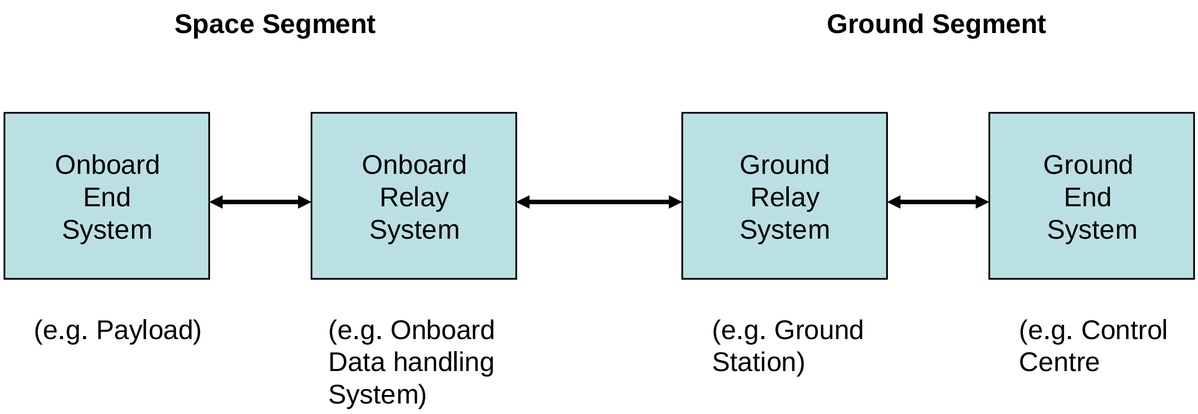

Figure 3-3: Simplified end-to-end communication system

Figure 3-4 takes the same end-to-end communication system and locates the various E-50 standards. The following should be noted:

- The use of onboard subnetwork protocols (typically MIL 1553, CAN and SpaceWire)
- The use of the CCSDS Space Link Extension (SLE) protocols for ground communication.
- The use of the Space Packet Protocol as the end-to-end routing protocol
- The use of CFDP as an example Application Layer protocol

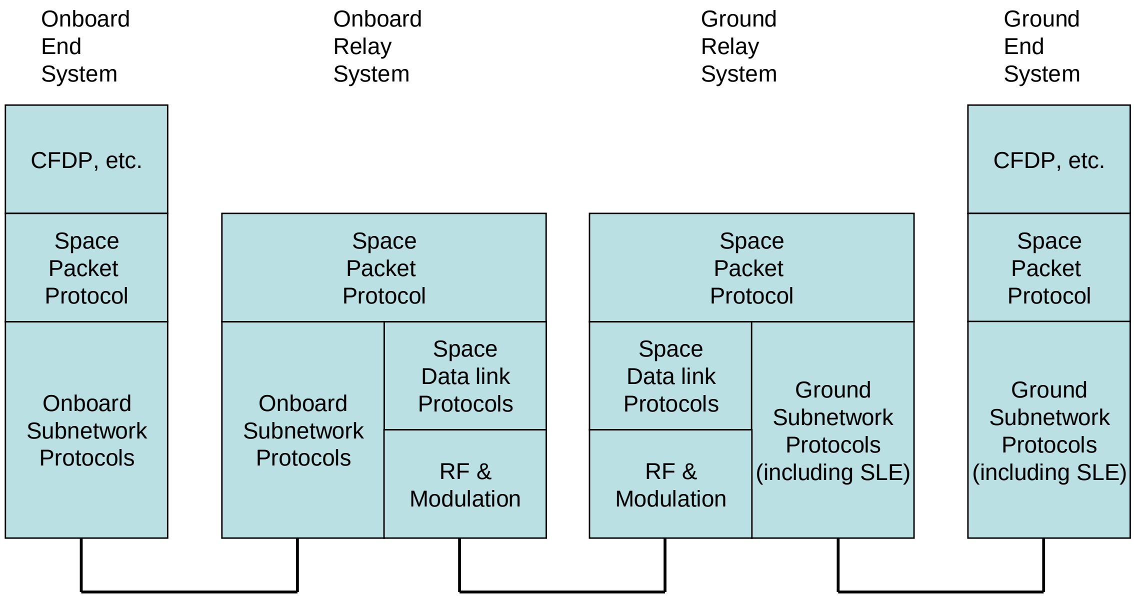

Figure 3-4: End-to-end system with example protocol stacks

Figure 3-5 shows a scenario involving store and forward capability for use in architectures where continuous end-to-end communications is not possible. This is typically the case in planetary exploration missions using orbiting relays. In such a case CFDP provides an end-to-end file transfer capability, automatically taking custody of data in the orbiter and storing it until an opportunity arises for onward communication.

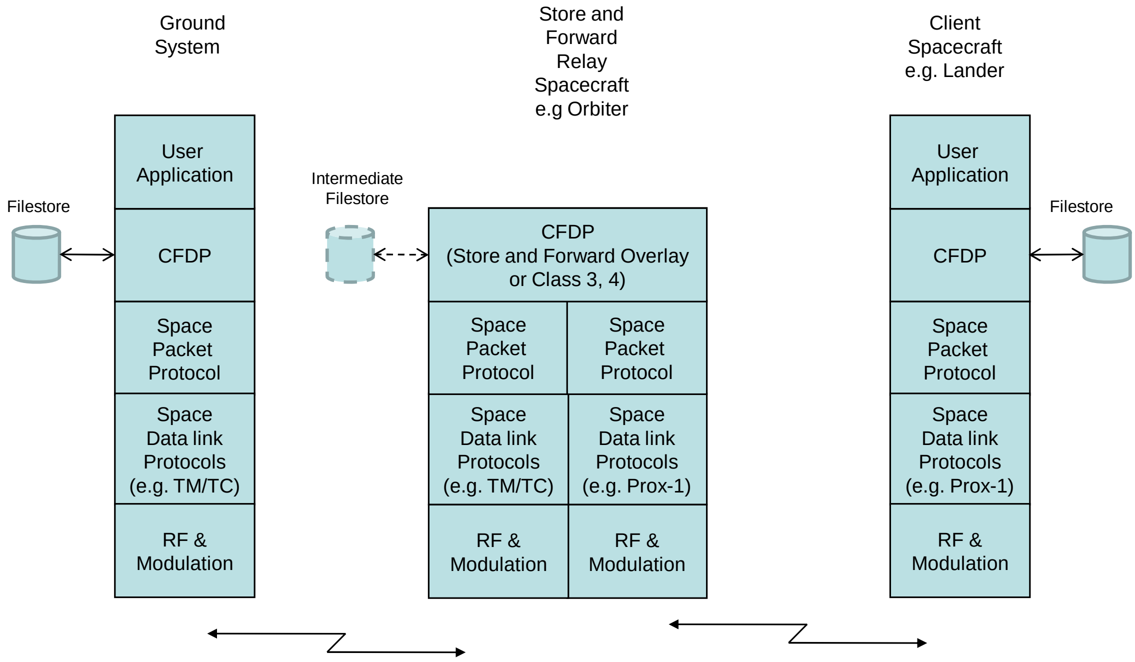

Figure 3-5: Store and forward system with example protocol stacks

4 Standards overview

4.1 Introduction

This section provides an overview of the standards that are used within the E-50 stack of protocols. See Table 3-1 and Table 3-2 for a list of the standards covered in this section.

For each standard, this section provides information on:

- the purpose of the standard and its intended usage
- the capabilities specified in the standard
- any options within the standard
- recommended practices, and pitfalls to avoid, when applying the standard.

4.2 E-ST-50: Communications

4.2.1 Purpose and usage

The purpose of the ECSS-E-ST-50 standard is to guide the development of the end-to-end data communications system for spacecraft. It defines the activities to be performed as part of the space communication system engineering process, in accordance with the ECSS-E-ST-10 *System engineering* standard. In this respect it is a process oriented standard, which is concerned with the way in which the space communication system is achieved.

The standard also includes specific requirements on space communication systems in respect of functionality and performance.

4.2.2 Description

ECSS-E-ST-50 Clause 4 contains a description of space communication systems and architecture and the communications engineering process. It also discusses spacecraft control considerations.

Clause 5 contains the requirements. Clause 5.2 has requirements for the space communication system engineering process. It specifies activities and the formal project documents to be produced. The remaining clauses specify requirements on the function and performance of the space communications system:

- clause 5.3: space communication system
- clause 5.4: telecommanding
- clause 5.5: telemetry
- clause 5.6: space link
- clause 5.7: space network
- clause 5.8: ground network

ECSS-E-ST-50 also includes Annexes containing the document requirements definitions (DRDs), which specify the requirements for the contents of the project documents specified in clause 5.2

4.2.3 Options

There are no options associated with ECSS-E-ST-50.

4.2.4 Recommended practice

ECSS-E-ST-50 should be applied to all missions.

4.3 E-ST-50-02: Ranging and Doppler tracking

4.3.1 Purpose and usage

The Ranging and Doppler Tracking standard defines methods of measuring range and range rate of a spacecraft by an Earth station.

Its stated purpose is to:

> "Ensure compatibility between space agencies' spacecraft transponders and the ranging and Doppler tracking facilities of the Earth stations for the Space Operation, Space Research and Earth Exploration Satellite services.
>
> Ensure, as far as possible, compatibility between space agencies' spacecraft transponders and other networks from which they request support.
>
> Ensure an adequate level of ranging and Doppler tracking accuracy for missions conforming to this standard.
>
> Facilitate the early design of flight hardware and ensure that the resulting interfaces and system performances are compatible with given ranging and Doppler tracking configurations and specifications."

The standard applies to projects with unprocessed ranging accuracies of 2,5 ns to 30 ns and Doppler tracking accuracies of 0,1 mm/s to 1 mm/s. Projects with tracking accuracies less stringent than these may use the CCSDS 401.0-B recommendations.

The standard defines the requirements concerning spacecraft transponder and Earth station equipment for the purposes of ranging and Doppler tracking. It also provides criteria for determining the extent to which the accuracy of the measurements is influenced by equipment effects.

4.3.2 Description

The ranging and Doppler tracking system is a spacecraft tracking system capable of providing information on the range and range rate between a spacecraft and an Earth station. It uses an active transponder on-board the spacecraft for the retransmission to the ground of an Earth-to-Space link signal: ranging signal generation and measurement are performed in the Earth station. As a baseline, it is assumed that the spacecraft transponder is used not only for ranging purposes, but also for receiving telecommand signals from Earth and for transmitting telemetry signals to Earth modulated on the same RF carriers. When a transponder is used exclusively for ranging, the requirements in the standard concerning sharing with telecommand and telemetry have no relevance.

Figure 4-1 shows a functional block diagram of the ranging and Doppler tracking process as addressed in the standard.

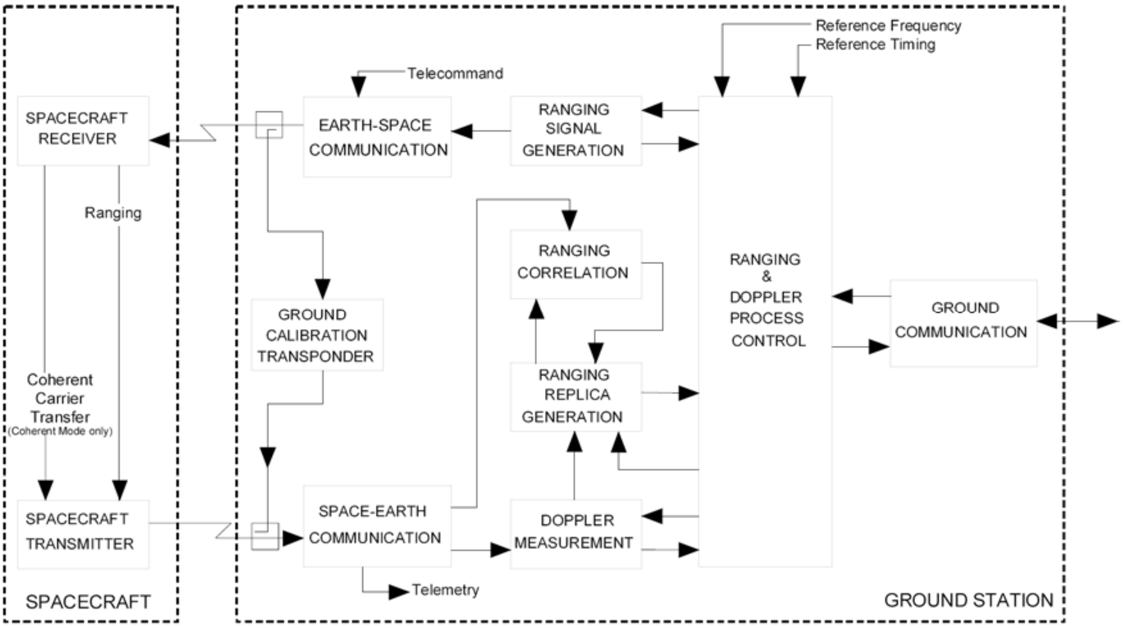

Figure 4-1: Ranging and Doppler tracking - Functional block diagram

The standard describes the functional and performance requirements for:

- the Earth-to-Space link function, employing ground communication, process control, ranging signal generation and Earth-to-Space communication;
- the transponder function, either spacecraft transponder or ground-calibration transponder depending on the application;
- the Space-to-Earth link function, employing Space-to-Earth communication, Doppler measurement, ranging replica generation, ranging correlation, process control and ground communication;
- the link-control function, resident partly in the Space-to-Earth and Earth-to-Space communication and partly in the process control;
- the data-acquisition function, concerned with collection, measurement, processing and transfer of data to the control centre, employing the process control and the ground communication functions.

The standard also describes the process for compatibility testing of implementations. An Annex discusses compatibility with CCSDS-compliant ground station networks.

4.3.3 Options

Table 4-1 presents the options that can be used for ranging and Doppler tracking.

Other ranging and range rate measurement techniques exist such as Differential One-way Ranging and regenerative ranging. However, these techniques are not included in the ECSS-E-ST-50-05 standard. CCSDS are working on recommendations which address these techniques for spacecraft orbital determination.

Table 4-1: Options for Ranging and Doppler tracking

Option	Comment
Tone Frequency	ECSS-E-ST-50-02, “Spectral sharing - Ranging tone selection”
Ranging Code Length	Mission dependent
Simultaneous Ranging and Telecommanding	ECSS-E-ST-50-02, “Modulation – Telecommand and ranging”
RF Frequency and turn-around ratios	ECSS-E-ST-50-05, Clause on “Frequency bands allocated to the Space Radiocommunications services” and Table of “Turnaround frequency ratios for coherent transponder operation”
Transponder Coherency	For deep space operations and for those missions using two-way Doppler measurements
Modulation Index	ECSS-E-ST-50-02, “Spacecraft transponder – Modulation index”

4.3.4 Recommended practice

Ranging tone selection

The nominal ranging tone frequency is selected within the range 100 kHz – 1,5 MHz and in a region of the transponder bandwidth where the group delay is stable within the specified accuracy. The ranging tone frequency selected for a particular mission is usually dependent on whether it is a Near Earth or Deep Space mission. The ranging tone frequency is usually placed at a higher frequency for Deep Space missions than for Near Earth missions due to the ranging accuracy required.

In order to minimise interference between the ranging tone and the telecommand or telemetry signal, careful consideration is required when selecting the subcarriers and tone frequencies as outlined in ECSS-E-ST-50-02, clause on “Spectral sharing – Ranging tone selection – Selection of the frequency: Interference between ranging and telemetry”.

4.4 E-ST-50-05: Radio frequency and modulation

4.4.1 Purpose and usage

The ECSS-E-ST-50-05 standard defines the radio communication techniques used for the transfer of information between spacecraft and Earth stations in both directions, and for the tracking systems used for orbit determination. It should be noted that the standard does not apply to:

- other space telecommunication services (i.e. waveforms used by communications satellite payloads);
- spacecraft supported by data relay satellites.

The standard is applicable to all spacecraft supported by Earth stations and to all controlled Earth stations operating in the Space Operation, Space Research and Earth Exploration Satellite services as defined in the International Telecommunication Union (ITU) Radio Regulations. Under the term Earth Exploration Satellite service, Meteorological Satellite services are also included.

The purpose of the standard is to ensure:

- Compatibility of frequency usage and modulation schemes between space agencies' spacecraft and Earth stations for the Space Operation, Space Research and Earth Exploration Satellite services.
- As far as possible, compatibility between the spacecraft and other networks with which they can work.
- Standardization of frequency usage and modulation schemes within the space projects.
- Conformity of spacecraft and Earth station parameters to international radio regulatory provisions (Radio Regulations of the ITU) and with national regulatory provisions (e.g. national frequency plans).
- That the parameters of spacecraft and Earth stations are properly chosen and listed in advance of their use, thus permitting coordination with other interested parties.
- That, within the above limitations, the frequency usage and modulation schemes are optimized.

4.4.2 Description

The standard lists the frequency bands available for Space Operation, Space Research and Earth Exploration Satellite services, discusses specific conditions for the use of certain frequency bands and outlines frequency assignment procedures.

Requirements and constraints are given on transmitted signals concerning, for example, spectral occupation, RF power levels and protection of other radio services. These requirements and constraints ensure that the transmitted signals do not encroach on other space or non-space communications activities. The standard also defines the permissible modulation methods and parameters.

Annexes discuss, among others:

- Cross support from other networks;
- Protection of the Ariane-5 RF system from interference caused by the Ariane 5 payload spacecraft;
- Specification of the major technical requirements that are relevant for the interface between spacecraft and Earth stations with particular attention to the contents of interface control documentation;
- Conditions where the standard may be deviated from because of mission-specific requirements;
- Differences from CCSDS 401.0-B.

4.4.3 Differences from CCSDS 401.0-B

4.4.3.1 Introduction

This section discusses the differences between ECSS-E-ST-50-05 and the equivalent CCSDS Recommended Standard, CCSDS 401.0-B.

4.4.3.2 Difference 1

4.4.3.2.1 Description

The CCSDS recommendation includes FQPSK-B (Feher QPSK version B), GMSK and baseband filtered/shaped OQPSK modulations. ECSS-E-ST-50-05 only includes GMSK and filtered OQSPK. Furthermore, filtered OQPSK is restricted in ECSS-E-ST-50-05 to specific types of filter with specified roll-off factor whereas the CCSDS recommendation allows an infinite number of filters. It is important to note, however, that these modulations are all of the quaternary offset type and therefore interoperable.

4.4.3.2.2 Rationale

ECSS-E-ST-50-05 has restricted the selection, to avoid including unnecessary modulation schemes with a low likelihood of ever being used. Also, a reduction of the options in the CCSDS recommendation is anticipated: for example, FQPSK-B and shaped OQPSK as well as some filtering options may be removed.

4.4.3.2.3 Potential impact

No impact is to be expected as the FQPSK-B (Feher QPSK version B) and baseband shaped OQPSK modulations are in the process of being removed from the CCSDS 401.0-B RF & Modulation recommendation as no agency intends to make use of them. To be noted also that GMSK and Filtered OQPSK cross-compatibility has been verified and the results are reported in CCSDS 413.0-G, *Bandwidth-Efficient Modulations*.

4.4.3.3 Difference 2

4.4.3.3.1 Description

The CCSDS recommendation includes SRRC-4D 8PSK TCM (available options are 2.0 b/s/Hz, 2.25 b/s/Hz, 2.5 b/s/Hz and 2.75 b/s/Hz; Square Root Raised Cosine filter with a = 0.35 or a = 0.5), GMSK, and filtered OQPSK (with any type of filters). ECSS-E-ST-50-05 only includes SRRC-4D 8PSK TCM (available options are 2.0 b/s/Hz, and 2.5 b/s/Hz; Square Root Raised Cosine filter with a = 0.35 or a = 0.5), GMSK, and filtered OQPSK restricted to specific types of filter with specified roll-off factor.

4.4.3.3.2 Rationale

The SRRC-4D 8PSK TCM restriction was agreed in 2002, as two options of value for ‘a’ were deemed sufficient for the needs of future European missions.

4.4.3.3.3 Potential impact

No impact as ground terminals are flexible enough to support all four options for ‘a’. The restriction on the number of options simplifies the onboard design. To be noted that only European agencies have indicated so far their intention of using SRRC-4D 8PSK TCM on a space mission.

4.4.4 Options

The ECSS-E-ST-50-05 standard covers the Earth-space and space-Earth links for all missions of the Space Operations, Space Research (near-Earth and deep space) and Earth Exploration Satellite services. These different services call for different standards options because of ITU requirements. The standard covers all these options and clearly identifies which option is for which service. The main parameters affected by the type of service are:

- Frequency band;
- Requirements on transmitted signals, namely spectral occupancy, RF power levels, protection of other radio services;
- Permissible modulation methods and parameters
- A number of technical requirements that are relevant for the interface between spacecraft and Earth stations;
- Operational aspects, such as signal acquisition.

4.4.5 Recommended practice

4.4.5.1 Frequency allocations

Special conditions on the use of frequency bands are provided in full detail in clause 4.2 of the standard. It is important to note that severe restrictions apply on the use of the 2110-2120 MHz band (deep space) for new frequency assignments. The projects should by all means avoid considering this band for new frequency assignments.

Most Earth Exploration Satellite missions currently operate their high speed payload telemetry data downlink in the 8025-8400 MHz frequency band. The bandwidth limitations, the congestion of the band, and the requirements on out-of-band emissions in the 8400-8450 MHz frequency band make it difficult to use this frequency band for very high data rates. The 25.5-27 GHz is also allocated for the Earth Exploration Satellite missions and should be considered for missions requiring these very high data rates for their payload downlink telemetry.

At the date of publication, there is no ground infrastructure in Europe which supports the frequency bands 25.5-27 GHz, 37-38 GHz and 40.0-40.5 GHz.

When selecting a modulation scheme for the telecommand and the telemetry links, the project must take into account the possible need for a ranging service. This may imply the use of one or more modulation schemes for the telecommand and/or telemetry links in order for the satellite to be compatible with ECSS-E-ST-50-02.

The International Telecommunications Union recognises two categories of Space Science missions which have different frequency band allocations; Deep Space missions which are at a distance greater than 2 million kilometres from the Earth and Near Earth missions which are at a distance of less than 2 million kilometres from the Earth.

NOTE The ECSS-E-ST-50-05 standard provides more details on the TT&C frequency allocations and associated bandwidth constraints.

4.4.5.2 The ‘Transponder Dual Lock’ issue

For the latest generation of Near Earth and Lagrange point missions, the TT&C receiver subsystem combines the uplink signals from both low gain antennas by using a 3dB hybrid combiner. This ensures that the ‘best’ TC uplink signal is seen by both the prime and redundant transponder receivers simultaneously. Note that this architecture assumes that it has been checked that any interfering area induced by antennae coupling is out of the nominal LOS (Line Of Sight) domain. The ESA Near Earth mission Cryosat-2 uses this type of TT&C architecture as shown in Figure 4-2. Cryosat-2 is an Earth Exploration Satellite.

This TT&C configuration avoids the situation where one transponder could be potentially operating very close to its receiver telecommand threshold (e.g. at ~ -122 dBm) and therefore not operating at a sufficiently low bit error rate on the telecommand channel to ensure valid telecommands are received and processed by the packet telecommand decoder, whilst at the same time the other transponder receiver is operating well above its telecommand threshold level. This operational scenario is known as the ‘Transponder Dual Lock’ condition and can potentially cause the rejection of telecommands on the telecommand uplink.

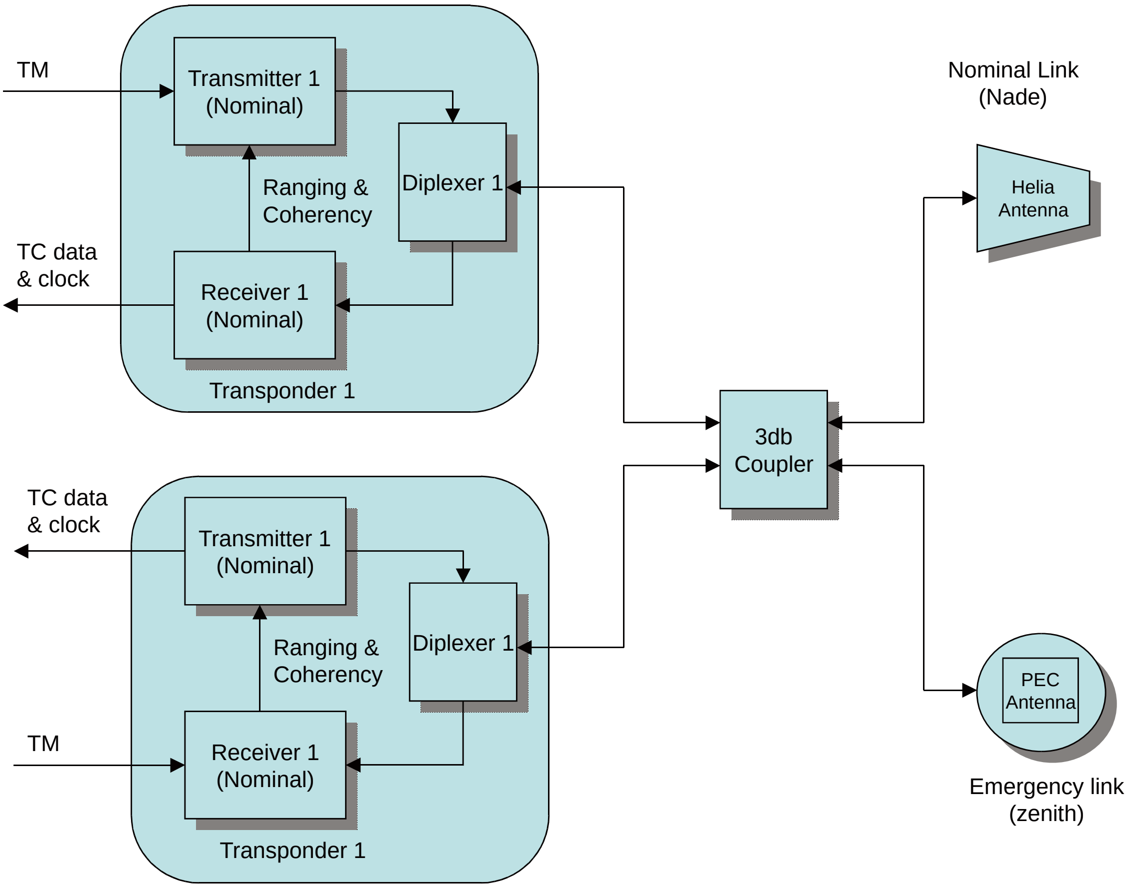

Figure 4-2: Cryosat-2 S-band TT&C subsystem

It is recommended that the COP-1 sequence controlled service (AD mode) should be used for all Near Earth missions in order to ensure a reliable retransmission capability in the case of lost telecommands. It is also strongly recommended that a realistic compatibility test is conducted, prior to launch, using a representative ground station infrastructure. This test configuration needs to include all the elements of the ground station and mission control system which are considered necessary in order to simulate the realistic mission operational scenario.

The operational parameters for the COP-1 protocol need to be tailored for individual missions, based upon the Earth to Satellite distance, in order to optimise the performance. The robustness of the communications uplink when operating in COP-1 sequence controlled service (AD mode) as opposed to expedited service (BD mode), is considered to be the best operational remedy to the problem of frame rejection, which can potentially be caused by the transponder receivers if they are operating in the 'dual lock' condition.

A number of additional recommendations have been proposed, both from an operational perspective and from an on-board system design configuration, which alleviate the 'dual lock' condition for future missions. More information can be found in the report TEC-ETC/2008.66/PMcM "The transponder dual lock issue: Background and recommended remedial actions, June 2008" .

4.5 E-ST-50-01: Telemetry synchronization and channel coding

4.5.1 Purpose and usage

The ECSS-E-ST-50-01 standard, *Space data links - Telemetry synchronization and channel coding*, defines procedures and data structures in the Synchronization and Channel Coding Sublayer of the Data Link Layer discussed in section 3.3.2. The equivalent CCSDS Recommended Standard is CCSDS 131.0-B, *TM Synchronization and Channel Coding*.

ECSS-E-ST-50-01 establishes a common implementation of space telemetry channel coding systems. It addresses:

- coding for improvement of link margin;
- frame synchronization applicable to both coded and uncoded telemetry;
- randomisation to achieve sufficient bit transition density for bit synchronization.

The specification does not attempt to quantify the relative coding gain or the merits of each scheme, nor the design requirements for encoders or decoders. Section 5.2.2 gives some performance data for the coding schemes specified in the standard and further performance details can be found in the informational report CCSDS 130.1-G.

4.5.2 Description

The standard addresses the issue of telemetry channel coding. Telemetry channel coding enables reconstruction of the data with low error probability, thus improving the performance of the channel.

A channel code is the set of rules that specify the transformation of elements of a source alphabet to elements of a code alphabet. The elements of the source alphabet and of the code alphabet are called symbols. Depending on the code, the symbols can consist of one or more bits.

A convolutional code is 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. The constraint length is the number of consecutive input bits that are used to determine the value of the output symbols at any time.

For telecommunication channels which are bandwidth-constrained and cannot accommodate the increase in bandwidth caused by the basic convolutional code, the standard also specifies a punctured convolutional code which has the advantage of smaller bandwidth expansion.

The Reed-Solomon (R-S) code is a powerful burst error correcting code. In addition, the code has the capability to indicate the presence of uncorrectable errors, with an extremely low undetected error rate. The Reed-Solomon code has the advantage of smaller bandwidth expansion than the convolutional code.

Concatenation is the use of two or more codes to process data sequentially, with the output of one encoder used as the input of the next. In a concatenated coding system, the first encoding algorithm that is applied to the data stream is called the outer code. To achieve a greater coding gain than the one that can be provided by the convolutional code or Reed-Solomon code alone, a concatenation of the convolutional code with the Reed-Solomon code can be used for improved performance.

A turbo code is a block code formed by combining two component recursive convolutional codes. A turbo code takes as input a block of information bits. The input block is sent unchanged to the first component code and bit-wise interleaved to the second component code. The interleaving process is a fixed bit-by-bit permutation of the entire input block. The output is formed by the parity symbols contributed by each component code plus a replica of the information bits. The turbo codes specified in the standard can be used to obtain even greater coding gain where the environment tolerates the bandwidth overhead.

The methods for synchronization specified in the standard apply to all telemetry channels, coded or uncoded. An attached sync marker (ASM) is attached to the code block or transfer frame. The ASM can also be used for resolution of data ambiguity (sense of ‘1’ and ‘0’) if data ambiguity is not resolved by the modulation method used.

The data unit that consists of the ASM and the Reed-Solomon or turbo codeblock or transfer frame is called the channel access data unit (CADU), as shown in Figure 4-3.

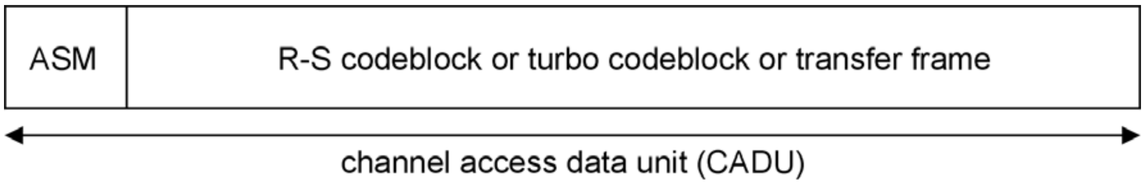

Figure 4-3: Format of channel access data unit (CADU)

Successful bit synchronization at the receiving end depends on the incoming signal having a minimum bit transition density. The standard specifies the method of pseudo-randomizing the data to improve bit transition density.

Figure 4-4 and Figure 4-5 give an overview of how pseudo-randomization and synchronization are combined with the different coding options at the sending and receiving end. At the sending end, the order of convolutional encoding and modulation is implementation dependent. At the receiving end, the order of demodulation, frame synchronization and convolutional decoding is implementation dependent. The figures are not intended to imply any hardware or software configuration in a real system.

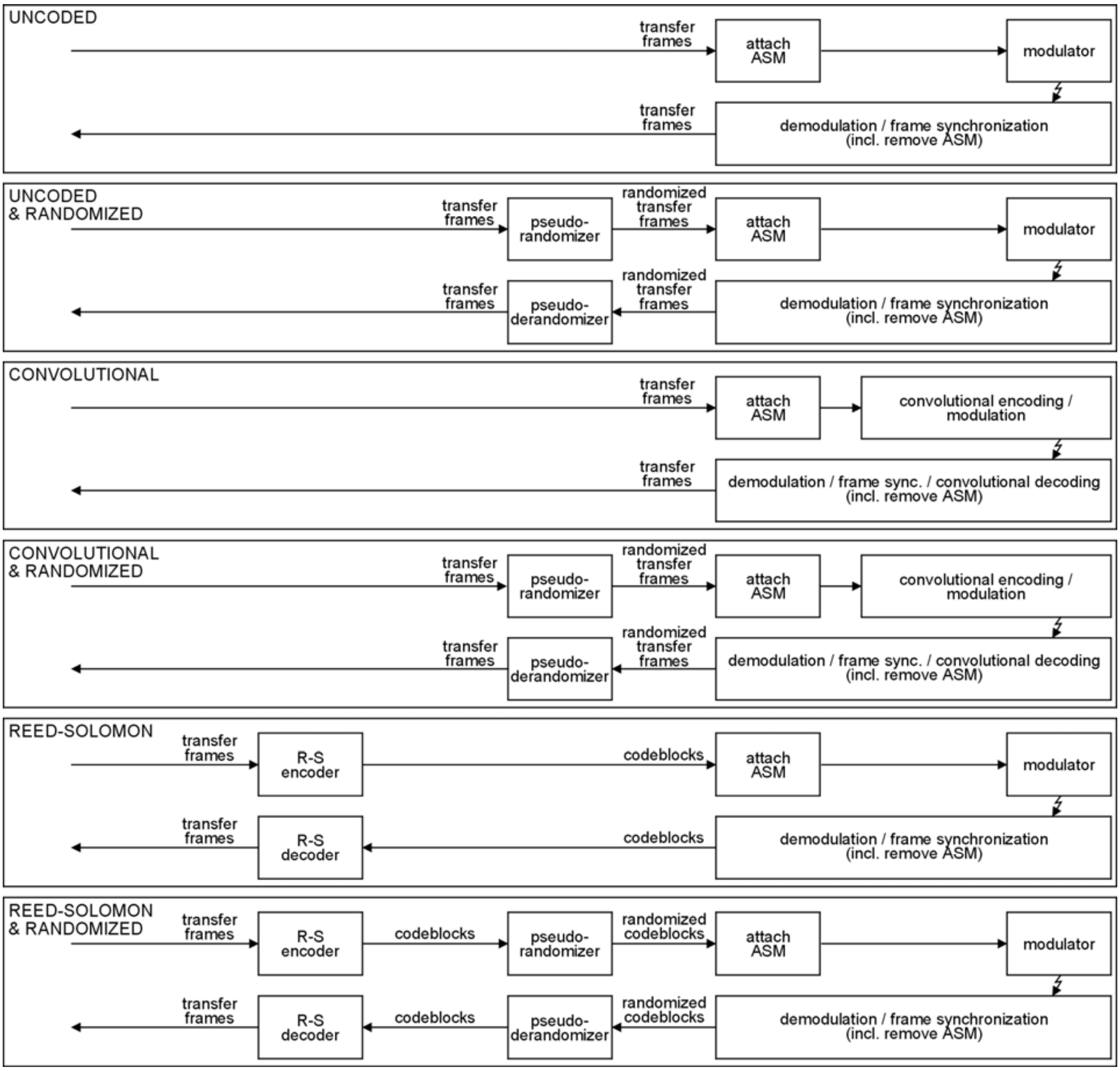

Figure 4-4: Coding, randomization and synchronization configurations (1)

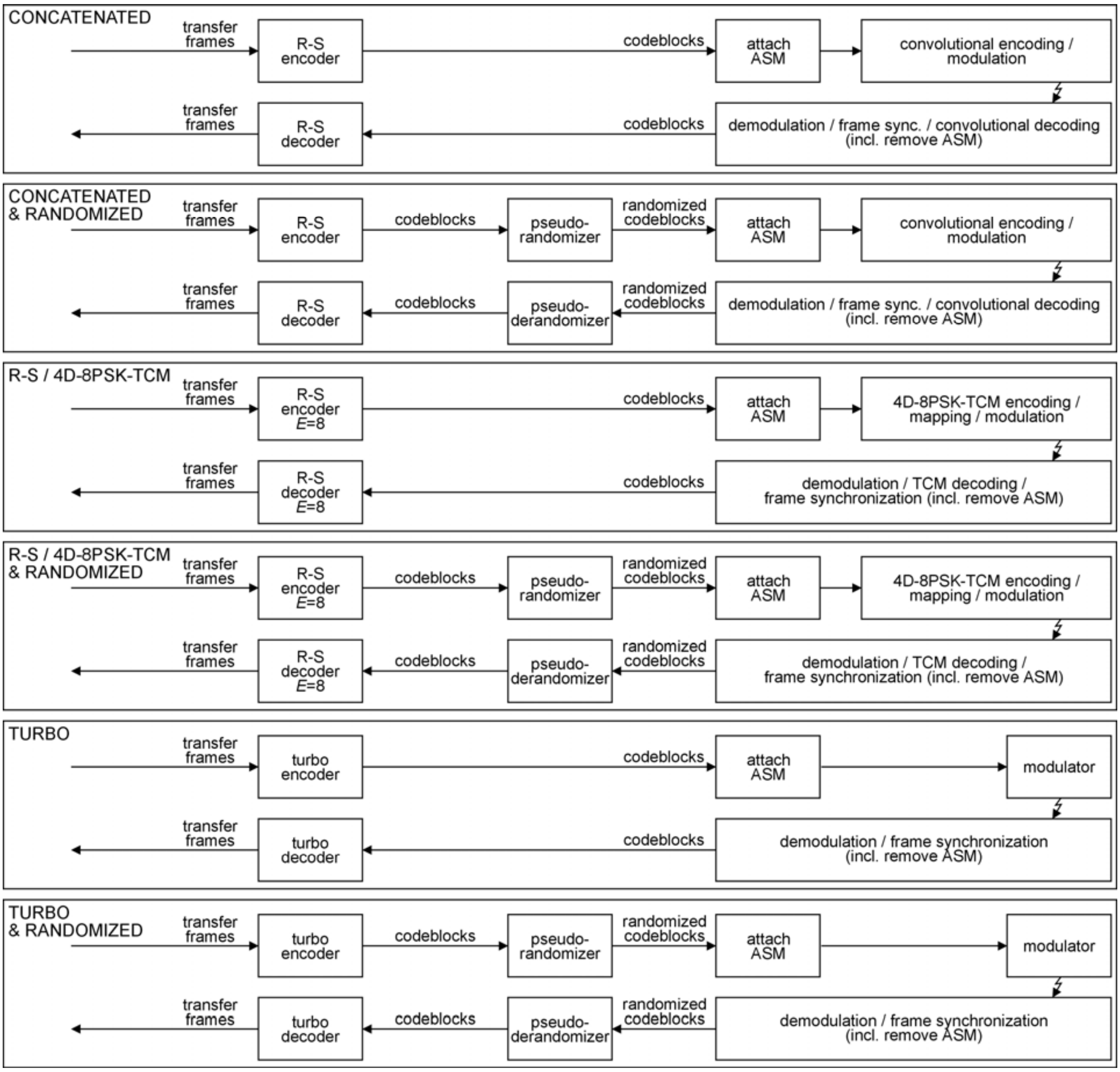

Figure 4-5: Coding, randomization and synchronization configurations (2)

Annexes discuss:

- Reed-Solomon transformations and coefficients;
- telemetry transfer frame lengths compatible with convolutional coding, Reed-Solomon coding and turbo coding;
- application profiles with reference to coding scheme and performances;
- changes from the preceding ESA PSS standard and differences from CCSDS recommendations (see section 4.5.3);
- mission configuration parameters, showing the options and values that can be taken by the parameters specified in ECSS-E-ST-50-01;
- guidance on turbo code patent rights.

4.5.3 Differences from CCSDS 131.0-B-1

4.5.3.1 Introduction

This section discusses the differences between ECSS-E-ST-50-01 and the equivalent CCSDS Recommended Standard, CCSDS 131.0-B-1. ECSS-E-ST-50-01 down-selects coding options from the wider range specified in CCSDS 131.0-B-1.

4.5.3.2 Difference 1

4.5.3.2.1 Description

The CCSDS recommendations include Reed-Solomon codes with E=16 and with E=8. ECSS-E-ST-50-01 includes Reed-Solomon codes with E=16. ECSS-E-ST-50-01 also includes Reed-Solomon codes with E=8, but restricted to channels where the modulation scheme is 4D-8PSK-TCM. In that case, interleaving depth I=8 is mandatory.

4.5.3.2.2 Rationale

Reliability. For E=8, there is an increase in the probability of decoding errors. A decoding error occurs when the decoder finds a codeword other than the transmitted codeword (i.e. when the error pattern is within a distance less than E symbols of a nonzero R-S codeword). For E=8, the general probability of a decoder error is around 10^{-5}, compared to 10^{-13} for E=16. The specific use of E=8 with 4D-8PSK-TCM has a better error detection probability than in the general case, provided that interleaving depth I=8 is used, as required by ECSS-E-ST-50-01. In that case, probability of an R-S decoding error is around 10^{-11} for a decoder output BER of 10^{-5} (see Annex G for more details). Decoding error contribution to overall BER is negligible.

The improvement in the code rate by using E=8 instead of E=16 is limited (0.94 instead of 0.87). A CRC can be included in the transfer frame to identify undetected decoding errors with E=8 but further reduces the difference of efficiency, without giving the same level of reliability as for E=16. Nevertheless, for bandwidth limited high rate telemetry channels, the improvement in code rate (8%) provided by using E=8 instead of E=16 can be significant. Therefore, the ECSS-E-ST-50-01 standard authorizes the usage of R-S with E=8 as outer code to the 8PSK-TCM coded modulation specific to high rate telemetry links.

4.5.3.2.3 Potential impact:

Interoperability of ECSS compliant ground systems with CCSDS compliant satellites implementing R-S with E=8 on the downlink. Nevertheless, this impact is very unlikely to occur since, due to the rationale developed above, only bandwidth limited high rate telemetry links are likely to use E=8 in conjunction with 8PSK-TCM, which is the combination allowed by the ECSS-E-ST-50-01A standard.

4.5.3.3 Difference 2

4.5.3.3.1 Description

ECSS-E-ST-50-01 and the CCSDS recommendations include concatenated codes, with the Reed-Solomon code as the outer code and the convolutional code as the inner code. The additional Reed-Solomon codes in the CCSDS recommendations lead to a greater number of concatenated codes.

4.5.3.3.2 Rationale

This difference is a result of the restrictions in ECSS-E-ST-50-01 on the use of Reed-Solomon codes with E=8. See *Difference 1* above.

4.5.3.3.3 Potential impact:

No potential impact has been identified.

4.5.3.4 Difference 3

4.5.3.4.1 Description

The CCSDS recommendations include turbo codes with rates 1/2, 1/3, 1/4, and 1/6. ECSS-E-ST-50-01 only includes turbo codes with rates 1/2 and 1/4.

4.5.3.4.2 Rationale

Turbo code with rate 1/6 gains only 0.2 dB at FER=10^{-4} (frame length = 8920 bits) with respect to turbo code rate 1/4 and expands the used frequency band dramatically. Turbo code with rate 1/2 is included in ECSS-E-ST-50-01 since it gives a 1.5 dB gain over the typical concatenated encoding scheme while slightly reducing the bandwidth. The performance of turbo code rate 1/3 is between turbo code rate 1/4 and turbo code rate 1/2 and therefore it is not included. If limitation of bandwidth is an issue, then turbo code rate 1/2 can be used, if not then turbo code rate 1/4 can be used. No mission requiring the use of turbo code rate 1/3 or 1/6 has been identified.

4.5.3.4.3 Potential impact

No potential impact has been identified.

4.5.4 Options

Annex G of the standard lists the mission configuration parameters. These include the channel coding scheme, the transfer frame length, and the presence or absence of pseudo-randomization. Other options vary depending on the choice of coding scheme.

The proforma of the Protocol Implementation Conformance Statement (PICS) can also be used as a tool for considering the options of a standard. For ECSS-E-ST-50-01, there is a draft PICS proforma for a sending end implementation (see Annex A of this handbook) and for a receiving end implementation (see Annex B of this handbook). For an overview of PICS, see section 5.8.

4.5.5 Recommended practice

4.5.5.1 Preferred coding schemes

Missions calculate the frame error rate (FER) that is required for a link in order to meet mission objectives and then select the coding scheme and other link characteristics in order to achieve the required FER. Annex D of the standard gives preferred coding schemes to achieve frame error rates (FER) in the range 10^{-4} to 10^{-6}. Compressed data may require lower residual FERs and are outside the scope of the annex.

NOTE See section 5.2.2 for discussion on the performance of the coding schemes.

4.5.5.2 Choice of basic or punctured convolutional code

If the convolutional code is selected for a link, there remains a choice between the basic and punctured code. If the bandwidth occupation resulting from using the basic convolutional code is compatible with the maximum occupied bandwidth specified for the mission then it is recommended practice to select the basic convolutional code in preference to the punctured code.

NOTE See section 5.2.2 for discussion on selecting coding options.

4.5.5.3 Use of pseudo-randomizer

In order to maintain bit synchronization with the received telemetry signal, the data capture system depends on the incoming signal having a minimum bit transition density. None of the coding schemes defined in ECSS-E-ST-50-01 can, by itself, guarantee sufficient channel symbol transitions to keep receiver symbol synchronizers in lock. The standard therefore defines a pseudo-randomizer that can be used to increase the symbol transition density. It is recommended that missions should use the pseudo-randomizer. Missions have encountered problems with their telemetry links because the pseudo-randomizer was not used and sufficient randomness was not ensured by other means.

4.5.5.4 Pseudo-randomizer for high data rate telemetry

For some high data rate telemetry links (typically above 2 megasymbols/second), the period of the pseudo-random binary sequence in the specified pseudo-randomizer may not be sufficient. At the time of issue of ECSS-E-ST-50-01, no randomization scheme had been agreed within ECSS for these links. There is a study in progress.

4.6 E-ST-50-03: Telemetry transfer frame protocol

4.6.1 Purpose and usage

The ECSS-E-ST-50-03 standard, *Space data links - Telemetry transfer frame protocol*, defines procedures and data structures in the Data Link Protocol Sublayer of the Data Link Layer discussed in section 3.3.2. The equivalent CCSDS Recommended Standard is CCSDS 132.0-B, *TM Space Data Link Protocol*.

The telemetry transfer frame protocol provides a standardized data structure for the transmission of space-acquired data over a telemetry space data link. Usually, the source of the data is located in space and the receiver is located on the ground. However, the standard may also be used for space-to-space telemetry data links.

The telemetry transfer frame is a data structure that provides an envelope for transmitting data units such as packets over a noisy space communications channel. It carries information in the transfer frame primary header that enables the receiving 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. It is compatible with the telemetry channel coding defined in ECSS-E-ST-50-01; thus the transmitted data can be recovered with extremely high reliability.

Multiple, individual, asynchronous application processes on board a space vehicle can generate variable-length packets at different rates, and these packets can then be multiplexed together into a synchronous stream of fixed-length coded transfer frames for reliable transmission to the receiving system. The transfer frame data structure provides the elements to enable the variable-length packets from a number of application processes on a spacecraft to be multiplexed into a sequence of fixed-length frames. Short packets can be contained in a single frame, while longer ones can 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.

In a typical application, the transfer frame data field of a telemetry transfer frame is used to transport packets. However, the field can be used for other data units. Within a physical channel, there can be some virtual channels where the frames carry packets and other virtual channels where the frames carry other data units.

The term Packet Telemetry is sometimes applied to the protocol stack consisting of the following set of standards:

- CCSDS 133.0-B, *Space Packet Protocol*
- ECSS-E-ST-50-03 (or CCSDS 132.0-B)
- ECSS-E-ST-50-01 (or CCSDS 131.0-B)

The term Packet Telemetry was used in earlier CCSDS and ESA PSS standards but it is not used in recent standards.

4.6.2 Description

4.6.2.1 Overview of frames

The standard contains the definition for Telemetry Transfer Frames. The frames are fixed-length data structures, suitable for transmission at a constant frame rate on a space data channel. The fixed length frames provide a mechanism for transferring various data types including CCSDS Space Packets over the channel.

The telemetry transfer frame protocol can operate in various configurations of the telemetry space link, depending on the telemetry channel coding scheme and security options selected. Proper operation of the protocol can only be obtained if a high-quality data channel is provided between the peer entities of the protocol (see section 4.5).

The format of the TM Transfer Frame is shown in Figure 4-6.

Note that the ASM added by the synchronization and channel coding procedures is not considered to be part of the TM Transfer Frame. The length of the ASM is not included when calculating the length of the frame.

<table>
<tr><td rowspan="3">TRANSFER FRAME PRIMARY HEADER</td><td colspan="3">TRANSFER FRAME SECONDARY HEADER (optional)</td><td rowspan="3">TRANSFER FRAME DATA FIELD</td><td colspan="2">TRANSFER FRAME TRAILER (optional)</td></tr>
<tr><td colspan="2">TRANSFER FRAME SECONDARY HEADER ID</td><td rowspan="2">TRANSFER FRAME SECONDARY HEADER DATA FIELD</td><td rowspan="2">OPERATIONAL CONTROL FIELD (optional)</td><td rowspan="2">FRAME ERROR CONTROL FIELD (optional)</td></tr>
<tr><td>TRANSFER FRAME SECONDARY HEADER VERSION NUMBER</td><td>TRANSFER FRAME SECONDARY HEADER LENGTH</td></tr>
<tr><td>6 octets</td><td>2 bits</td><td>6 bits</td><td>up to 63 octets</td><td>variable</td><td>4 octets</td><td>2 octets</td></tr>
<tr><td></td><td colspan="3">up to 64 octets</td><td></td><td colspan="2">up to 6 octets</td></tr>
</table>

Figure 4-6: TM Transfer Frame

4.6.2.2 Master channels and virtual channels

The TM Transfer Frame supports the division of the physical channel into master channels and virtual channels by means of identifier fields in the frame header.

A master channel is identified by the values of the Transfer Frame Version Number (TFVN) and the Spacecraft Identifier (SCID). Within a given physical channel, a master channel consists of all the frames which have the same Transfer Frame Version Number and the same Spacecraft Identifier. On a physical channel which carries TM Transfer Frames, the standard makes the restriction that all the frames have the same Transfer Frame Version Number.

For a typical space mission, all the frames on a physical channel have the same value for the Spacecraft Identifier, so in this case there is one master channel on the physical channel. However, multiple master channels can share a physical channel, which, for example, can be the case when one spacecraft is transporting another spacecraft such as a probe.

A master channel is divided into virtual channels using the Virtual Channel Identifier field. In a TM Transfer Frame it is a 3-bit field and therefore supports up to eight virtual channels on a master channel.

Virtual channels enable one physical channel to be shared among multiple higher-layer data streams, each of which can have different characteristics. The mechanisms and parameters for sharing access by the virtual channels to the physical channel are implementation dependent.

4.6.2.3 Data carried in the Transfer Frame Data Field

Every TM Transfer Frame contains the Transfer Frame Data Field, which is the main data-carrying field in the frame. Within a virtual channel, the length of the field is static during a mission phase.

There are status fields in the frame header which are related to the use of the field. The Synchronisation Flag indicates how the Transfer Frame Data Field is used:

- When Synchronisation Flag is '0', the Transfer Frame Data Field carries packets;
- When Synchronisation Flag is '1', the Transfer Frame Data Field carries other data units.

The use is fixed for a virtual channel for a mission phase.

When the field is carrying packets, then the packets are restricted to certain packet types. Each packet has a defined Packet Version Number, as specified in CCSDS 135.0-B-3. The Packet Version Number occupies the first three bits of the packet header. When extracting packets from the Transfer Frame Data Field at the receiving end, the telemetry transfer frame protocol uses the Packet Version Number to locate the length fields in a packet header.

When the field is carrying other data units, then the contents of the Transfer Frame Data Field are implementation dependent. The standard does not specify the format or handling of the data. The data can consist of any type of data units.

For legacy reasons, the standard includes the specification of asynchronously inserted data, for transporting data that has been recorded in an on-board memory device. This is an optional use. It is one example of using frames with Synchronisation Flag set to '1'. The specification does not restrict the use of such frames for other purposes.

Within a physical channel or master channel, there can be some virtual channels that are carrying packets and some virtual channels that are carrying other data units, in any combination supported by the implementation.

4.6.2.4 Other data carried in a TM Transfer Frame

In addition, the TM Transfer Frame has two optional fields for carrying data: the Transfer Frame Secondary Header and the Operational Control Field.

The Transfer Frame Secondary Header can be used to carry fixed-length mission-specific data. Its use can be associated with a virtual channel or with a master channel. The length of the data carried in the Transfer Frame Secondary Header can be from 1 to 63 octets.

If the Transfer Frame Secondary Header is associated with a master channel, then the Transfer Frame Secondary Header is present in every frame on the master channel. In this case, the Transfer Frame Secondary Header has the same length for all virtual channels of the master channel.

If the Transfer Frame Secondary Header is associated with a given virtual channel, then the Transfer Frame Secondary Headers of other virtual channels can be absent or can be used for other purposes. For example, the Transfer Frame Secondary Header can have a different length on different virtual channels.

The standard includes the specification of the Transfer Frame Secondary Header to provide a 24-bit extended virtual channel frame count. This is an optional use for the Transfer Frame Secondary Header. The specification does not restrict the use of the Transfer Frame Secondary Header for other purposes.

The Operational Control Field can be used to carry status information to control the operation of the telecommand space link or other spacecraft activities. Like the Transfer Frame Secondary Header, its use can be associated with a virtual channel or with a master channel. The length is always 32 bits.

4.6.2.5 Transfer Frame Primary Header

The fields of the Transfer Frame Primary Header are shown in Figure 4-7. They include:

- Transfer Frame Version Number, which is always set to 00;
- Spacecraft Identifier (the Secretariat of the CCSDS assigns Spacecraft Identifiers according to the procedures in CCSDS 320.0-B – see 4.15);
- Virtual Channel Identifier;
- Flag to indicate presence of the Operational Control Field;
- Master Channel Frame Count, which provides a running count of the frames transmitted through the same master channel;
- Virtual Channel Frame Count, which provides a running count of the frames transmitted through the same virtual channel;
- Flag to indicate presence of the Transfer Frame Secondary Header;
- Synchronisation Flag to indicate the use of the Transfer Frame Data Field, as described above.

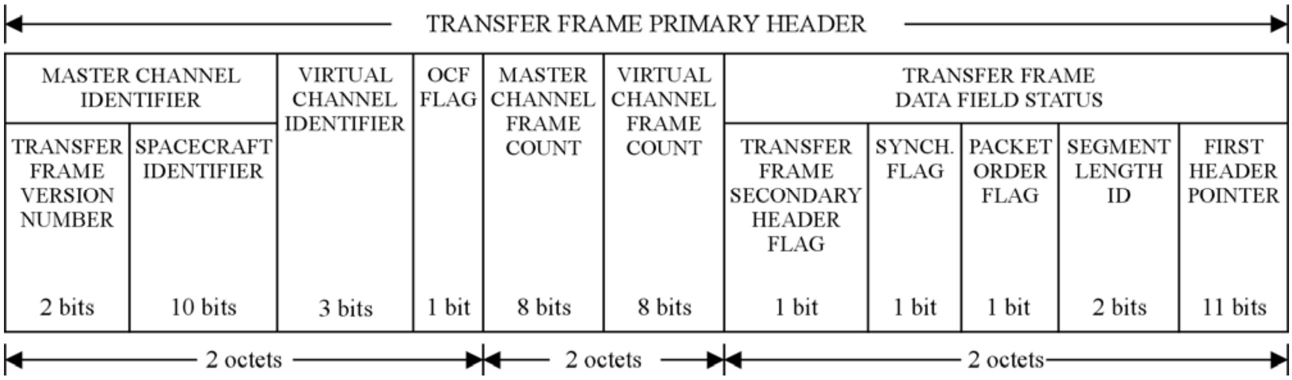

Figure 4-7: TM Transfer Frame Primary Header

The use of the remaining fields in the header depends on the value of the Synchronisation Flag:

- When the Synchronisation Flag is '0', the values in the fields are specified by the standard. In this case, the First Header Pointer carries information on the data within the Transfer Frame Data Field.
- When the Synchronisation Flag is '1', the values in the fields are undefined by the standard. In this case, the use of the fields is implementation dependent.

4.6.2.6 Frame Error Control Field

The Frame Error Control Field provides a capability for detecting errors which can be introduced into the frame during the transmission and data handling process. It uses a cyclic redundancy code (CRC).

If the physical channel uses Reed-Solomon encoding, then the Frame Error Control Field is optional. Otherwise the use of the field is mandatory.

The use of the field is fixed for a physical channel for a mission phase.

4.6.2.7 Annexes

The standard contains annexes addressing:

- implementation of the CRC for the Frame Error Control Field;
- changes from the preceding ESA PSS standard;
- differences from CCSDS recommendations;
- mission configuration parameters, showing the options and values that can be taken by the parameters specified in ECSS-E-ST-50-03.

4.6.3 Differences from CCSDS 132.0-B-1

4.6.3.1 Introduction

This section discusses the differences between ECSS-E-ST-50-03 and the equivalent CCSDS recommendation, CCSDS 132.0-B-1.

4.6.3.2 Difference 1

4.6.3.2.1 Description

ECSS-E-ST-50-03 defines the use of the Transfer Frame Secondary Header for extending the Virtual Channel Frame Count. The extended count is not specified in the CCSDS recommendations.

4.6.3.2.2 Rationale

The extended Virtual Channel Frame Count is a feature from the preceding ESA PSS standard that does not exist in CCSDS and that has been used by ESA missions.

The use of this extended count depends on ground processing and on the use of virtual channels by a mission. In general, it provides means for more accurate sequencing of received frames and the resolution of sequence ambiguities within a virtual channel. For example:

- The extraction of packet data on ground may be complicated by the presence of streams of Transfer Frames previously recorded on board the spacecraft and played back later, generally through a real-time virtual channel. A large count number can ease retrieval.
- The extended Virtual Channel Frame Count can improve the means for checking the correct sequencing of received data and for identification of losses, without reference to the "ground receive time".

4.6.3.2.3 Potential impact

- The extended Virtual Channel Frame Count is optional.
- The use of the extended Virtual Channel Frame Count does not impact interoperability at frame level (e.g. use of RAF/RCF SLE services).
- If the extended Virtual Channel Frame Count is in use for a virtual channel or master channel, then it excludes any other use of the Transfer Frame Secondary Header on that channel.

4.6.3.3 Difference 2

4.6.3.3.1 Description

In CCSDS 132.0-B-1, the name OID Transfer Frame is used for a telemetry transfer frame with a specific value in its First Header Pointer, meaning that it has Only Idle Data in its Transfer Frame Data Field. In ECSS-E-ST-50-03, the term OID Transfer Frame is not used but the meaning of the First Header Pointer value is the same.

4.6.3.3.2 Rationale

An OID Transfer Frame can carry useful data in the Transfer Frame Secondary Header and the Operational Control Field. The "OID" part of the name only applies to the Transfer Frame Data Field, so ECSS-E-ST-50-03 does not apply it to the frame as a whole.

4.6.3.3.3 Potential impact

The difference is only in the use of the name and therefore has no direct impact on implementations or missions.

4.6.4 Options

Annex D of the standard lists the mission configuration parameters. Most of the parameters are fixed for a mission phase, and, in practice, are often fixed for the duration of the mission.

The annex shows the options and values that can be taken by the parameters as specified in the standard. It is the responsibility of mission designers to check the availability of support for the options and values selected for the mission.

For the physical channel, the parameters include the length of the TM Transfer Frame. Note that TM Transfer Frames are fixed length, so throughout a mission phase every transfer frame transmitted on the physical channel has the same length. Other options and parameters include the presence of the Frame Error Control Field, and the Spacecraft Identifier values used on the channel.

There are further options and parameters for a master channel and for a virtual channel, including the use of the Transfer Frame Secondary Header and the Operational Control Field.

The value for the Synchronization Flag is a parameter of a virtual channel. For a virtual channel with Synchronization Flag set to '0', there are additional parameters for the maximum packet length and for the packet types used on the virtual channel.

The proforma of the Protocol Implementation Conformance Statement (PICS) can also be used as a tool for considering the options of a standard. For ECSS-E-ST-50-03, there is a draft PICS proforma for a sending end implementation (see Annex C of this handbook) and for a receiving end implementation (see Annex D of this handbook). For an overview of PICS, see section 5.8.

4.6.5 Recommended practice

4.6.5.1 Use of the Operational Control Field

In the standard, the Operational Control Field can be used to carry a Type-1-Report or a Type-2-Report. It is recommended practice that the Operational Control Field is only used for a Type-1-Report, which is a Communications Link Control Word (see section 4.7.3).

4.6.5.2 Asynchronously inserted data

The standard includes the specification of asynchronously inserted data, for transporting data that has been recorded in an on-board memory device. It is recommended practice that asynchronously inserted data is only used for legacy support.

4.6.5.3 Virtual channels

Virtual channels enable the various on-board data sources to be given access to the physical channel by assigning them transmission capacity on a frame-by-frame basis. They can be used to separate sources or destinations with different characteristics. For example, if a payload contains an imaging instrument which produces long packets with many thousands of octets, and a number of other instruments which generate smaller packets, a possible system architecture is 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 can also be used to enable easy separation in the receiving system of data streams for different destinations.

4.6.5.4 Maximum packet length

If a virtual channel is carrying packets, there is a mission configuration parameter for the maximum packet length. It is recommended practice to consider timing issues when selecting the maximum length. While a long packet is being transmitted, other data sources cannot access the virtual channel. The maximum length places an upper limit on the waiting time imposed on the other data sources. The upper limit for the waiting time depends on the maximum latency times defined for the data sources.

4.6.5.5 Frames with detected errors

At the receiving end, some frames may be found to contain uncorrectable errors. For a virtual channel which carries packets, it is recommended practice that the Transfer Frame Data Fields of these frames are not used by the packet extraction function.

4.6.5.6 Gaps in the sequence of frames

The receiving end can use the count fields of a frame (Master Channel Frame Count, Virtual Channel Frame Count and the optional extended Virtual Channel Frame Count) to detect gaps in the arriving stream of frames. Whenever a missing frame is detected or a frame is found to contain uncorrectable errors, it is recommended practice to signal that a gap has occurred.

4.7 E-ST-50-04: Telecommand protocols, synchronization and channel coding

4.7.1 Introduction

Currently there is one E-50 standard for telecommand space data links: E-ST-50-04, *Space data links - Telecommand protocols, synchronization and channel coding*. Figure 4-8 shows the relationship to the equivalent CCSDS Recommended Standards which are:

CCSDS 231.0-B, *TC Synchronization and Channel Coding*

CCSDS 232.0-B, *TC Space Data Link Protocol*

CCSDS 232.1-B, *Communications Operation Procedure-1*

In the old ESA PSS series, there was a standard giving the telecommand decoder specification. There is currently no equivalent ECSS standard. Section 4.7.7 discusses some of the issues for telecommand decoders.

The term Packet Telecommand is sometimes applied to the protocol stack consisting of the following set of standards:

- CCSDS 133.0-B, *Space Packet Protocol*
- ECSS-E-ST-50-04 (or the equivalent CCSDS standards)

The term Packet Telecommand was used in earlier CCSDS and PSS standards but it is not used in recent standards.

4.7.2 Purpose and usage

The ECSS-E-ST-50-04 standard is designed to meet the requirements of space missions for reliable transfer of telecommands over space communications links.

The standard specifies data structures and protocols for a telecommand space data link and also specifies the physical layer operation procedure. Usually, the source of data on a telecommand space data link is located on the ground and the receiver is located in space. However, the Standard may also be used for space-to-space telecommand data links.

ECSS-E-ST-50-04 relates primarily to the Data Link Layer discussed in section 3.3.2 of this handbook and it also specifies some procedures in the Physical Layer discussed in section 3.3.1.

The functions and protocols addressed in the standard consist of:

- segmentation sublayer;
- transfer sublayer;
- synchronization and channel coding sublayer;
- procedures in the physical layer at the sending end.

Figure 4-8 shows how the layers and sublayers in the standard relate to the Data Link Layer and Physical Layer. The figure also shows the clause within ECSS-E-ST-50-04 and the reference number of the equivalent CCSDS Recommended Standards.

Layer / Sublayer in ECSS-E-50-04	Clause		CCSDS Recommended Standard
Segmentation sublayer	5	Data Link Protocol Sublayer of Data Link Layer	CCSDS 232.0-B
Transfer sublayer	6		
COP-1 protocol	7		CCSDS 232.1-B
Synchronization and channel coding sublayer	8	Synchronization and Channel Coding Sublayer of Data Link Layer	CCSDS 231.0-B
Physical layer	9	Physical layer	

Figure 4-8: Layers and sublayers in ECSS-E-ST-50-04

4.7.3 Description

4.7.3.1 Overview

Figure 4-9 shows the relationship of the telecommand layers and sublayers and the data structures passed between the peer entities.

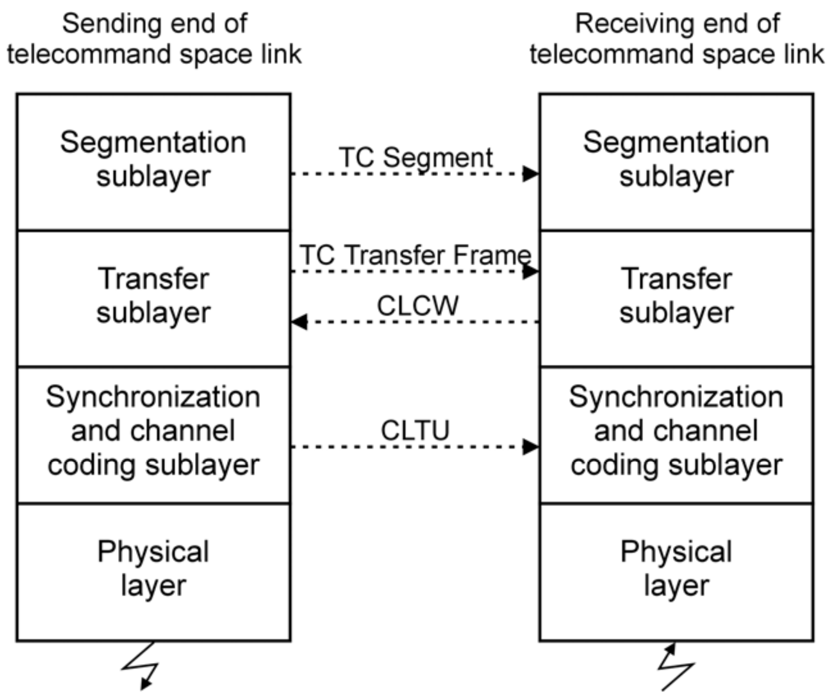

Figure 4-9: Telecommand layers, sublayers and data units

4.7.3.2 Multiplexing for sharing transmission resources

The standard provides two levels of multiplexing, so that multiple users can share access to the space data channel:

- The segmentation sublayer has a MAP multiplexing function, where each virtual channel can support up to 64 multiplexer access points (MAPs). Each MAP is identified by the MAP Identifier of the TC Segment.
- The transfer sublayer has a virtual channel multiplexing function. Each virtual channel is identified by the Virtual Channel Identifier and Spacecraft Identifier fields of the TC Transfer Frame. On a typical physical channel, all the frames have the same Spacecraft Identifier and in this case the physical channel has up to 64 virtual channels.

4.7.3.3 Segmentation sublayer overview

The segmentation sublayer accepts variable-length octet-aligned data units from the next higher layer or sublayer at the sending end and delivers them to the next higher layer or sublayer at the receiving end. Within the segmentation sublayer they are referred to as user data units. In a typical application, the user data units are packets, but the segmentation sublayer can also handle other data units.

The segmentation sublayer provides three functions:

- The blocking function combines multiple user data units so that they are carried in a single TC Transfer Frame. The user data units for the blocking function are limited to packets with a recognised CCSDS version number in the first three bits.

- The segmentation function breaks long user data units into multiple shorter portions at the sending end and reassembles them at the receiving end.
- The multiplexing function multiplexes user data units allocated to different multiplexer access points (MAPs) together so that they share one virtual channel of the transfer sublayer.

The segmentation function and the MAP multiplexing function both use fields in the TC Segment data structure.

Each of the three functions provided by the segmentation sublayer is relatively simple, but the many options and combinations result in a complex sublayer.

4.7.3.4 Segmentation sublayer at the sending end

In a typical scenario, all the data units generated by the segmentation sublayer at the sending end are TC Segments. However, the standard also supports the generation of data units that are not TC Segments. The two kinds of generated data unit cannot be mixed within a virtual channel. Within a physical channel, there can be some virtual channels that carry TC Segments and some that do not carry TC Segments.

For a virtual channel carrying TC Segments:

- Within that virtual channel, the segmentation sublayer can support MAP multiplexing.
- Within that virtual channel, the segmentation sublayer can optionally support the blocking function, or the segmentation function, or both functions, on a MAP by MAP basis. The functions are performed independently for each MAP.
- For any MAP that does not have the segmentation function enabled, all the user data units for that MAP must be short enough to be carried in a single TC Segment.
- For any MAP with the blocking function enabled, all the user data units for that MAP must be packets.
- All the data units generated by the segmentation sublayer for that virtual channel are TC Segments.

For a virtual channel not carrying TC Segments:

- Within that virtual channel, the segmentation sublayer can optionally support the blocking function. The function is performed independently for the virtual channel. If blocking is enabled, all the user data units for that virtual channel must be packets.
- All the user data units for the virtual channel must be short enough to be carried in a single TC Transfer Frame.
- None of the data units generated by the segmentation sublayer for that virtual channel are TC Segments.

The data units generated by the segmentation sublayer have variable length. They are passed to the next sublayer below, which, in a typical scenario, is the transfer sublayer. In this case, the data unit from the segmentation sublayer occupies the Transfer Frame Data Field of a TC Transfer Frame. The maximum length of the data units is constrained by the maximum length of the TC Transfer Frame.

The objective of the ECSS-E-ST-50-04 standard is the reliable transfer of data units containing telecommands. This is not limited to the safe delivery of the data units but also includes confirmation to the sending end that the data units have been delivered. The sequence-controlled service of the transfer sublayer uses acknowledgements to obtain positive confirmation that data units have arrived at the transfer sublayer at the receiving end. The service generates transfer notification signals with confirm responses. The segmentation sublayer at the sending end takes the information obtained from the confirm responses and makes it available to the next higher layer or sublayer.

4.7.3.5 Segmentation sublayer at the receiving end

If the segmentation function is used at the sending end, then a reassembly function is used at the receiving end. It reassembles the data from a sequence of TC Segments to recreate the original user data unit. If a failure occurs during reassembly, the actions are undefined in the general case.

Failures during reassembly can be caused by the loss of TC Transfer Frames in the transfer sublayer or below. The risk of frame loss can be greatly reduced by using the sequence-controlled service of the transfer sublayer.

The standard also defines an optional Packet Assembly Controller (PAC) to perform the reassembly. It specifies additional actions for the reassembly function, including the behaviour in the event of a reassembly failure. (Despite the word "packet" in its name, the PAC is not limited to handling packets. The name is from earlier standards.) Each instance of the PAC uses two MAP identifiers and so reduces the number of MAPs available for data transfer.

4.7.3.6 Transfer sublayer

The transfer sublayer uses the variable-length TC Transfer Frames to reliably transport data through the telecommand channel to the receiving spacecraft. The transfer sublayer contains procedures for generating and processing TC Transfer Frames. The procedures include a multiplexing function so that multiple virtual channels can share a physical channel.

The operation of the transfer sublayer protocol uses status information which is passed from the receiving end to the sending end of the sublayer. The information is contained in a communications link control word (CLCW). The CLCW can be transmitted in the Operational Control Filed of a telemetry transfer frame as described in section 4.6.

The transfer sublayer includes the communications operation procedure (COP), which operates the transmission protocol of the sublayer. It contains the retransmission procedures required to reliably deliver TC Transfer Frames to the spacecraft. A COP consists of a pair of synchronized procedures which execute within one virtual channel at the sending and receiving ends of the sublayer. The standard defines one COP, called COP-1. The sending end procedure for COP-1 is called the frame operation procedure (FOP-1) and the receiving end procedure is called the frame acceptance and reporting mechanism (FARM-1).

COP-1 supports two services types:

- The sequence-controlled service ensures with a high probability of success that data is delivered in order, without duplication and with no omissions. The sequence-controlled service is based on an automatic repeat request (ARQ) procedure of the "go-back-n" type. The service uses acknowledgements, contained in CLCWs, to obtain positive confirmation that data units have arrived at the transfer sublayer at the receiving end.
- The expedited service delivers data in order and without duplication but with possible omissions. The service does not provide any confirmation of delivery.

COP-1 includes a number of controls, such as timers and windows. The values of the controls are implementation parameters or operational parameters, which can be used to fine-tune the performance of the protocol.

Figure 4-10 shows the format of the TC Transfer Frame.

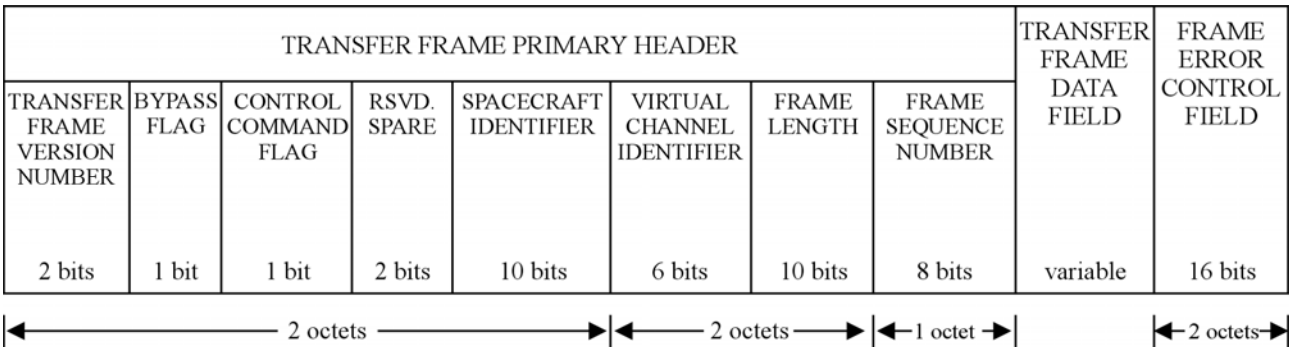

Figure 4-10: TC Transfer Frame

The fields of the TC Transfer Frame include:

- Transfer Frame Version Number, which is always set to 00;
- The Bypass Flag and Control Command Flag are interpreted as in Table 4-2.

Table 4-2: Interpretation of Bypass Flag and Control Command Flag

Bypass Flag	Control Command Flag	Interpretation
0	0	AD frame, carrying data in the sequence-controlled service
0	1	Reserved for future application
1	0	BD frame, carrying data in the expedited service
1	1	BC frame, carrying a control command

- Spacecraft Identifier (the Secretariat of the CCSDS assigns Spacecraft Identifiers according to the procedures in CCSDS 320.0-B – see section 4.15);
- Virtual Channel Identifier;
- Frame Length - unlike the TM Transfer Frame, the frames are variable in length, so a frame length field is necessary;
- Frame Sequence Number, which is used for the sequence-controlled service of COP-1;
- Transfer Frame Data Field, which carries a frame data unit (FDU) received from the sublayer above (AD or BD frame), or a COP-1 control command (BC frame);
- Frame Error Control Field uses a cyclic redundancy code (CRC) for detecting errors but, unlike the TM Transfer Frame, the field is mandatory.

The data units passed down from the transfer sublayer at the sending end are TC Transfer Frames but, at the receiving end, the data units received from below are candidate TC Transfer Frames, which can include fill data or can be truncated. The receiving end of the transfer layer therefore includes the frame delimiting and fill removal procedure, which validates the length of a candidate frame and removes any fill data. The procedure uses the Frame Length field in the transfer frame header.

4.7.3.7 Synchronization and channel coding sublayer

The synchronization and channel coding sublayer establishes an error controlled data channel through which data can be transferred.

At the sending end, the data received from the next higher sublayer are encoded with a Bose-Chaudhuri-Hocquenghem (BCH) block code to reduce the effects on the data of noise in the physical channel.

The communications link transmission unit (CLTU) is the data structure which carries the input data from the synchronization and channel coding sublayer at the sending end to the same sublayer at the receiving end. The CLTU contains data as a contiguous series of encoded BCH code blocks. The CLTU provides synchronization for the code blocks and it delimits the beginning of the TC Transfer Frame.

The input data for the BCH codeblocks of a CLTU consist of a single TC Transfer Frame. (However, there is an exception for the case that an extra layer, such as a security layer, comes between the transfer sublayer and the synchronization and channel coding sublayer.)

The synchronization and channel coding sublayer capabilities include the resolution of data ambiguity which can result from certain channel modulation schemes. The synchronization and channel coding sublayer also includes procedures for data randomizing.

For the receiving end, the specification of the sublayer includes a description of the logic for extracting CLTUs from the incoming bit stream. The logic depends on receipt of information from the physical layer concerning the state of the channel. The logic needs to know if the channel is active (bit modulation is detected and bit lock is achieved) or inactive (otherwise).

4.7.3.8 Physical layer

The physical layer provides the radio frequency data path and its associated physical layer operation procedure (PLOP). The standard data structures within the layer are the acquisition sequence, the CLTU and the idle sequence. They are used to provide synchronization of the symbol stream.

The PLOP consists of a sequence of carrier modulation modes which provide for acquisition and synchronization of the telecommand channel. Two procedures, PLOP-2 and PLOP-1, are defined but the standard specifies PLOP-2 should be used in preference to PLOP-1.

4.7.3.9 Annexes

The standard contains annexes addressing:

- implementation of the CRC for the Frame Error Control Field;
- changes from the preceding ESA PSS standard;
- differences from CCSDS recommendations;
- performance issues relating to the synchronization and channel coding sublayer and the PLOP-2 procedure of the physical layer;
- mission configuration parameters, showing the options and values that can be taken by the parameters specified in ECSS-E-ST-50-04.

4.7.4 Differences from CCSDS recommendations

4.7.4.1 Introduction

This section discusses the differences between ECSS-E-ST-50-04 and the equivalent CCSDS Recommended Standards: CCSDS 231.0-B-1, CCSDS 232.0-B-1 and CCSDS 232.1-B-1.

4.7.4.2 Difference 1

4.7.4.2.1 Description

The CCSDS recommendations assume that the sublayer below the segmentation sublayer is the transfer sublayer. ECSS-E-ST-50-04 makes some assumptions about the sublayer below but does not assume that it is the transfer sublayer. The CCSDS recommendations similarly assume that the next higher sublayer to the transfer sublayer is the segmentation sublayer.

4.7.4.2.2 Rationale

Forward compatibility. The general intent is that a future change in one (sub)layer should not require modification of the normative or informative text for the unchanged adjacent (sub)layers. It also eases the insertion of additional layers into the system, such as security layers. An expected benefit is the use (and re-use) of a stable reference and possible future equipment parts in various application and configuration contexts.

4.7.4.2.3 Potential impact

The difference is only in the wording and therefore has no direct impact on implementations or missions.

4.7.4.3 Difference 2

4.7.4.3.1 Description

ECSS-E-ST-50-04 includes the optional Packet Assembly Controller in the segmentation sublayer. The CCSDS recommendations do not include the Packet Assembly Controller. As a result, the minimum length of a TC Segment in ECSS-E-ST-50-04 is one octet and in the CCSDS recommendations the minimum length of a segment is two octets.

4.7.4.3.2 Rationale

Backward compatibility. The Packet Assembly Controller (PAC) is defined in the preceding ESA PSS standard. A PAC control segment has a length of one octet and consists of a MAP reset command for the PAC to leave lockout state following a reassembly failure.

4.7.4.3.3 Potential impact

- The use of the PAC is optional. ECSS-E-ST-50-04 also includes the option to use the segmentation function without the PAC.
- Currently available hardware in Europe uses the PAC for the segmentation function. However, European missions generally do not use the segmentation function.
- The use of the one-octet PAC control segment could impact interoperability (e.g. use of SLE-FSP services).

4.7.4.4 Difference 3

4.7.4.4.1 Description

In the CCSDS recommendations, multiple Transfer Frames can be put in a single CLTU (see clause 4.2.1 of CCSDS 231.0-B-1). In ECSS-E-ST-50-04, a CLTU contains exactly one frame.

4.7.4.4.2 Rationale

- Reliability. For TC Transfer Frames of similar lengths, placing multiple frames in a CLTU results in reduced performance. The reduced performance is caused by the increase in the number of BCH codeblocks per CLTU.
- Backward compatibility. The handling of multiple TC Transfer Frames in a CLTU is not supported by the frame delimiting and fill removal procedure in currently available hardware in Europe. (The procedure, in the transfer sublayer, processes the data decoded from an incoming CLTU.)

4.7.4.4.3 Potential impact

When using PLOP-2, there is slightly greater bandwidth overhead with one-frame-per-CLTU compared to multiple-frames-per-CLTU. The improved error correction performance more than compensates for the slight increase. If PLOP-1 is used, the increase in bandwidth is larger.

4.7.4.5 Difference 4

4.7.4.5.1 Description

In ECSS-E-ST-50-04, all TC Transfer Frames include the Frame Error Control Field. In the CCSDS recommendations this field is optional. As a result, the minimum length of a TC Transfer Frame in this Standard is eight octets and in the CCSDS recommendations the minimum length of a TC Transfer Frame is six octets.

4.7.4.5.2 Rationale

- Reliability. ECSS-E-ST-50-04 requires the use of the error-correction mode for the BCH decoding. When used in error-correction mode, the error detection capability of the BCH code is not sufficient. The use of the CRC in the Frame Error Control Field is therefore needed to limit the probability of undetected frame error at the receiving end.
- Backward compatibility. Frame validation procedure, which performs the CRC error syndrome check, is systematically applied (not optional) in available hardware in Europe.

4.7.4.5.3 Potential impact

The use of the Frame Error Control Field with error-correction mode for the BCH decoding is fully compatible with CCSDS recommendations. Therefore, no potential impact has been identified.

4.7.4.6 Difference 5

4.7.4.6.1 Description

In ECSS-E-ST-50-04, the receiving-end channel always accepts a CLTU Start Sequence with a one-bit error. In the CCSDS recommendations this behaviour is optional.

4.7.4.6.2 Rationale

Better link performance because of reduced probability of CLTU rejection.

4.7.4.6.3 Potential impact

The acceptance of a CLTU Start Sequence with a one-bit error, together with error-correction mode for the BCH decoding, is fully compatible with CCSDS recommendations. Therefore, no potential impact has been identified.

4.7.4.7 Difference 6

4.7.4.7.1 Description

In ECSS-E-ST-50-04, the randomization in the synchronization and channel coding sublayer is always used. In the CCSDS recommendations randomization is optional.

4.7.4.7.2 Rationale

Reliability. Experience with earlier missions has demonstrated the problems that can arise from the failure to use randomization.

For example, problems can be caused by frames that consist of all zeroes. When such a frame is encoded with the BCH code, it produces a CLTU where each codeblock has 56 bits set to zero, seven bits set to 1 and a final bit set to zero.. Several codeblocks like this appear as a kind of square wave and are difficult for transponders to handle successfully, such that even if frame synchronization is likely to be maintained, the bit synchronization can be lost since there can be several codeblocks with no individual bit transition.

4.7.4.7.3 Potential impact

Some currently-available on-board hardware in Europe does not support randomization.

4.7.4.8 Difference 7

4.7.4.8.1 Description

In ECSS-E-ST-50-04, the BCH codeblocks of a CLTU are always decoded in the error-correcting mode, known as single error correction (SEC). In the CCSDS recommendations, the codeblocks can also be decoded in the error-detecting mode, known as triple error detection (TED).

4.7.4.8.2 Rationale

Better link performance because of reduced probability of CLTU rejection. (See Annex D of ECSS-E-ST-50-04 for the performance data.)

4.7.4.8.3 Potential impact

The use of error-correction mode for the BCH decoding as specified in ECSS-E-ST-50-04 is fully compatible with CCSDS recommendations. Therefore, no potential impact has been identified.

4.7.4.9 Difference 8

4.7.4.9.1 Description

In PLOP-2 in ECSS-E-ST-50-04, the idle sequence (minimum 8 bits) between CLTUs is always used. In PLOP-2 in the CCSDS recommendations, the idle sequence between CLTUs is optional.

4.7.4.9.2 Rationale

Reliability. Use of an idle sequence of at least 8 bits avoids synchronization lockout. Synchronization lockout can cause the loss of the following frame. (See clause 6.5.2 of CCSDS 231.0-B-1 for a description.)

4.7.4.9.3 Potential impact

This use of the idle sequence is fully compatible with CCSDS recommendations. It is also compatible for interoperability (e.g. use of Forward CLTU service). Therefore, no potential impact has been identified.

4.7.4.10 Difference 9

4.7.4.10.1 Description

In ECSS-E-ST-50-04, the acquisition sequence always has a length of at least 128 bits. The CCSDS recommendations have a preferred minimum length of 128 bits for the acquisition sequence.

4.7.4.10.2 Rationale

The length of the acquisition sequence is adapted to the performance of the transponder.

4.7.4.10.3 Potential impact

This minimum length of 128 bits is fully compatible with CCSDS recommendations. Therefore, no potential impact has been identified.

4.7.4.11 Difference 10

4.7.4.11.1 Description

In ECSS-E-ST-50-04, the idle sequence always has a length of at least 8 bits. The CCSDS recommendations have no minimum length for the idle sequence.

4.7.4.11.2 Rationale

Reliability. See difference 8 above.

4.7.4.11.3 Potential impact

No potential impact has been identified. See difference 8 above.

4.7.4.12 Difference 11

4.7.4.12.1 Description

In the CCSDS recommendations, the use of the No Bit Lock Flag in a CLCW is optional and mission-specific. In ECSS-E-ST-50-04, the use of the flag is defined and it is always used.

4.7.4.12.2 Rationale

Improved Observability for the operation of the TC space link. The information carried by the flag is used in the operation of the carrier modulation modes of the PLOP procedure. (See clause 9.3.2.3 of the standard.)

4.7.4.12.3 Potential impact

No potential impact has been identified.

4.7.4.13 Difference 12

4.7.4.13.1 Description

In the CCSDS recommendations, some names of data structures and other entities are different, as shown in Table 4-3.

Table 4-3: Name correspondence between CCSDS and ECSS-E-ST-50-04

Name in CCSDS recommendations	Name in ECSS-E-ST-50-04
Segment	TC Segment
Service Data Unit	user data unit
User Data	Segment Data Field

4.7.4.13.2 Rationale

- TC Segment: the name has been kept for backward compatibility.
- User data unit: ECSS-E-ST-50-04 does not define a service so the term “Service Data Unit” is unsuitable.
- Segment Data Field: clearer name for a field which is part of a TC Segment.

4.7.4.13.3 Potential impact

The difference is only in the names and therefore has no direct impact on implementations or missions.

4.7.4.14 Difference 13

4.7.4.14.1 Description

In ECSS-E-ST-50-04, an FDU arrived indication from FARM-1 always has a parameter for the FDU aborted indication. In CCSDS 232.1-B-1, this parameter is optional.

4.7.4.14.2 Rationale

At the receiving end, FARM-1 uses an FDU Arrived Indication to signal the arrival of an FDU to the next higher sublayer. If FARM-1 has a single back-end buffer per virtual channel, a newly arrived frame can overwrite, and so erase, the previous FDU. The FDU aborted indication is used to inform the next higher sublayer that the overwriting has occurred. (See clause 7.5 of the standard for details.) Some currently-available on-board hardware in Europe has FARM-1 with a single back-end buffer.

4.7.4.14.3 Potential impact

When using FARM-1 with a single back-end buffer per virtual channel, special operating considerations apply, which are specified in ECSS-E-ST-50-04. More generally, users should be aware of any buffering limitations in FARM-1 for a virtual channel and should understand the relationship between uplink and on-board flow control. On-board systems should be designed to eliminate delays in copying or otherwise processing the FDUs delivered by FARM-1.

4.7.4.15 Difference 14

4.7.4.15.1 Description

ECSS-E-ST-50-04 includes procedures for the behaviour of a FARM-1 implementation which provides only one back-end buffer for a virtual channel. CCSDS 232.1-B-1 does not consider this special case.

4.7.4.15.2 Rationale

Some currently-available on-board hardware in Europe has FARM-1 with a single back-end buffer, which can result in a newly-arrived FDU overwriting the previous FDU. Therefore the special handling for this case needs to be addressed in ECSS-E-ST-50-04.

4.7.4.15.3 Potential impact

See Difference 13 above.

4.7.5 Options

Annex E of the standard lists the mission configuration parameters.

The annex shows the options and values that can be taken by the parameters as specified in the standard. It is the responsibility of mission designers to check the availability of support for the options and values selected for the mission.

For the physical channel, the options and parameters include the length of the acquisition sequence, the physical layer operation procedure (PLOP-2 or PLOP-1), the maximum length of the TC Transfer Frame and the virtual channel multiplexing algorithm.

The parameters for a virtual channel include the values for the Spacecraft Identifier and Virtual Channel Identifier, and the values of the COP-1 timer and other controls. The use of TC Segments is an option for a virtual channel:

- If the virtual channel uses TC Segments, then there are parameters for the MAP multiplexing algorithm. For each MAP, there are parameters for the optional use of the blocking and segmentation functions.
- If the virtual channel does not use TC Segments, then there are parameters for the optional use of the blocking function.

For a MAP or virtual channel with the blocking function, there are additional parameters for the packet types used.

The proforma of the Protocol Implementation Conformance Statement (PICS) can also be used as a tool for considering the options of a standard. For ECSS-E-ST-50-04, there is a draft PICS proforma for a sending end implementation (see Annex E of this handbook) and for a receiving end implementation (see Annex F of this handbook). For an overview of PICS, see section 5.8.

4.7.6 Recommended practice

4.7.6.1 Segmentation function of the segmentation sublayer

It is recommended practice to use the sequence-controlled service for any MAPs which use the segmentation function. The Sequence Flags of the Segment Header provide information about the sequential position of the segment, to enable the receiving end to reassemble the user data units (e.g. packets) from a stream of segments. However, the flags do not provide a sequence count, so the reassembly process depends on the stream of segments being delivered in sequence and without omissions. The sequence-controlled service of the transfer sublayer can provide delivery in sequence and without omissions.

4.7.6.2 Virtual channel identifiers

In a typical case, a spacecraft uses multiple telecommand virtual channels. It is recommended practice to assign the telecommand virtual channel identifier values so that the digital distance (i.e. hamming distance) between the values is maximized. This reduces the risk that an error introduced in a CLCW during on-board data handling changes the virtual channel identifier value into the value of another virtual channel used by the spacecraft.

A CLCW with a virtual channel identifier value consisting of all zeroes or all ones can be generated when a decoder fails or is powered off. Therefore it is recommended practice to avoid virtual channel identifier values consisting of all zeroes or all ones.

4.7.6.3 CLCW fields

ECSS-E-ST-50-04 includes requirements for the setting of the No RF Available Flag and the No Bit Lock Flag in a CLCW. The values of the flags report the state of the physical channel as a whole and not just the state of a single decoder.

The Virtual Channel Identification field of a CLCW carries the full 6-bit value of the virtual channel identifier.

4.7.6.4 COP-1 sequence-controlled service and expedited service

It is recommended that near-Earth missions should use the COP-1 sequence-controlled service. It is recommended to restrict the use of the expedited service to exceptional operational circumstances, such as spacecraft recovery operations.

4.7.6.5 COP-1 performance for deep space missions

The retransmission mechanism of the COP-1 sequence-controlled service works well for near-Earth missions with a short light trip delay but it is less efficient for deep space missions. An additional, optional, retransmission mechanism for use by deep space missions is being studied.

4.7.6.6 Physical layer operation procedure (PLOP)

It is recommended practice to use PLOP-2 instead of PLOP-1. PLOP-2 supports a higher data throughput on the physical telecommand channel.

4.7.7 Telecommand decoder specification

There is currently no ECSS standard or handbook for the telecommand decoder specification. The old PSS-04-151 standard, *Telecommand decoder specification*, from the ESA PSS series has been withdrawn but it contained important information. ECSS is considering the production of a telecommand decoder handbook.

The telemetry and telecommand packet utilization standard, ECSS-E-ST-70-41, has an annex that defines the use of the Command Pulse Distribution Unit (CPDU) and its interactions with the telecommand decoders specified by the old PSS standard.

4.8 CCSDS 732.0-B: AOS Space Data Link Protocol

4.8.1 Purpose and usage

4.8.1.1 Background

The Advanced Orbiting Systems (AOS) Recommendation emerged in the early 1990s in response to requirements for the International Space Station. Typical Advanced Orbiting Systems were deemed to include manned and man-tended space stations, unmanned space platforms, free-flying spacecraft and advanced space transportation systems. Many of these systems needed services to concurrently transmit multiple classes of digital data (including audio and video) through space/ground, ground/space and space/space data channels at relatively high combined data rates. They also needed the capability to interface with, and exploit, the rich service environment of the, then current, worldwide Open Systems Interconnection (OSI).

The principal difference between the conventional and AOS recommendations is the much wider repertoire of services provided for Advanced Orbiting Systems. The conventional concepts of "telemetry" and "telecommand" become blurred. Instead, the forward and return space links were to become the vehicles for extensive two-way interchange of many different classes of digital message traffic between ground and space.

Consequently, symmetric services and protocols were provided for AOS so that video, audio, high-rate telemetry, low-rate transaction data, etc., can be exchanged through the space link in either direction. To handle different classes of data that share a single link, various transmission schemes (e.g., asynchronous, synchronous, isochronous) were provided, as were different user data formatting protocols (e.g. bitstreams, octet blocks and packetized data) and different grades of error control.

Strong efforts were made to ensure that the AOS Recommendation was downward-compatible with the suite of conventional CCSDS protocols. Maintaining this level of compatibility was essential in order to encourage the widespread deployment of a standard data handling service infrastructure within the ground systems.

For the first time, capabilities were included to run commercially derived ground network protocols into the space segment. The AOS Recommendation therefore provided a space-adapted analog of the terrestrial concept of an "integrated services digital network". The Recommendation defined the CCSDS Principal Network which included a ground network and an on-board network, in addition to the space links. A Path Service and an Internet Service to provide end-to-end data transfer across the CCSDS Principal Network were also defined.

The original AOS Recommendations were specified in CCSDS-701.0-B. The document is now obsolete and in 2003 it was replaced by the current AOS Recommended Standard, CCSDS-732.0-B, which is limited to the transfer of data across a single space link. The remainder of the discussion of AOS in this handbook concerns the current specification in CCSDS-732.0-B.

4.8.1.2 Current specification of the AOS Space Data Link Protocol

4.8.1.2.1 Overview of AOS Transfer Frames

The CCSDS AOS Space Data Link Protocol uses AOS Transfer Frames, which are fixed-length transfer frames similar to the TM Transfer Frames. TM Transfer Frames are discussed in section 4.6.2 and are defined in ECSS-E-ST-50-03, *Telemetry transfer frame protocol*, and in the equivalent Recommended Standard CCSDS 132.0-B, *TM Space Data Link Protocol*.

Figure 4-11 shows the format of the AOS Transfer Frame and Figure 4-12 shows the fields in its Transfer Frame Primary Header. These can be compared with Figure 4-6 and Figure 4-7 which show the format of the TM Transfer Frame and its primary header. Table 4-4 lists some of the differences between the two frame types.

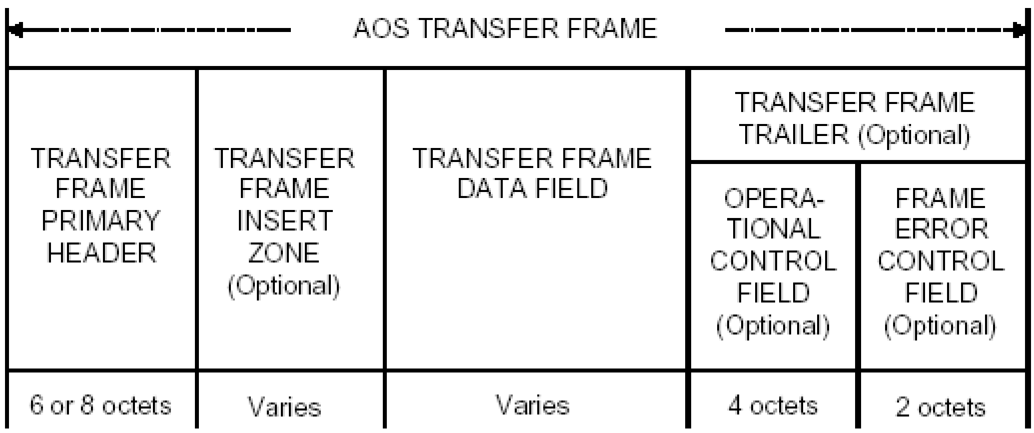

Figure 4-11: AOS Transfer Frame

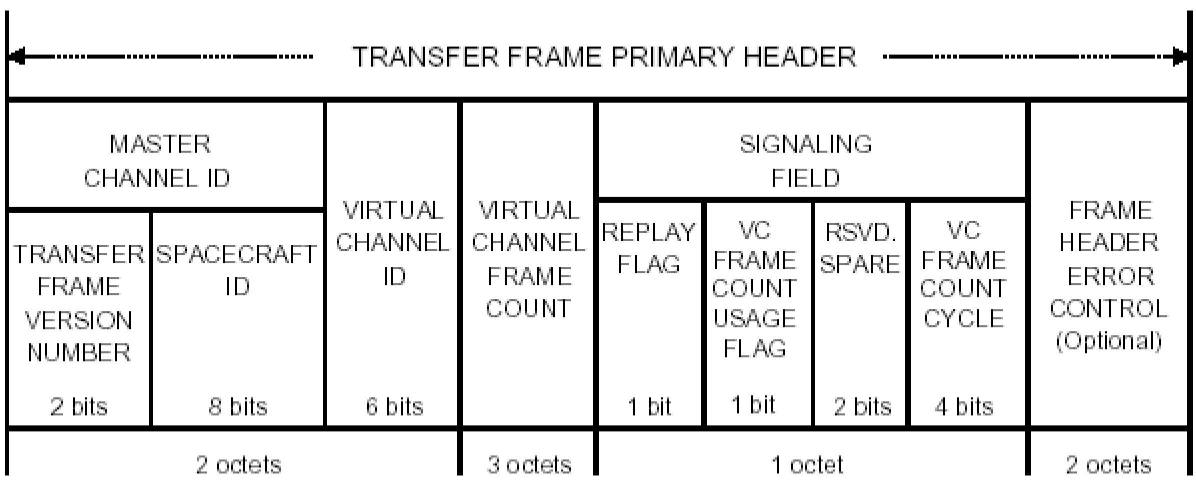

Figure 4-12: AOS Transfer Frame Primary Header

Table 4-4: Some differences between AOS and TM transfer frames

Field	AOS Transfer Frame	TM Transfer Frame
Transfer Frame Version Number	binary '01' Version-2 Transfer Frame	binary '00' Version-1 Transfer Frame
Spacecraft Identifier	8 bits	10 bits
Virtual Channel Identifier	6 bits 64 Virtual Channels	3 bits 8 Virtual Channels
Master Channel Frame Count	(none)	8-bits
Virtual Channel Frame Count	24-bits (plus optional 4 bits in VC Frame Count Cycle)	8-bits (plus optional extended count in Transfer Frame Secondary Header)
pointer for start of first packet	in the data field	field in the primary header
field for auxiliary data	Insert Zone	Transfer Frame Secondary Header for more limited service
error control for frame header	optional 16-bit Frame Header Error Control provides Reed-Solomon error correction for header values	-

4.8.1.2.2 Services of the AOS Space Data Link Protocol

CCSDS 732.0-B, *AOS Space Data Link Protocol* and CCSDS 132.0-B, *TM Space Data Link Protocol*, both include formal definitions of services for transporting data in the various data fields of the transfer frames.

The major difference in the services is the Insert Service of the AOS protocol, which uses the optional Insert Zone of an AOS Transfer Frame. The service provides the transfer of private, fixed-length, octet-aligned service data units across a space link. If the Insert Service is defined for a physical channel, then every Transfer Frame of the physical channel has an Insert Zone carrying data for the Insert Service. By comparison, the TM protocol provides services using the optional Transfer Frame Secondary Header of a TM Transfer Frame. The services are more flexible because they are not tied to the physical channel but they have a more restricted maximum length.

Most of the services for using the Transfer Frame Data Field and the Operational Control Field are the same for both protocols, with minor differences in parameters and in the way some types of data are carried. For example, the Packet Service in each case supports the same types of packets, but the pointer for the first packet within a frame is carried in a slightly different way. Both protocols have a service for carrying a 32-bit field, such as a telecommand CLCW, in the Operational Control Field.

The AOS protocol defines an additional service for using its Transfer Frame Data Field, called the Bitstream Service, which provides the transfer of a serial string of bits across a space link. The service requires exclusive use of a Virtual Channel.

Note that ECSS-E-ST-50-03 (see section 4.6) is fully compatible with CCSDS 132.0-B for the way that data is carried in a TM Transfer Frame but it does not include the formal definitions of the services.

4.8.1.2.3 Use of Synchronization and Channel Coding

CCSDS 732.0-B, *AOS Space Data Link Protocol* and CCSDS 132.0-B, *TM Space Data Link Protocol*, both specify the use of CCSDS 131.0-B, *TM Synchronization and Channel Coding*, for the Synchronization and Channel Coding Sublayer. The ECSS equivalent to CCSDS 131.0-B is ECSS-E-ST-50-01, discussed in section 4.5.

4.8.2 Options

The Recommended Standard CCSDS 732.0-B includes a section which lists the managed parameters for a mission. Most of the parameters are fixed for a mission phase, and, in practice, are often fixed for the duration of the mission. The managed parameters section shows the options and values that can be taken by the parameters specified in the Recommended Standard.

For the physical channel, the options and parameters include the length of the AOS Transfer Frame, the presence and length of the Insert Zone, the presence of the Frame Error Control Field, and the Spacecraft Identifier values used on the physical channel.

There are further options and parameters for a master channel and for a virtual channel, including the use of the Transfer Frame Data Field and the Operational Control Field. For a virtual channel which supports the Packet Service, there are additional parameters for the maximum packet length and for the packet types used on the virtual channel.

4.8.3 Recommended practice

4.8.3.1 AOS and TM on return links

Although not formally addressed in the Recommended Standard, it is possible to run “hybrid” configurations where the conventional and AOS protocols are mixed. For example, the telecommand protocols specified in ECSS-E-ST-50-04 are used on the forward link with the AOS protocol on the return link. AOS includes the Operational Control Field which can be used to carry the telecommand CLCWs.

The actual adoption of the AOS protocols has been for high rate downlinks in e.g. Envisat and International Space Station. Some mission scenarios have two physical downlinks, for example:

- a high rate downlink using AOS Transfer Frames for payload data
- a downlink using TM Transfer Frames for spacecraft housekeeping data.

Note that ECSS-E-ST-50-03 forbids the use of TM Transfer Frames and AOS Transfer Frames together on the same physical link.

Future missions should monitor ongoing standards development for their high speed downlink requirements.

4.8.3.2 AOS on the forward link

Current practice does not recommend the use of AOS for the forward link.

4.8.3.3 Packets over AOS links

In principle, the CCSDS end-to-end services (Space Packet Protocol, CFDP and Encapsulation Service) can be used over an AOS return link in the same way as for the TM Transfer Frame Protocol specified in ECSS-E-ST-50-03. Implementation support should be confirmed in each case.

Of the AOS protocol services, the Packet Service will support the Space Packet Protocol whereas high rate streaming payloads may find the need to use the Virtual Channel Access (VCA) Service.

4.9 CCSDS Proximity-1 Space Link Protocol

4.9.1 Purpose and usage

The Proximity-1 Space Link Protocol was developed by CCSDS to support missions that require reliable point-to-point communication between two spacecraft. Typical mission scenarios include:

- a base station and a roving vehicle, such as Pathfinder;
- an orbiter and a lander, such as NASA Mars Reconnaissance Orbiter communicating with ESA ExoMars rover or NASA MER rovers.

As the name suggests, Proximity-1 provides reliable data transfer between spacecraft operating in close proximity. These can be spacecraft clusters or the link between orbiting and landed elements on interplanetary missions. The need for such a protocol was, in part, a response to the requirement for mission interoperability between Mars spacecraft.

Proximity-1 is specified in three CCSDS Recommended Standards:

- CCSDS 211.0-B *Proximity-1 Space Link Protocol - Data Link Layer*
- CCSDS 211.1-B *Proximity-1 Space Link Protocol - Physical Layer*
- CCSDS 211.2-B *Proximity-1 Space Link Protocol - Coding and Synchronization Sublayer*

4.9.2 Description

Because of the diverse environments to be expected at Mars, Proximity-1 incorporates a number of unique features.

A key departure from previous CCSDS link protocols is in recognition of the ephemeral nature of Proximity-1 contact periods due to orbiter motion and planetary rotation and of the remoteness, due to delay, of the mission control centre. To maximize the contact opportunity, Proximity-1 incorporates a handshake mechanism which allows connection between the orbiting and landed elements to be established locally without recourse to precise mission scheduling. It also allows for mid-contact renegotiation of link data rates to maximize the data throughput opportunities against a background of dynamically changing link budgets.

Proximity-1 has the ability to operate with transceivers that are frequency agile, allowing for a hailing channel to identify and initiate communications and a separate working channel to carry the data interchange. This feature can be used to reduce interference between landed elements and orbiting elements when multiple elements have overlapping view periods.

Proximity-1 provides for full duplex, half duplex and simplex operation. In full duplex operation there is simultaneous bidirectional flow of data between the mission elements. In half duplex operation the transceivers take turns sending and receiving data. In simplex operations there is communication in one direction only: one mission element transmits and the other receives. In simplex mode there is no feedback, so there can be no confirmation that data units were received without gaps.

Radiometric aspects are of significant importance at Mars and their setting can be controlled by use of Proximity-1 directives. For a given transceiver, the associated details are documented in an interface specification.

Proximity-1 is a symmetrical protocol. The same protocol and options may be used for both the forward and return links of two-way communications, and the link data rates may be symmetrical. This has the advantage that the same equipment can be used in both ends of the communication link.

For a typical Proximity-1 link, however, a number of factors affect the communications:

- The two-way communications data rates may be vastly different; with the return rate often being significantly higher than the forward rate. This rate differential will often be as high as 256 to 1 and can be as high as 2000 to 1.
- There can be variations in the over-flight geometry, signal path and strength during a pass.
- There can be variations in the one way light time when communicating with different mission elements.

To overcome these factors, it is important to tune the configuration and operational parameters associated with a specific communications session.

The key data transport data unit of the Proximity-1 protocol is the Proximity-1 transfer frame. The frame can be of variable length. Before the frame is transmitted, a 32-bit cyclic redundancy check field is appended to the end of the frame. This provides a strong error detection capability so that frames containing errors can be rejected at the receiving end.

The Recommended Standard 211.0-B defines two data transfer services for carrying data in a Proximity-1 frame:

- a service to deliver packets: a packet has a defined Packet Version Number, as specified in CCSDS 135.0-B
- a service to deliver user defined data: the data consist of a sequence of octets in a format unknown to the Proximity-1 protocol.

In addition, there are timing services, which are needed for Proximity-1 operations in order to provide time correlation data among communicating units and time-derived ranging measurements.

The data transfer services have two grades of service:

- Sequence-controlled, which can transfer data in order and without gaps or duplications
- Expedited, which can transfer data in order and without duplications but with possible gaps.

The heart of the sequence-controlled service is a simple, efficient, go-back-n retransmission scheme that uses a small part of the link bandwidth to acknowledge receipt of data. The receiver can explicitly request a retransmission when it detects a gap in the received data, and the transmitter will automatically perform retransmissions if the receiver does not acknowledge receipt within a (configurable) time-out period. The retransmission protocol is a simplified version of the one contained in the COP-1 telecommand protocol.

4.9.3 Options

Proximity-1 seeks to optimise use of orbiter/lander contact periods and has a large array of options for the physical links. These include operating frequencies, length of preambles, supported data rates, use of coding and many more.

For data transfer, the major options within Proximity-1 are the choice of service (packets or user defined data) and the grade of service (sequence-controlled or expedited).

4.9.4 Recommended practice

Adoption of options and choice of supported parameter values are dependent on:

- Mission characteristics with regard to lander/orbiter communications.
- Interoperability with the mission elements of other agencies, which is inevitable given the multi-spacecraft nature of the Proximity-1 links.

For this reason, the adopted practice depends on analysis of the mission characteristics and mission-to-mission negotiation. CCSDS is attempting to mitigate these processes by the production of protocol profiles, the first being for Mars missions and largely encompassing Proximity-1. See also the Informational Report CCSDS 740.0-G.

For intra-agency communications, Table 4-5 identifies the best practice for the major options and protocol parameters with regard to orbiter to lander communications at the UHF band in planetary exploration missions.

Table 4-5: CFDP best practice for options and parameters

Item	Recommendation
Hailing Frequency	UHF Channel 1
Channels Implemented	UHF Channel 1 mandatory Channels 0 and 2 to 7 optional
Polarization	Right hand circular
Data Rates	8kbps mandatory Others optional
Quality of Service	Sequence Controlled mandatory Expedited mandatory
Service	Packet Service
Communications Modes	Full Duplex mandatory Half Duplex mandatory
Carrier_Only_Duration	5s / 1s
Acquisition_Idle_Duration	4096 bits
Tail_Idle_Duration	4096 bits

4.10 CCSDS Packet Protocols

4.10.1 Introduction

CCSDS 133.0-B defines the Space Packet Protocol, which can be used to route higher-layer data through the data system that includes a space link and the onboard and ground subnetworks. The protocol is in the Network Layer discussed in section 3.3.3. Addressing capabilities are provided by the 11-bit Application Process Identifier (APID) carried in the header of a Space Packet. An optional APID Qualifier, which is not part of the Space Packet, can provide additional addressing information.

CCSDS 133.1-B, *Encapsulation Service*, includes the specification of the Encapsulation Packet. The Encapsulation Packet provides an alternative to the Space Packet for the encapsulation of data units such as IPv6 datagrams. Its main addressing capability is the 3-bit Protocol ID field in the header of the Encapsulation Packet.

4.10.2 CCSDS 133.0-B: Space Packet Protocol

4.10.2.1 Purpose and usage

The Space Packet Protocol efficiently transfers space application data of various types and characteristics over a network that involves a space link. The protocol provides a unidirectional data transfer service from a single source user application to one or more destination user applications through one or more subnetworks.

CCSDS 133.0-B includes the specification of the Space Packet, which replaced three earlier packet specifications:

- the Source Packet, from Packet Telemetry
- the TC Packet, from Packet Telecommand
- the Version-1 CCSDS Packet, from AOS

The earlier specifications all had the same Packet Version Number (PVN) and basic data structure, with some differences in the names and meanings of fields. The earlier packet names are still in use in some cases and can be found in standards which have not been updated since the publication of CCSDS 133.0-B.

4.10.2.2 Description

4.10.2.2.1 Space Packet

The protocol data unit of the Space Packet Protocol is the Space Packet. Space Packets are variable in length (or may be fixed at the discretion of the user application) and are transmitted at variable intervals. Aside from a header that identifies the packet, the internal data content of a packet is completely under the control of the user application. Each user application can define the organization and content of packets independently of other user applications and with a minimum of constraints imposed by the transmission mechanisms of the underlying subnetworks.

4.10.2.2.2 Logical Data Path (LDP)

The Space Packet Protocol provides services to transfer space application data from the source user application to the destination user application through the underlying subnetworks. The path from the source user application to the destination user application is called the Logical Data Path (LDP) and the major function performed by the protocol is the routing of the data along the LDP. As the data traverse the subnetworks of the LDP, they are carried by subnetwork-specific mechanisms using protocols provided by the subnetworks. See Figure 3-4 for an example, which shows CFDP as the user application.

An LDP is uniquely identified by a Path Identifier (Path ID) which consists of two parts: the Application Process Identifier (APID) and an optional APID Qualifier. The APID is contained in the Packet Primary Header of the Space Packet. The APID Qualifier does not appear in the Space Packet: if the APID Qualifier is used, then its value is usually carried by one or more protocols of the underlying subnetworks.

If Space Packets are transferred over a space link with one of the Space Data Link Protocols (TM, TC or AOS) then the Master Channel Identifier (MCID) of the Space Data Link Protocol is used as the APID Qualifier. The MCID consists of the Transfer Frame Version Number and the Spacecraft Identifier (SCID).

When a Space Packet is transferred along an LDP, the Space Packet Protocol entities at the source and intermediate system examine the Path ID of the incoming packet and route it through appropriate subnetworks using the mechanisms provided by the subnetworks. The LDP itself is preconfigured and is established by management, including the routing information (i.e. mapping from Path IDs to subnetwork addresses).

4.10.2.2.3 Services

CCSDS 133.0-B defines two services for the Space Packet Protocol:

- The Octet String Service accepts a delimited octet string from the source user. It appends a packet header for onward communication and subsequently removes the header, delivering an octet string to the destination user.
- The Packet Service accepts a Space Packet, pre-formatted by the service user. It delivers the packet to the destination user.

4.10.2.3 Options

A Logical Data Path has a single source and one or more destinations. The option for multiple destinations is used, for example, to send telemetry data from a single onboard application process to multiple applications on the ground. The multiple destination addresses are configured in the LDP, established by management.

The specification of a Space Packet accommodates a length between 7 and 65,542 octets. A mission can choose to set a lower value for the maximum length of its Space Packets.

The header of a Space Packet includes the 1-bit Packet Type field, which is set to 0 for a telemetry packet and to 1 for a telecommand packet. Usually, telemetry packets are associated with a downlink and telecommand packets with an uplink. However, the exact definition is not always obvious, for example on a link between two spacecraft. The meaning of the Packet Type field is established by the project that uses the Space Packet Protocol.

4.10.2.4 Recommended practice

4.10.2.4.1 Packet utilization standard

ECSS-E-ST-70-41, the telemetry and telecommand packet utilization standard (PUS), defines the use of fields in Space Packets: see section 4.12 below.

4.10.2.4.2 Use and allocation of APIDs

For a typical mission where a single spacecraft has direct links to its ground system, the APIDs can be used to identify the associated onboard application process:

- the APID of a telemetry packet identifies the application process that is the source of the packet
- the APID of a telecommand packet identifies the application process that is the destination of the packet

This is similar to the use of the APID in the PUS (see section 4.12 below). The 11-bit APID field is more than adequate for addressing the application processes within a single spacecraft. The spacecraft represents a single naming domain for APIDs and the APID Qualifier need not be used.

APIDs are unique only in a single naming domain. In a more complex mission scenario, there can be multiple naming domains for APIDs, for example, when there are shared ground segments or when orbiters provide relay services to landed elements. An APID naming domain usually corresponds to a spacecraft, or to an element of a constellation of cooperating space vehicles.

An APID Qualifier identifies a naming domain for APIDs. The assignment of APIDs to LDPs within a naming domain is controlled by the space project that owns the naming domain. Each space project needs to establish the APID naming domains to be used in the project.

4.10.2.4.3 Long packets

On a telemetry link, a Space Packet is generally transferred in one or more TM (or AOS) Transfer Frames. As the length of the packet increases, so does the number of frames required to transfer the packet, and the loss of one of the frames because of errors on the link can result in loss of all the data in the packet. The advantages of long packets, such as reduced overhead, should be weighed against any risk of increased data loss.

On a telecommand link, the sequence-controlled service of COP-1 can provide protection against lost frames. However, telecommand links generally have low data rates, leading to a long elapsed time to transfer a long packet, and the data is not available to the onboard application until the whole packet has arrived. Long packets can also result in increased onboard buffering requirements.

4.10.3 CCSDS 133.1-B: Encapsulation Service

4.10.3.1 Purpose and usage

The purpose of the Encapsulation Service defined in CCSDS 133.1-B is the transfer of data units which do not have a Packet Version Number (PVN) authorized by CCSDS.

The service supports the transfer of IPv6 packets and can be used as an alternate transfer mechanism for IPv4 packets. In addition, further network layer protocols identified by CCSDS can be carried as well as purely private protocols.

4.10.3.2 Description

The Encapsulation Service transfers a variable-length, delimited, octet-aligned data unit supplied by the service user: the format and content of the data unit are unknown to the service provider. The service encapsulates the data unit, either in a single Space Packet or in a single Encapsulation Packet, and passes it to the packet transfer service of the underlying Space Data Link Protocol. No more than one data unit is encapsulated into a single packet.

When Space Packets are used for the encapsulation, the maximum length of the data unit is 65,536 octets.

CCSDS 133.1-B includes the specification of the Encapsulation Packet, which has a variable length header. The shorter headers reduce the encapsulation overhead while an Encapsulation Packet with the longest form of header can accommodate a data unit with a length of nearly 4 gigabytes.

The Encapsulation Packet header includes the 3-bit Protocol ID field, which identifies the protocol of the encapsulated data unit. Its values are defined by CCSDS and currently include values for IPv6, IPv4 and CFDP.

4.10.3.3 Options

Options for the Encapsulation Service include the encapsulating packet type (Space Packet or Encapsulation Packet) and the maximum and minimum lengths for the data unit to be encapsulated.

4.10.3.4 Recommended practice

Data units that have an authorized PVN can be directly transferred over a space link by the Space Data Link Protocols (TM, TC, AOS and Proximity-1) and therefore do not generally need to be encapsulated. Authorized PVNs currently include the Space Packet, the Encapsulation Packet and the IPv4 datagram. However, in current implementations of the Space Data Link Protocols, support can be limited to Space Packets.

If encapsulation is needed, then the Encapsulation Service can be considered. Because support for the transfer of Encapsulation Packets can be limited in current implementations, the Space Packet is recommended for the encapsulating packet, unless there is a requirement to provide a capability to transfer very long data units.

If the encapsulating packets are Space Packets, then the Encapsulation Service is similar to the Octet Service defined for the Space Packet Protocol. The length limit for the data unit of 65,536 octets is sufficient for most space applications.

4.11 Internet Protocols

4.11.1 Introduction

Since their inception, CCSDS protocols have had an inherent capability to support network protocols. Although initial recommendations were based on an architecture where applications directly accessed the space link, the packet nature of the space link provides direct support to network protocols in the same way that terrestrial local or wide area networks do.

In order to explicitly support IP, the Space Data Link Protocols (TM, TC, AOS and Proximity-1) have a capability to directly transfer IPv4 datagrams, as packets, across a space link, without any form of packet encapsulation. In the case of IPv6 datagrams, the arrangement of the version number has required the introduction of an encapsulation service to transfer the datagrams, with minimal bandwidth and processing overhead. This is no different to terrestrial solutions where the growth of network layer capabilities has led, for example, to the introduction of Multiprotocol Encapsulation (MPE) to carry packet-oriented protocols.

Using these mechanisms, network layers including IPv4, IPv6 and the Space Packet Protocol can co-exist in the same space links.

Note also that CCSDS application layer protocols are predicated on universal subnetwork or network datagram protocols and, as such, can easily traverse IP-based networks.

4.11.2 RFC 791: Internet Protocol Version 4

4.11.2.1 Purpose and usage

IPv4 is at the core of all Internet communications. Its usage in the ECSS domain is yet to be identified.

The datagrams of IPv4 can be directly transferred by the Space Data Link Protocols over a space link which supports this facility. Otherwise they can be transferred via the Encapsulation Service described in 4.10.3.

4.11.2.2 Description

For a description of IP see any number of extant sources. Details are not supplied here although the following section (4.11.3) does describe the major differences in IPv6.

4.11.2.3 Options

The advanced options which can be adopted in IP are unlikely to be of use in a space implementation. However, there are many options associated with the use of protocols peripheral to IP and these are discussed in section 4.11.2.4.

A parameter which may need to be tuned is the Maximum Transmission Unit (MTU). This may require an analysis of aspects such queuing times, header overhead and link blocking.

4.11.2.4 Recommended practice

Terrestrial implementations of IP are usually accompanied by a number of peripheral protocols. These include:

- Address Resolution Protocol (ARP) – probably not required in a fixed spacecraft configuration
- Dynamic Host Configuration Protocol (DHCP) – again probably not necessary for fixed IP addresses
- Domain Name Service (DNS) – may be useful in making networks with human interaction more user friendly
- Various Routing Protocols (OSPF, RIP2 etc.) – only required in dynamic or congested networks with multiple routes. Probably not required.
- Virtual Router Redundancy Protocol (VRRP) or Hot Standby Redundancy Protocol (HSRP) – can be useful in setting up redundant fixed routes
- User Datagram Protocol (UDP) – Transport protocol adds multiplexing to multiple applications but retains datagram service of IP. Very useable in planned, non-congested environments such as spacecraft. Otherwise useful for non-critical data.
- Transmission Control Protocol (TCP) – Transport protocol with added reliability via connection oriented transmission. Superficially very attractive but beware statefulness of machines linked by TCP connections. Dubious performance over error-prone media (e.g. RF links). It is recommended to investigate Performance Enhancing Proxies (PEPs) such as the Space Communications Protocol Specification – Transport Protocol (SCPS-TP) defined in CCSDS 714.0-B.

4.11.3 RFC 1883: Internet Protocol Version 6

4.11.3.1 Purpose and usage

IPv6 is the successor to IPv4 and addresses the shortcomings in IPv4 which have been identified in v4’s massive deployment and range of applications.

The datagrams of IPv6 can be transferred via the Encapsulation Service described in 4.10.3.

4.11.3.2 Description

IPv6, developed within the last 15 years, is a large-scale re-design and re-engineering of IPv4 that essentially serves the same functions. However, IPv6 offers critical key upgrades and expanded capabilities in the areas of address space, security, network management, and quality of service.

The most visible – and significant – difference between the two protocols involves address space. IPv4 uses 32-bit (4-byte) addresses which limits the address space to just fewer than 4.3 billion possible unique addresses. In the early stages of the Internet development, this was more than adequate for both public and private Internet users. However, many of these addresses are now being reserved for private networks or multicast addresses. Because of this, there is a very real possibility of an IPv4 address shortage in the not-so-distant future.

In contrast, addresses in IPv6 are 128 bits long versus 32 bits in IPv4. And instead of 4.3 billion possible unique addresses, IPv6 increases that number to a theoretical possibility of about 340 trillion unique addresses.

IPv6 allows for the liberation of routers from fragmentation responsibilities and gives the ability to use the "flow label" rather than more intensive packet inspection. There are immediate benefits to this set up. In resource-constrained environments, IPv6 requires less processing than IPv4, which can result in reduced power demands and latencies, especially for routers.

In addition, global routing tables in IPv6 are potentially much simpler than their IPv4 counterparts, something which can further lower memory and computational resource requirements. This is especially significant for mobile applications. Routing for mobile nodes is more efficient in IPv6 than in IPv4, and smooth handover techniques for IPv6 also exist with no IPv4 equivalents.

Unlike IPv4, security features are standardized and mandated with IPv6. IPv6 offers address manipulation techniques and secure neighbour discovery features that bolster network security. These include resistance to scanning and auto configuration of addresses, which makes it much more complicated for a malicious attacker to probe an organization's systems for weaknesses.

Packet fragmentation is a major source of packet delays, or high latency, under IPv4. IPv6 offers numerous improvements in the packet format that help minimize delay, jitter, and packet loss.

IPv6 uses a more sophisticated approach to handle data from programs requesting priority handling. The flow label and priority fields in the IPv6 header are used by a host to identify packets that need special handling by IPv6 routers, such as non-default quality of service or "real time" service.

4.11.3.3 Options

See 4.11.2.3.

4.11.3.4 Recommended practice

See 4.11.2.4.

4.12 E-ST-70-41: Telemetry and telecommand packet utilization

4.12.1 Purpose and usage

ECSS-E-ST-70-41 is one of the ECSS E-70 ground systems and operations series of standards. It specifies the telemetry and telecommand packet utilization standard, also known as the PUS. The PUS was developed independently by ESA and there is no equivalent CCSDS Recommended Standard.

ECSS-E-ST-70-41 defines a set of Application Layer services for spacecraft monitoring and control. The details of the services are not discussed here because they are outside the scope of the E-50 standards. ECSS-E-ST-70-41 is included because of its interfaces to the E-50 protocols and because it defines the use of certain fields in the Space Packets of the Space Packet Protocol (see 4.10.2). In particular, it defines the use of the Packet Data Field and its Packet Secondary Header.

At the time of writing this handbook, the current issue of ECSS-E-ST-70-41 does not use the name Space Packet but uses the earlier names: telemetry source packet and telecommand packet. There are also minor differences in the names and details of fields in the packets: for example, the Packet Secondary Header is called the Data Field Header.

4.12.2 Description

ECSS-E-ST-70-41 specifies services in the following areas of spacecraft monitoring and control:

- Telecommanding
- Telemetry reporting
- Software management
- On-board operations scheduling
- On-board monitoring
- On-board operations procedures
- Attaching actions to on-board events
- On-board storage and retrieval
- Telemetry generation and forwarding
- Memory management
- Diagnostics and testing

In addition to the services, the standard has requirements and recommendations in these areas and includes descriptions of suggested methods and behaviours.

4.12.3 Options

The services contained in ECSS-E-ST-70-41 cover a wide spectrum of operational scenarios. For a given mission, only a subset of these services is appropriate. The options for deciding which services to select and the options for the parameters of the selected services are outside the scope of this handbook.

4.12.4 Recommended practice

4.12.4.1 Use of Application Process Identifier (APID)

In ECSS-E-ST-70-41, the value in the APID field of a packet corresponds uniquely to the on-board application process which is the destination for the packet (on a telecommand link) or the source of the packet (on a telemetry link). For a typical mission where a single spacecraft has direct links to the ground system, this use of the APID field is generally sufficient. However, a different use of the APID field may be needed in more complex cases, such as:

- missions where the packets may need to be transferred via intermediate spacecraft, such as orbiters or space station elements;
- missions where there is a shared ground segment;
- multi-spacecraft missions.

The Space Packet Protocol includes the option to use the APID field as the naming mechanism for a Logical Data Path (LDP) from the source application to the destination application(s). See section 4.10.2 above for a discussion of the LDP.

4.12.4.2 Reserved values of APID

ECSS-E-ST-70-41 reserves the following APID values:

- APID=2047 (“all ones”) is reserved for the idle packet
- APID=0 is reserved for the standard spacecraft time packet.

CCSDS 135.0-B reserves the following APID values:

- APID=2047 (“all ones”) is reserved for the idle packet
- APID values 2045 and 2046 are reserved for specific protocols
- APID values 2040 to 2044 are reserved by CCSDS for future use.

Projects should always observe the requirements of CCSDS 135.0-B. It is recommended practice to observe the reserved APID values in ECSS-E-ST-70-41.

4.12.4.3 Packet Error Control

ECSS-E-ST-70-41 specifies a 16-bit Packet Error Control field which occupies the last 16 bits of the Packet Data Field. In ECSS-E-ST-70-41, the presence of the Packet Error Control field is mandatory for all Space Packets on a telecommand link. The field is optional for Space Packets on a telemetry link: the presence of the field is fixed for each APID. Any mission which is using the PUS should observe the ECSS-E-ST-70-41 requirements for the Packet Error Control field.

The Packet Error Control field is not included in the Space Packet specification in CCSDS 133.0-B-1 but is expected to be included in a future issue.

4.13 CCSDS 727.0-B: CCSDS File Delivery Protocol (CFDP)

4.13.1 Purpose and usage

In recent years, CCSDS has concentrated on providing flexible and efficient transfer protocols for various data over space links. The basic CCSDS suite solves the data transfer problems for current missions in which the manipulation of onboard storage tends to be handled manually, or by ad hoc protocols developed privately. While this is an acceptable way of managing a limited amount of memory, with the rapid development and take-up of solid state mass memory this is no longer the case.

The availability of gigabytes of solid state memory leads to a new era of spacecraft operation, where much of the routine traffic to and from the spacecraft will be in the form of files. Furthermore, because of the random access nature of the onboard storage medium, it becomes possible to repeat transmission of data lost on the link and thus guarantee delivery of critical information.

The requirement for a space file transfer mechanism arose because:

- Spacecraft now use mass memory with very large data files.
- For cost reasons, the trend is toward more autonomous operation whereby the spacecraft 'decides' (for example) when it should download stored data and when it should upload new operational plans.
- Interoperability within and among Agencies, and between space-ground networks (e.g. toward interoperability with the ground-based Internet) is becoming increasingly important as economic considerations require consolidation of networks.
- Some of the new deep space missions do not have direct line of sight between Earth and final destination; rather, data must be relayed between a series of spacecraft, each providing a store-and-forward capability, until the final destination is reached.
- Spacecraft constellations (e.g. fixed or formation-flying) require efficient and reliable data file transfer, possibly through multi-paths.
- The increasing onboard use of real-time operating systems (such as VxWorks and RTEMS), which assume the presence of a 'file system', makes onboard data handling increasingly file-oriented.

While the onboard storage medium has rapidly evolved, the essential constraints of space missions, which mitigate away from simple adoption of a terrestrial file transfer solution, remain, namely:

- Systems resources that may be restricted in one or both of the entities involved in an end-to-end data transfer may include computational power and memory capacities, driven by the need for expensive parts qualification, as well as the need to limit power, weight, and volume in the remote end system.

- Environmental restrictions may include noisy, bandwidth-limited, asymmetrical, and interrupted communications links with very long propagation delays.
- User needs often include a requirement for early access to transferred data regardless of its quality, as well as a method of providing data of progressively increased quality.

In response to these factors, the CCSDS File Delivery Protocol (CFDP) has been developed to complement the existing CCSDS packet standards. CFDP provides the capability to transfer 'files' to and from a spacecraft mass memory. The content of these files may be anything from a conventional timeline update to an unbounded SAR image.

Files can be transferred reliably, where it is guaranteed that all data will be delivered without error, or unreliably, where a 'best effort' delivery capability is provided. Files can be transmitted with a unidirectional link, a half-duplex link, or a full-duplex link, with near-Earth and deep space delays. File transfer can be triggered automatically or manually.

4.13.2 Description

The Recommended Standard CCSDS 727.0-B defines CFDP.

CFDP was designed to support the transfer of true files stored in a true file system. However, because CFDP is based on the concept of an abstract 'virtual filestore' (which in practice might be implemented in ways that are wholly unlike conventional 'file systems'), and because the CFDP specification does not define exactly what a 'file' is, the protocol can in practice be used to convey blocks of data between repositories that may not look like file systems. Such use, however, is a private matter within the using organization and should be defined by a local technical note or agreement among the using parties.

CFDP has many unique characteristics when compared to terrestrial file transfer protocols, such as:

- Efficient operation over simplex, half-duplex, and full-duplex links.
- Transfers that can span ground station contacts (time disjoint connectivity).
- Transfers that can span multiple ground stations.
- Effectiveness over highly unbalanced link bandwidths.
- Minimization of link traffic.
- Data availability to the user as the file is received.
- Minimization of onboard memory requirements through buffer sharing.
- Operation through multiple intermediaries (multiple hops).
- End-to-end accountability even through multiple store-and-forward intermediaries.
- Automatic store-and-forward operation.
- Store-and-forward initiation before the file is completely received at the forwarding entity.

- Effectiveness spanning low Earth orbit and deep space.

CFDP enables the moving of a file from one filestore to another, where the two filestores are in general resident in separate data systems and often with an intervening space link. In addition to the purely file delivery-related functions, the protocol also includes file management services to allow control over the storage medium.

The protocol is independent of the technology used to implement data storage and requires only a few fundamental filestore capabilities in order to operate. It assumes a minimum of two filestores, typically one within the spacecraft and one on the ground, and operates by copying data between the two adjacent filestore locations.

The protocol makes no assumptions about the information being transferred and can be utilized for a wide range of applications involving the loading, dumping, and control of spacecraft storage. The protocol has been specifically designed to minimize the resources required for operation. It is also scaleable, so that only those elements required to fulfil the selected options need be implemented. The protocol can operate over a wide range of underlying communication services, specifically including CCSDS packet services.

In its simplest form, the protocol provides a *Core* file delivery capability operating across a single link. For more complex mission scenarios, the protocol offers *Extended* operation providing store-and-forward functionality across an arbitrary network, containing multiple links with disparate availability, as well as subnetworks with heterogeneous protocols.

The Core procedures constitute the interaction between two protocol entities with a direct network path between them. The sending entity is the entity from which the file is copied in a file copy operation. The receiving entity is the entity to which the file is copied in a file copy operation.

Where direct network connectivity between the source and destination is impossible, the Extended procedures automatically build an end-to-end file copy transaction by executing multiple file copy operations, as follows: one file copy operation between the source and the first waypoint; others between successive waypoints as necessary; and a final file copy operation between the last waypoint and the destination. Each of these is simply another instance of the Core file copy operation.

The reliability of a transaction is determined by whether the transaction is chosen to operate in unacknowledged mode or in one of the acknowledged modes. In unacknowledged mode data delivery failures are not reported to the sender and, therefore, cannot be repaired, although errors will be detected and erroneous data discarded. Reception of the complete file is therefore not guaranteed. In acknowledged mode, the receiver informs the sender of any undelivered file segments or ancillary data. These are then retransmitted, guaranteeing complete file delivery. Each transaction results in the copying of a single file from source to destination.

When the Extended procedures are operating, the sender or receiver of a given Protocol Data Unit (PDU) may be a ‘waypoint’ CFDP entity. Extended procedures are used when the original source of the PDU has no direct connectivity to the PDU’s final destination, but only to some intermediate entity. The waypoint entity in turn may have direct connectivity either to the

PDU's final destination or only to some further intermediate entity; the last waypoint entity in such a chain must have direct connectivity to the final destination of the PDU.

All such scenarios as described above force each end-to-end transmission to entail a series of two or more point-to-point exchanges of data. For this purpose, the CFDP architecture is extended as follows:

> A *Store and Forward Overlay* (SFO) system is added to user applications as a standard user operation. The user application at each relay point examines each incoming file and, if the accompanying data in the PDU indicates that the file's final destination is elsewhere, initiates another point-to-point file transmission either to the final destination or to another relay point that is farther along the route.

> *Extended procedures* are added to CFDP itself. The CFDP entity at each relay point checks the final destination of each incoming file and, if necessary, initiates another point-to-point transmission toward that destination; the file is never delivered to the user application at any relay point.

The Extended procedures and the store and forward overlay are alternatives, and it is not necessary that both be implemented, although such implementation is acceptable. Extended procedures might be selected, for example, if it is desirable that portions of a file be forwarded by a waypoint immediately upon receipt, rather than only upon receipt of the entire file. This can get the initial parts of a file delivered to the final destination sooner than under SFO, even though final reception of the entire file at the final destination should take about the same length of time either way. Conversely, if detailed reporting on transaction status is required and there is more than one waypoint in the transaction path, then the store and forward overlay system might be preferred.

It is expected that in deep space use, a pair of CFDP entities that have files to exchange may at any given moment be unable to communicate; for example, a spacecraft orbiting Mars may be on the far side of the planet, unable to transmit to Earth. For this reason CFDP, when using store and forward overlay or Extended procedures, is built entirely on a *store and forward* communication model. If transmission of a file from Earth to a Mars-orbiting spacecraft is interrupted when the spacecraft passes behind the planet, the CFDP entities at both ends of the transmission simply store their outbound PDUs (possibly in non-volatile memory, to assure continued service even in the event of an unplanned system reset) until the spacecraft re-emerges and transmission can resume.

Using powerful forward error correction coding can minimize data loss in communication across deep space but cannot eliminate it altogether. Consequently CFDP supports optional 'acknowledged' modes of operation in which data loss is automatically detected and retransmission of the lost data is automatically requested. However, the large signal propagation delays that characterize interplanetary transmission limit the usefulness of the retransmission strategies commonly used in terrestrial protocols.

CFDP's retransmission model is one of *concurrent transmission*: data PDUs for multiple files may be transmitted as rapidly as possible, one after another, without waiting for acknowledgment, and requests for retransmission are handled asynchronously as they are received. As a result, portions of multiple files may be in transit concurrently.

The determination of how and when a waypoint entity forwards a PDU toward its target entity is an implementation matter. In general it is desirable to forward each PDU as soon as possible, rather than wait until custody of an entire file has been taken before forwarding any part of it; this approach minimizes the time required for complete end-to-end transmission of the data. In practice, however, immediate forwarding will frequently be impossible, because radio contact among CFDP entities is typically discontinuous. The waypoint entity in such cases must store PDUs in some persistent medium, such as an intermediate copy of the transmitted file, until forwarding is practical.

4.13.3 Options

CFDP defines four classes of service:

- Class 1 – Unreliable Transfer (no CFDP waypoints)
- Class 2 – Reliable Transfer (no CFDP waypoints)
- Class 3 – Unreliable Transfer via One or More Waypoints in Series (using Extended procedures)
- Class 4 – Reliable Transfer via One or More Waypoints in Series (using Extended procedures)

In addition, the Store and Forward Overlay (SFO) is defined as an upper layer to CFDP.

The essential difference between Extended procedures and SFO is that the Extended procedures are incorporated into the body of CFDP and provide for simultaneous reception and forwarding of data within a file whereas the SFO receives and forwards data on a file by file basis.

4.13.4 Recommended practice

CCSDS has developed three Green Books (CCSDS 720.1-G, CCSDS 720.2-G, CCSDS 720.3-G) providing background information on the operation and use of CFDP. The reader is encouraged to refer to these for more detailed information. This subsection highlights a number of points of a more general nature to be considered by the mission implementer.

CFDP can in principle be used in all scenarios involving the transfer of files to and from the spacecraft. The selection of class of service to be used will strongly depend on the nature of the mission and in particular the quality of the underlying links. For example, the forward link is already well projected against error and may incorporate automatic retransmission using the telecommand COP-1 protocol. Under these circumstances the unreliable class, combined with reporting of the received file, will provide an acceptable solution. The return link is also well protected against error but may be subject to outages due to atmospheric conditions, particularly for X-band links. Under these circumstances a class 2 reliable transfer may be more appropriate. Classes

3 and 4 are intended for multi-hop configurations where more than one satellite is used in the communication path to and from earth.

CFDP operates on the basis of a 'Transaction', which is essentially a communication session between the sending and receiving CDFP protocol engines. There are timers associated with the 'Transaction'. For unreliable transfer modes of operation the timers will automatically terminate once the last CFDP (EOF) PDU has been transmitted. For reliable transfer modes there are several timers which must be set in accordance with the link characteristics, in particular the round trip delay. In addition to timers, CFDP has a parameter known as "link lost", essentially a parameter raised by the underlying transmission service to indicate that the transmission service is no longer available (typically related to LOS in the space ground link). If this signal is received by the CFDP protocol it will 'freeze' the latest transmission status. Once the "link lost" signal has been removed the protocol will continue from where it left off. This signal offers a relatively simple mechanism to suspend and resume CFDP transmission between mission contact periods.

It should be noted that CFDP provides automatic retransmission of missing file segments based on information emitted by the receiver. It does not support the manual manipulation of missing segments whereby an operator may send a separate telecommand request for transmission of a missing segment. The automatic operation assumes access to the forward and return transmission paths, and for retransmission purposes, makes assumptions on the round trip time between sender and receiver. The implementer must therefore ensure the necessary transmission bandwidth is available and the timers are set accordingly.

CFDP requires an underlying packet based transmission capability which is able to encapsulate the CFDP PDU and direct it to the receiving CFDP entity. The CCSDS Space Packet Protocol (see section 4.10.2) will serve this purpose for the majority of missions. For missions using the ECSS-E-ST-70-41 (PUS) standard (see 4.12), a decision must be taken as to the use of PUS. In principle, PUS is not required as the CCSDS APID provides the necessary information for routing packets between the two communicating CFDP entities. Where transmission bandwidth is at a premium, directly multiplexing the CFDP PDUs into the CCSDS packet also provides the lowest overhead. However, it is also possible to include the use of PUS based packets for CFDP transmission as long as CFDP cross-support from another Agency is not a mission requirement.

Recommended practices for the use of SFO and Extended procedures (i.e. CFDP service classes 3 and 4) are not currently defined. This is related to future development and take-up of a CCSDS Delay Tolerant Networking (DTN) initiative.

4.14 CCSDS 135.0-B: Space Link Identifiers

4.14.1 Purpose and usage

The Recommended Standard CCSDS 135.0-B, *Space Link Identifiers*, is the central repository for reserved values of identifiers associated with the CCSDS protocols. Despite its title, the standard is not limited to the protocols which run directly over the space links but also includes values for protocols in the network layer and above.

The identifiers are used to identify protocols, addresses and data formats. The standard shows how these identifiers are managed, and, where applicable, lists the identifier values that are defined or reserved by CCSDS.

4.14.2 Description

Table 4-6 shows the identifiers covered by the standard that are relevant to the ECSS protocols discussed in this handbook. In particular, the standard lists the reserved values of the APID (Application Process Identifier) carried by the Space Packets of the Space Packet Protocol.

Table 4-6: Some of the identifiers covered by CCSDS 135.0-B

Identifier	Used by	Management Method
Transfer Frame Version Number	TM, TC, AOS, Proximity-1	Defined by CCSDS
Spacecraft Identifier (SCID)	TM, TC, AOS, Proximity-1	Assigned by CCSDS (see section 4.15)
Virtual Channel Identifier (VCID)	TM, TC, AOS	Managed by projects, with one value for AOS defined by CCSDS
Frame Secondary Header Version Number	TM	Defined by CCSDS
MAP Identifier (MAP ID)	TC	Managed by projects
Port Identifier (Port ID)	Proximity-1	Defined by CCSDS
CLCW Version Number	TC	Defined by CCSDS
Packet Version Number (PVN)	TM, TC, AOS, Proximity-1	Defined by CCSDS
Protocol Identifier	Encapsulation Packet	Defined by CCSDS
Application Process Identifier (APID)	Space Packet Protocol	Managed by projects, with some values defined or reserved by CCSDS

The Space Data Link Protocols provide some addresses (or identifiers) to identify data streams. The TM, TC and AOS Space Data Link Protocols all have the following three identifiers, which are carried in the header of the transfer frame:

- Transfer Frame Version Number (TFVN),
- Spacecraft Identifier (SCID),
- Virtual Channel Identifier (VCID).

The concatenation of a TFVN and a SCID is known as a Master Channel Identifier (MCID). All transfer frames with the same MCID on a physical channel constitute a master channel. A master channel consists of one or more virtual channels, each of which is identified with a VCID. See 4.6.2 for a description of master channels and virtual channels in the context of the TM Space Data Link Protocol.

In most cases, all the transfer fames on a physical channel have the same MCID, and the master channel is identical with the physical channel. However, a physical channel may carry transfer frames with different MCIDs (with the same TFVN). In such a case, the physical channel consists of multiple master channels.

A physical channel is identified with a Physical Channel Name, which is set by management and not included in the header of the transfer frames.

The TC Space Data Link Protocol uses an optional identifier called the Multiplexer Access Point Identifier (MAP ID) that is used to create multiple streams of data within a virtual channel. All the transfer frames on a virtual channel with the same MAP ID constitute a MAP channel. If the MAP ID is used, a virtual channel consists of one or multiple MAP channels. See 4.7.3 for more about MAPs.

Figure 4-13 shows the relationship among the channels of the Space Data Link Protocols.

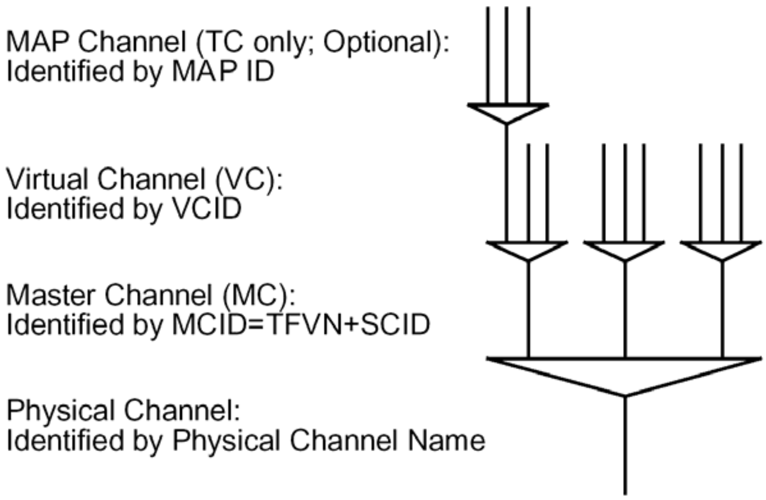

Figure 4-13: Relationships between channels of the Space Data Link Protocols

The Proximity-1 Space Link Protocol has the Transfer Frame Version Number and the Spacecraft Identifier but does not have the Virtual Channel Identifier. It has a 1-bit identifier called the Physical Channel Identifier. It also has another identifier called the Port Identifier that is used to route user data internally (at the transceiver's output interface) to specific logical ports, such as applications or transport processes, or to physical ports. Proximity-1 does not define a master channel.

Table 4-7 summarizes the addressing capability of the Space Data Link Protocols.

Table 4-7: Space link addressing

Identifiers	TM Space Data Link Protocol	TC Space Data Link Protocol	AOS Space Data Link Protocol	Proximity-1 Space Link Protocol—Data Link Layer
Transfer Frame Version Number (TFVN)	Always 1 (binary encoded number is 00)	Always 1 (binary encoded number is 00)	Always 2 (binary encoded number is 01)	Always 3 (binary encoded number is 10)
Spacecraft Identifier (SCID)	0 to 1023	0 to 1023	0 to 255	0 to 1023
Physical Channel Identifier (PCID)	N/A	N/A	N/A	0 to 1
Virtual Channel Identifier (VCID)	0 to 7	0 to 63	0 to 63	N/A
Multiplexer Access Point Identifier (MAP ID)	N/A	0 to 63	N/A	N/A
Port Identifier	N/A	N/A	N/A	0 to 7

Two types of addresses are used by the CCSDS packet and IP v4/v6 Network Layer protocols:

- Path Address
- End System Address.

A Path Address identifies a logical data path in the network from a source to one or multiple destinations. Configuration of logical data paths is done by management activities, and not by the protocol that transfers data. An End System Address identifies a single end system or a group of end systems. When an End System Address is used, a pair of End System Addresses must be used to identify both source and destination end systems.

Table 4-8 shows the addressing capability of the Network Layer protocols.

Table 4-8: Space network addressing

Protocol	Address name	Address type	Address length	Note
Space Packet Protocol	Application Process Identifier (APID)	Path address	11 bits	When APIDs have to be uniquely identified in a global network, the Spacecraft Identifier (SCID) is used to qualify APIDs
IP Version 4	IP Address	End system address	4 octets + 4 octets	
IP Version 6	IP Version 6 Address	End system address	16 octets + 16 octets	

4.14.3 Options

There are no options associated with the Recommended Standard CCSDS 135.0-B.

4.14.4 Recommended practice

In addition to the guidance on use of the APID in the Recommended Standard CCSDS 135.0-B, the ECSS *Telemetry and telecommand packet utilization* standard (PUS), ECSS-E-ST-70-41, makes use of the APID address space to differentiate PUS services. The PUS is discussed in section 4.12.

It should be noted that the APID allocations in CCSDS 135.0-B have been called into question with reference to CFDP. In this case multiple APIDs, rather than the allocated one, may be required to cater for duplex operation.

4.15 CCSDS 320.0-B: CCSDS Global Spacecraft Identification Field

4.15.1 Purpose and usage

The Recommended Standard CCSDS 320.0-B, *CCSDS Global Spacecraft Identification Field: Code Assignment Control Procedures*, establishes the procedures governing CCSDS Spacecraft Identification (SCID) field codes which are used on space links. The standard defines procedures for the requesting, assigning, using, relinquishing, and managing of SCIDs.

4.15.2 Description

The purpose of the CCSDS SCID is to serve as a mechanism for the identification of:

- a simple spacecraft having only one logical space-ground link;
- an association between space-based and ground-based application processes with complex spacecraft having more than one logical space-ground link. Therefore, a single spacecraft may be assigned more than one SCID.

This identification may be used only throughout a spacecraft's active phases, e.g., simulations, pre-launch testing, and in-orbit operations. As quickly as practical after reception of telemetry data, the SCID should be replaced with a globally unique, unambiguous, permanent, and SCID-independent label for the spacecraft and payload data sets. Thereafter, access to and identification of the data sets should use the SCID-independent label, rather than the value in the SCID field.

The procedures in CCSDS 320.0-B are intended to eliminate the possibility that data from any given CCSDS-compatible vehicle will be falsely interpreted as being from another CCSDS-compatible vehicle during the periods of simulation, testing, or mission operations.

The Recommended Standard identifies the roles of various agencies and bodies in controlling SCIDs, procedures for controlling them and the relevant agency contact points.

4.15.3 Options

There are no options associated with this Recommended Standard.

4.15.4 Recommended practice

4.15.4.1 Agency representative

All issues related to SCID assignment should be directed to the relevant agency representative.

4.15.4.2 Length of the SCID field in the transfer frame

Note that for a link using TM Transfer Frames (see 4.6.2) or TC Transfer Frames (see 4.7.3) or the frames defined for Proximity-1 (see 4.9.2), the SCID field in the frame header has 10 bits. For a link using AOS Transfer Frames (see 4.8), the SCID field in the frame header has only 8 bits.

4.15.4.3 Entity with multiple SCIDs

There is no single scenario which leads to multiple spacecraft IDs being allocated to an entity. The scenario could occur in, for instance:

- orbiter/lander or carrier/lander combinations
- spacecraft with multiple modules (e.g. space stations)
- spacecraft with temporarily docked service vehicles
- clusters of communicating spacecraft sharing external links.

4.15.4.4 Spacecraft with two telemetry links

Recommended practice considers two cases for a spacecraft with two telemetry downlinks:

- One link carries TM Transfer Frames and the other carries AOS Transfer Frames. The same SCID can be used for both links: the frames can be distinguished by the different transfer frame version numbers. In this case, the SCID value needs to be small enough to be carried in the 8-bit SCID field of an AOS Transfer Frame.
- Both links carry TM Transfer Frames. In this case, two SCIDs are needed in order to distinguish between the frames from the two links. The presence of the optional fields in the frames can vary between the links so that, for example, the frames on one link have the Operational Control Field but the frames on the other link do not.

4.15.4.5 Physical channel with two SCIDs

The TM, AOS and TC Space Data Link Protocols each contain the option to multiplex frames containing different values of the SCID field onto a single physical channel. Current practice does not recommend the use of two or more SCIDs on a physical channel.

4.16 Data Compression

4.16.1 Purpose and usage

There are two CCSDS Recommended Standards for data compression. Both standards relate to the Application Layer discussed in section 3.3.5:

- CCSDS 121.0-B: *Lossless Data Compression*
- CCSDS 122.0-B: *Image Data Compression*

Data compression, also known as source coding, is the process of encoding information so that it uses fewer bits than the unencoded representation. The advantages of data compression for space applications include:

- reduction of transmission channel bandwidth
- reduction of the buffering and storage requirement
- reduction of data transmission time at a given data rate.

Lossless data compression preserves source data accuracy and removes redundancy in the data source. In the decompression process, the original data can be reconstructed from the compressed data by restoring the removed redundancy; the decompression process adds no distortion. This technique is particularly useful when data integrity cannot be compromised. The price it pays is generally a lower compression ratio, which is defined as the ratio of the number of original uncompressed bits to the number of compressed bits including overhead bits necessary for signaling parameters.

Lossless data compression is applicable to a wide range of digital data, both imaging and non-imaging, where the requirement is for a moderate data-rate reduction constrained to allow no distortion to be added in the data compression / decompression process.

Under lossy compression, quantization or other approximations used in the compression process result in the inability to reproduce the original data set without some distortion.

CCSDS 121.0-B specifies a lossless data compression. The data compression specified in CCSDS 122.0-B has the option for lossless or lossy compression.

For lossless compression of imaging data, potential users are strongly encouraged to compare the performance (in terms of compression ratio) between the lossless data compression algorithm of CCSDS 121.0-B and the lossless mode of operations of CCSDS 122.0-B. In effect, the latter should provide significantly better performance, but the price to pay is, of course, increased complexity. A trade-off of performance against complexity should be made in the case of imaging data.

See section 5.5 for further discussion on choosing a compression algorithm.

4.16.2 CCSDS 121.0-B: Lossless Data Compression

4.16.2.1 Description

The Recommended Standard CCSDS 121.0-B specifies an adaptive data compression algorithm that has widespread applicability to many forms of digital data. In particular, the science data from many types of imaging or non-imaging instruments are well suited for the application of the algorithm. The compression algorithm can accommodate input samples ranging from 1 to 32 bits per sample.

In the standard, the characteristics of the data compression are specified only to the extent necessary to ensure multi-mission support capabilities. The specification does not attempt to quantify the relative bandwidth reduction, the merits of each approach discussed, or the design requirements for coders and associated decoders. The decompression process is not addressed. Implementers and mission planners are strongly encouraged to consult the informational report CCSDS 120.0-G for detailed background and application information associated with the standard, including performance and implementation.

The standard addresses the data compression, the associated data formats and the methods for packetization when using the Space Packets of the Space Packet Protocol (see 4.10.2) to transport the data. In CCSDS 121.0-B, Space Packets are called source packets.

4.16.2.2 Options and parameters

4.16.2.2.1 Introduction

CCSDS 121.0-B provides one option and it also includes several parameters that are tunable by the user to suit the application.

4.16.2.2.2 Option for selection/definition of a pre-processor

The algorithm is composed of two modules: the first is the pre-processor, which is a reversible transform. Its task is to decorrelate the input data stream so that the statistics at the output of the module are as close as possible to a Laplacian distribution after a potential reordering of symbols which is done by the mapper. The pre-processor should be designed according to the type of correlation exhibited by the input stream. This correlation can be strictly monodirectional or multi-directional. The simplest pre-processor, and the one most commonly used with this algorithm, is the predictive coder, where the value of the current sample is predicted using a linear combination of previously encoded samples. The coefficients of this linear combination can be static or adaptive. The difference (or residue) between the incoming sample and the prediction is coded by the entropy coder which is the second module of the algorithm. The better the pre-processor, the lower the entropy level of its output to the coder.

4.16.2.2.3 Parameter for block sizes (8 or 16 samples)

The block size should be selected according to the variability of the entropy level of the input data. Best approach is, as often in data compression, simulation on representative data sets.

4.16.2.2.4 Parameter for number of CDS per packet (1 to 4096)

The output from the coding process forms coded blocks called Coded Data Sets (CDS). A Space Packet generally carries multiple CDS. The packetization provides error containment because any bit error on a CDS will cause desynchronisation of the decoding process and potentially corrupt decompressed data until a resynchronisation occurs. This resynchronisation mechanism is only provided by the packet header. Therefore, opting for a large number of CDS per packet will improve performances in terms of compression ratio, but at the expense of extensive error propagation in case of transmission or storage errors.

4.16.2.2.5 Parameter for number of CDS between references

References are samples inserted uncoded into the compressed data stream so as to avoid endless propagation of an error due to predictive coding and decoding performed in the pre and post processor. Increasing the number of CDS between references will typically increase compression ratio but also increase error propagation in case of a noisy channel.

4.16.2.3 Recommended practice

Recommended practice is limited to the only option of this algorithm, the pre-processor. The simplest decorrelator of all should be considered first, i.e. nearest neighbour prediction (preceding pixel on the same line).

4.16.3 CCSDS 122.0-B: Image Data Compression

4.16.3.1 Description

The Recommended Standard CCSDS 122.0-B establishes a standard for a data compression algorithm applied to two-dimensional digital spatial image data from payload instruments and specifies how this compressed data is formatted into segments to enable decompression at the receiving end. The data compression algorithm is applicable to a wide range of space-borne digital data, where the requirement is for a scalable data reduction, including the option to use lossy compression, which allows some loss of fidelity in the process of data compression and decompression.

Under lossy compression, quantization or other approximations used in the compression process result in the inability to reproduce the original data set without some distortion. The increased information content of data subjected to lossless compression results in a larger volume of compressed data for a given source data set.

The compression technique described in the standard can be used to produce both lossy and lossless compression. It differs from the widely used JPEG2000 standard in several respects:

- it specifically targets high-rate instruments used on-board spacecraft;
- a trade-off has been performed between compression performance and complexity with particular emphasis on spacecraft applications;
- its lower complexity supports fast and low-power hardware implementation;
- it has a limited set of options and parameters, supporting its successful application without in-depth algorithm knowledge.

The compressor consists of two functional parts, depicted in Figure 4-14, a Discrete Wavelet Transform (DWT) module that performs decorrelation and a Bit-Plane Encoder which encodes the decorrelated data.

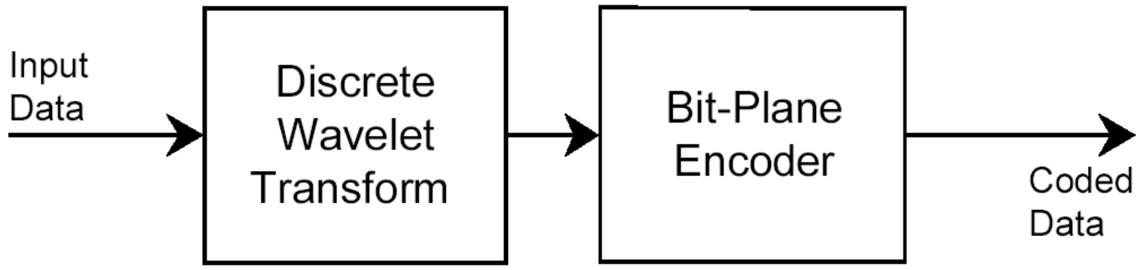

Figure 4-14: Functional parts of the compressor

The standard supports both frame-based input formats produced, for example, by CCD arrays (called image frames) and strip-based input formats produced by push-broom type sensors (called image stripmaps). An image pixel dynamic range of up to16 bits is supported. The algorithm specified supports a memory-effective implementation of the compression procedure.

The standard does not attempt to explain the theory underlying the operation of the algorithm. The characteristics of instrument data are specified only to the extent necessary to ensure multimission support capabilities. The specification does not attempt to quantify the relative bandwidth reduction, the merits of each approach discussed, or the design requirements for coders and associated decoders. For information on these issues, the reader is encouraged to consult the informational report CCSDS 120.1-G.

4.16.3.2 Options and parameters

4.16.3.2.1 Introduction

CCSDS 122.0-B provides several options and it also includes several parameters that are tunable by the user to suit the application. Chapter 3 of CCSDS 120.1-G-1, *Image Data Compression – Informational Report*, provides potential users with additional information on options and parameters.

4.16.3.2.2 Option for type of compression: *lossless or lossy*

Depending on mission requirements bit accurate (lossless) compression can be required or not. CCSDS 122.0-B accommodates both cases. Nevertheless, it should be noted that lossless compression is usually very ineffective in terms of compression performances (poor compression ratio). Accepting a minor and visually unnoticeable degradation will improve significantly the compression performances (typically multiply by 2). Therefore, the decision to go lossless on image compression for a mission should be carefully traded-off with on-board storage and transmission constraints.

If lossless mode is wanted, the user has to:

- select integer DWT
- set DCStop to 0, BitPlaneStop to 0, and StageStop to 3
- for each segment, set the value of SegByteLimit sufficiently large to accommodate the compressed data volume needed for losslessly encoding that segment (for convenience, SegByteLimit can be set to the maximum allowed value of 2^{27}).

4.16.3.2.3 Option for type of Discrete Wavelet Transform (DWT): *integer or floating point*

For the lossless mode, integer DWT is mandatory. But for lossy mode, CCSDS 122.0-B includes two types of DWT:

- The integer DWT provides a comparatively low-complexity transform implementation-wise, while it will give a slightly lower compression performance, especially for moderate to high compression ratio.
- Floating point DWT is appropriate for software implementation.

For hardware and high throughput implementation, integer and floating point DWT should be traded-off considering the difference in compression performances, which can be fairly minor in a given mission environment (type of images, range of compression ratio targeted, …).

4.16.3.2.4 Option for type of control: *controlling "compressed data volume" or "image quality"*

There are schematically two scenarios for an on-board image compression system:

- It interfaces with a fixed rate telemetry link and therefore is required to output a compressed data stream at a constant data rate.
- It interfaces with a mass memory which accepts variable rate input data and in that case can produce a constant quality compressed data stream.

In fact, users of compression systems should be aware that:

- fixed compressed data rate means variable image quality
- fixed image quality means variable compressed data rate.

CCSDS 122.0-B allows both options : fixed data rate or fixed image quality. A hybrid scheme of control is also provided, whereby the compression algorithm will stop encoding when it encounters the first of two user defined limits: a maximum data volume or a minimum image quality. This hybrid mode enables the user to satisfy the image quality requirements of the mission most of the time, without oversizing the output interface of the compression unit to cope with the rare case of high information density images.

Note: The "fixed data rate" mode can be made "octet accurate" (i.e. compressed data volume can be controlled by the octet). The "fixed image quality" mode needs to be calibrated for the type of images the mission will acquire. In effect, the relationship between compression error and the number of bit planes transmitted (parameter which controls the image quality in the algorithm) is image dependent. Nevertheless, it should be fairly constant for a given mission/instrument.

4.16.3.2.5 Parameter for number of blocks per segment (S)

Wavelet coefficients (produced by the DWT decorrelator) are coded in blocks of 8x8 using a tree-based bit-plane encoder. Encoded blocks are grouped into segments. Each segment is encoded entirely independently from the others. Therefore, no error propagation can occur between two segments. The segment is the primary error containment data structure. Furthermore, the segment header and the algorithm enable the user to vary certain compression parameters at segment level, such as the targeted data rate for the segment or the targeted image quality. Selecting the number of blocks per segment should take into account:

- Error propagation requirement of the mission (maximum image area lost due to an error).
- Compression unit internal memory size: the entire segment should be stored inside the compression unit before it can be completed for release.
- Smaller segments means more segment header information in the compressed bitstream therefore lower compression performances.
- Compression rate control requirements of the mission, which can be at image level or at a sub-level (block of lines, …).

4.16.3.2.6 Rice code parameters

Rice code is used to entropy code the bit-planes. This entropy coder has two variants of operation with respect to the selection of Rice code option:

- optimal selection (which means exhaustively computing the best option – the one that produces the least amount of bits)
- heuristic selection (which uses a heuristic selection formula).

Both selection methods give similar performance with a typical 0,5% difference in terms of output data rate in favour of optimal selection. Therefore, choosing between the two methods is a matter of implementation complexity.

4.16.3.2.7 Parameter for custom subband weights

For integer DWT, a default set of weights is provided in the CCSDS 122.0-B standard. These weights are multiplication factors which should be used at the output of the DWT to amplify wavelet coefficients according to their contribution to the reconstruction of the image. The set of weights given in the standard will provide the optimal compression performance if a Mean Square Error (MSE) image quality criterion is being used to judge the compression.

The user can decide to change the default set of weights if the quality of the compressed image is not to be judged by a quadratic quality criterion like MSE, or if, for example, the image quality is to be measured not on the decompressed image directly but on some post-processed image (e.g. after deconvolution).

No weights should be used with the floating point DWT.

4.16.3.3 Recommended practice

All the options and parameters offered by the CCSDS 122.0-B standard are useful ones. They have been introduced to cope with the diversity of mission requirements with regards to compression. The trade-offs to be performed on a given mission have been indicated in the preceding section.

For transmission of the compressed data stream over a telemetry space link, it is recommended that an integer number of segments is inserted in a telemetry packet. If the main source of end-to-end error on the link is packet loss, then the number of segments per packet should be kept low (ideally one). In effect, it is useless to have a small error containment structure at segment level if most errors cause the loss of a full packet containing many segments.

4.17 CCSDS 301.0-B: Time Code Formats

4.17.1 Purpose and usage

Time codes are digital representations of time information. The purpose of Recommended Standard CCSDS 301.0-B is to establish a small number of standardized time code formats designed for applications involving data interchange in space data systems. It does not address timing performance issues such as stability, precision and accuracy.

4.17.2 Description

The Recommended Standard defines four time codes:

- Unsegmented Time Code (CUC)
- Day Segmented Time Code (CDS)
- Calendar Segmented Time Code (CCS)
- ASCII Calendar Segmented Time Code (ASCII)

4.17.3 Options

In addition to the option to choose one of the four time codes, there are options within each time code which affect, for example, the accuracy of the time representation.

4.17.4 Recommended practice

The Recommended Standard does not attempt to prescribe which code to use for any particular application. The rationale behind the design of each code is described in Annex B of CCSDS 301.0-B and may help the application engineer to select a suitable code.

The telemetry and telecommand packet utilization standard, ECSS-E-ST-70-41, has an annex that specifies spacecraft time protocols. The procedures in the annex use the time code formats from CCSDS 301.0-B.

4.18 E-ST-50-12: SpaceWire. Links, nodes, routers and networks

4.18.1 Purpose and usage

SpaceWire is a communications network for use onboard spacecraft. The SpaceWire standard, ECSS-E-ST-50-12, specifies a serial interconnection interface at physical, electrical and data communication protocols, enabling the reliable sending of data at high-speed (between 2 Mbits/s and typically 200 Mbits/s). SpaceWire links can be used to interconnect devices implemented on the same board, modules within a unit and as an interface between equipments.

SpaceWire links are full-duplex, point-to point serial data communication links. Networks can be built to suit particular applications using point-to-point data links and routing switches.

One of the main advantages of SpaceWire is its low complexity and the fact that it can be implemented easily in both ASICs and FPGAs with a low gate count implementation. SpaceWire is supported by several radiation tolerant ASICs designed by or for ESA, NASA and JAXA, and extensive test and development facilities are available.

4.18.2 Description

SpaceWire is designed to connect together high data-rate sensors, processing units, memory sub-systems and the down link telemetry sub-system. It provides high-speed, bi-directional, full-duplex, data links which connect together the SpaceWire enabled equipments. Application information is sent along a SpaceWire link in discrete packets. Control and time information can also be sent along SpaceWire links.

A SpaceWire network comprises links, nodes and routers. A SpaceWire node is typically an electronic equipment that needs to use the services of the SpaceWire network. One or more SpaceWire interfaces connect the node to SpaceWire links. The other ends of these links are either connected directly to other nodes or are connected to a SpaceWire router which provides an indirect connection to other nodes.

A SpaceWire router comprises a set of SpaceWire interfaces and a non-blocking cross-bar switch. Packets of information arriving at a router are forwarded towards their destination by switching them to pass out of the most appropriate link attached to the router. Each SpaceWire packet comprises a packet destination address, the information being carried in the packet (cargo) and an end of packet marker. The destination address is either a unique identity code for the destination node or a description of the path through the SpaceWire network to the destination node. The SpaceWire router uses the destination address to determine which link it should use to forward a packet.

The standard covers the following protocol levels:

- Physical level: Defines connectors, cables, cable assemblies and printed circuit board tracks.
- Signal level: Defines signal encoding, voltage levels, noise margins, and data signalling rates.
- Character level: Defines the data and control characters used to manage the flow of data across a link.
- Exchange level: Defines the protocol for link initialization, flow control, link error detection and link error recovery.
- Packet level: Defines how data for transmission over a SpaceWire link is split up into packets.
- Network level: Defines the structure of a SpaceWire network and the way in which packets are transferred from a source node to a destination node across a network. It also defines how link errors and network level errors are handled.

4.18.3 Options

The ECSS-E-ST-50-12 standard does not contain options as far as point to point links are concerned. Nevertheless, the standard keeps some parameters open, such as the link speed and packet lengths. The same applies for packet routing aspects for which addressing schemes are fully defined but not implementation level features linked to a specific routing switch.

ECSS-E-ST-50-12 is complemented by a set of data transfer protocols as specified in ECSS-E-ST-50-11.

NOTE At the time of writing this handbook, ECSS-E-ST-50-11 is under development.

4.18.4 Recommended practice

A specific normative section in the standard provides conformance criteria. It is highly recommended to assess a specific implementation or usage conditions against these criteria.

In particular, it is worth stressing that the high transmission bandwidth offered by Low Voltage Data Signaling drivers (LVDS) induces specific requirements and precautions at physical and electrical interface implementation level. SpaceWire links built according to the standard have demonstrated the capability to reach extremely low, not to say non-measurable bit error rates, even in adverse EMC conditions. As this property is driving higher level quality of service aspects, users of the SpaceWire standard have to be particularly cautious when departing from the requirements and recommendations as expressed by the standard in this respect.

As packets are allowed to be of arbitrary size, application level decisions have to be taken in order to manage this feature and/or bound the packet size to a defined value. This aspect has to be considered in particular when SpaceWire links are interconnected via routers as part of an on-board network.

More information about *SpaceWire – Links, nodes, routers and networks* can be found at: http://spacewire.esa.int.

4.19 E-ST-50-13: MIL 1553 Standard extension

4.19.1 Purpose and usage

The ECSS-E-ST-50-13 standard provides a comprehensive set of requirements for all communicating equipment units and components onboard a spacecraft, which are connected to a single (redundant) data bus according to MIL-STD-1553B. It includes the definition of a set of data transmission related services and protocols for operation in conjunction with the MIL-STD-1553B data bus (Milbus).

4.19.2 Description

Although the Milbus has been around for a significant period, it is still used in many spacecraft as the preferred real-time bus. The MIL-STD-1553 standard concentrates on the physical interface and does not specify the protocols to be used to support data transfer over the bus. As a result of this an enormous number of protocols have been developed, typically a new or slightly different version for each mission. All protocols provide essentially the same capability but each new protocol requires detailed specification development and debug leading to largely unnecessary costs and efforts.

In an attempt to reduce the number of new developments, this ECSS standard defines a single approach which should serve the needs of the majority of missions. Although the standard focuses on the specification of single-bus architecture, questions related to multiple-bus-architectures or the use of repeaters for separable busses (for launchers) are also addressed.

This ECSS standard introduces the concept of services and it maps these to the services being defined by CCSDS for onboard communication (see CCSDS SOIS in section 6.1.4).

For more information on the ECSS-E-ST-50-13 standard, readers are directed to the standard itself, which has an excellent informative section containing a detailed description of the protocol and its capabilities.

4.19.3 Recommended practice

The ECSS-E-ST-50-13 standard should be applied to all spacecraft implementing the Milbus for command and control or for payload interfacing purposes.

4.20 E-ST-50-14: Spacecraft discrete interfaces

4.20.1 Purpose and usage

The ECSS-E-ST-50-14 standard, *Spacecraft discrete interfaces*, specifies a set of spacecraft onboard electrical interfaces. The standard defines discrete (i.e. point to point) interfaces commonly used for status measurement and control, as well as serial digital interfaces used for digital data acquisition and commanding of devices. It specifies:

- interface signal identification;
- interface signal waveforms;
- signal timing requirements;
- signal modulation;
- voltage levels;
- input and output impedance;
- over voltage protection requirements;
- bit ordering in digital data words;
- cabling requirements where appropriate.

The standard does not cover:

- connector requirements;
- digital data word semantics;
- message or block formats and semantics;
- interfaces for command and control serial busses (these are covered or planned to be covered by other ECSS E-50 standards).

The goal of this standard is to establish a single set of definitions for these interfaces and to promote generic, reusable implementations.

The standard is applicable to all projects, as well as to single components within units (e.g. ASICs). It may also be applied to R&D projects with low TRL (Technology Readiness Level), since it allows interoperability of components and standardization of test equipment.

4.20.2 Description

The interfaces specified in ECSS-E-ST-50-14 are intended to connect DHS (Data Handling System) core elements to DHS peripheral elements as shown in Figure 4-15. However, there are no technical reasons to prevent these interfaces being used between core elements where it is appropriate to do so, and the standard does not preclude such a configuration.

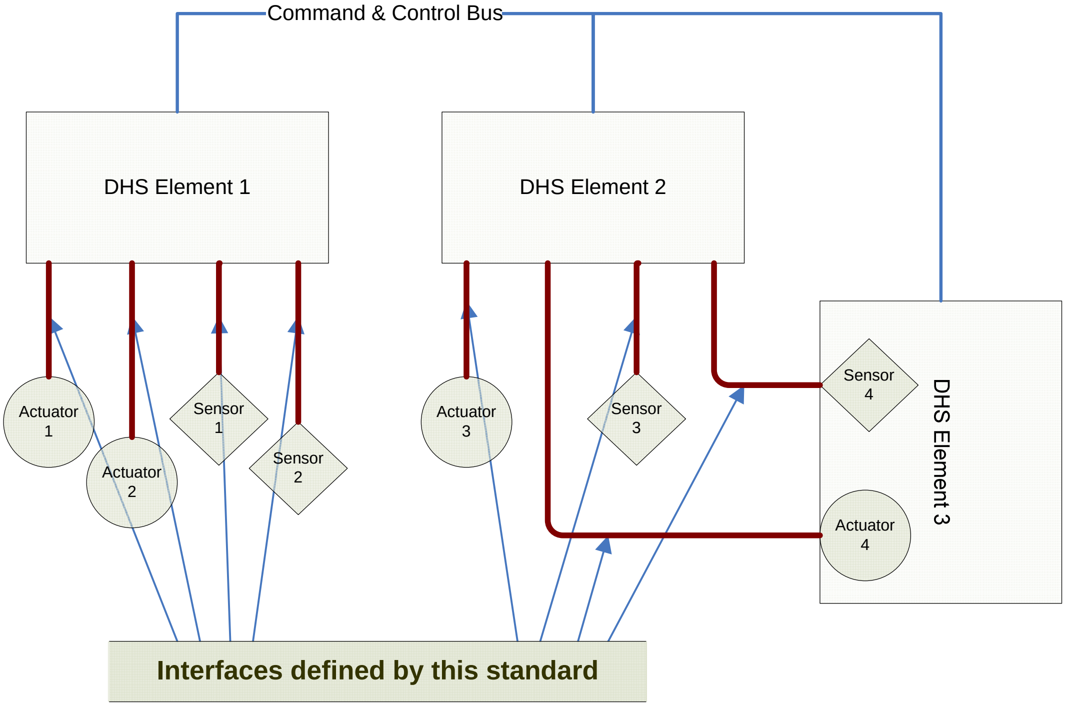

Figure 4-15: Architectural context of interfaces defined in ECSS-E-ST-50-14

A peripheral element can have more than one user interface and also user interfaces of different types, depending on its function and design. For example, some sensors can set threshold levels or sensitivities by means of data written to them. In this case a sensor may use an output serial digital interface to write the data in addition to an input serial digital interface to read the sensor value. Alternatively, some devices are signalled to indicate that they are commanded to acquire a data sample. In that case they can use a serial digital interface together with a pulse interface.

4.20.3 Options

When viewed in a specific context, the requirements defined in the standard can be tailored to match the actual requirements of a particular profile and circumstances of a project. Nevertheless, the tailoring is limited to the adoption of specific discrete interfaces listed below:

- Analogue signal interfaces
- Analogue signal monitor (ASM) interface
- Temperature sensors monitor (TSM) interfaces
- Bi-level discrete input interfaces
- Bi-level discrete monitor (BDM) interface
- Bi-level switch monitor (BSM) interface
- Pulsed command interfaces
- High power command (HPC) interfaces

- Low power command (LPC) interface
- Serial digital interfaces
- 16-bit input serial digital (ISD) interface
- 16-bit output serial digital (OSD) interface description
- 16-bit bi-directional serial digital (BSD) interface description

Modification of existing or addition of requirements within a specific interface definition may not be done.

4.20.4 Recommended practice

The ECSS-E-ST-50-14 standard:

- should be used as an applicable document in the SRD (System Requirements Document) for a project;
- should be used to support the avionics design.

5 Specific topics

5.1 Introduction

This section gives advice on various aspects of applying the E-50 standards in a mission.

5.2 Synchronization and channel coding

5.2.1 Functions of synchronization and channel coding

The ECSS E-50 standards specify the functions necessary for safely transferring Transfer Frames over space links:

- Delimiting/synchronizing Transfer Frames, so that the receiving system can find the beginning of each frame in the incoming stream of bits.
- Channel coding/decoding for forward error-correction, to provide recovery from errors that occur on the physical link.
- Error detection and frame validation.
- Bit transition generation/removal, to provide a minimum bit transition density so that the receiving system can maintain bit synchronization with the incoming signal.

With the exception of frame validation, these functions are in the Synchronization and Channel Coding Sublayer of the Data Link Layer (see 3.3.2). Two of the E-50 standards define synchronization and channel coding functions:

- ECSS-E-ST-50-01, *Telemetry synchronization and channel coding*, for telemetry links (TM and AOS). The equivalent CCSDS Recommended Standard is CCSDS 131.0-B, which contains more options than ECSS-E-ST-50-01.
- ECSS-E-ST-50-04, *Telecommand protocols, synchronization and channel coding*, for telecommand links. The equivalent CCSDS Recommended Standard is CCSDS 231.0-B, which contains more options than ECSS-E-ST-50-04.

For the Proximity-1 Space Link Protocol, the synchronization and channel coding functions are defined in CCSDS 211.2-B. Table 5-1 summarizes the functions and schemes provided by the synchronization and channel coding standards. For telemetry and telecommand, the table applies to the ECSS standards, ECSS-E-ST-50-01 and ECSS-E-ST-50-04, rather than the equivalent CCSDS Recommended Standards.

Table 5-1 Functions of synchronization and channel coding standards

	Telemetry (TM and AOS) **ECSS-E-ST-50-01**	**Telecommand** **ECSS-E-ST-50-04**	**Proximity-1** **CCSDS 211.2-B**
Error correction	Convolutional codes * Reed-Solomon codes * Turbo codes *	BCH code	Convolutional code *
Error detection / Frame validation	Reed-Solomon codes * Frame Error Control Field (1)	BCH code Frame Error Control Field (2)	32-bit Attached Cyclic Redundancy Code
Pseudo-randomization	Cyclic pseudo-noise sequence *	Cyclic pseudo-noise sequence	(not used)
Frame synchronization	32-bit Attached Sync Marker (3)	16-bit Start Sequence	24-bit Attached Sync Marker

* indicates an option

(1) 16-bit Cyclic Redundancy Code, specified in ECSS-E-ST-50-03 (TM) or CCSDS 732.0-B (AOS). Optional or mandatory depending on the channel coding in use.

(2) 16-bit Cyclic Redundancy Code, mandatory in ECSS-E-ST-50-04.

(3) For turbo codes, a longer Attached Sync Marker is used.

5.2.2 Coding options

For telecommand links, ECSS-E-ST-50-04 specifies the channel coding to be used: there are no options to consider. For performance information on the channel coding, see Annex D of ECSS-E-ST-50-04, which shows, for example, how the frame rejection probability increases with the length of the frame. For performance information on the telecommand channel coding options that are included in CCSDS 231.0-B but excluded from ECSS-E-ST-50-04, see the informational report CCSDS 230.1-G.

For telemetry links, ECSS-E-ST-50-01 offers a full range of channel coding options. Table 5-2 shows coding gains and radio frequency bandwidths for most of the coding options. For the coding gain figures for the Reed-Solomon code (E=8) with 4D-8PSK-TCM, see Annex D of ECSS-E-ST-50-01.

In Table 5-2, the radio frequency bandwidth is shown in comparison with an uncoded channel. It shows the ratio of the bandwidth of the coded channel to the bandwidth of an uncoded channel with the same information bit rate. (The ratio is the inverse of the code rate so, for example, for turbo code rate 1/2, the ratio is 2.)

In the table, the coding gains are given for frame error rate (FER) = 10^{-4} and 10^{-6}, frame length = 8920 bits and under the assumption that the channel is additive white Gaussian noise (AWGN) and BPSK modulated. The BPSK modulation has been assumed for the purpose of coding scheme performance comparison only: it is not intended to suggest that BPSK modulation is suitable or unsuitable for use on any specific space channel. Under the same conditions, the theoretical value of the lowest bit-energy-to-noise ratio Eb/No to achieve an FER of 10^{-4} with no coding scheme is 11.9 dB (± 0.05 dB). Similarly, to achieve an FER of 10^{-6} the value is 13.0 dB (± 0.05 dB).

The coding gain values already include allowance for the increase in symbol rate. It is not necessary to deduct, say, 3dB in achieved extra margin to account for a doubling of symbol rate when using rate1/2 convolutional coding.

Table 5-2: Coding gains and bandwidth expansions

Coding scheme	Bandwidth relative to uncoded channel	Coding gain for frame length = 8920 bits (dB)	
		FER = 10^{-4}	FER = 10^{-6}
Uncoded	1	0	0
(255, 223) Reed-Solomon only	1.14	5.4	6.2
Punctured convolutional rate 7/8	1.14	3.8	3.9
Punctured convolutional rate 5/6	1.2	4.9	
(255, 223) R-S and punctured convolutional rate 7/8	1.31	6.8	7.7
Punctured convolutional rate 3/4	1.33	5.3	5.4
(255, 223) R-S and punctured convolutional rate 5/6	1.37	7.5 (1)	8.4 (1)(2)
Punctured convolutional rate 2/3	1.5	5.8	
(255, 223) R-S and punctured convolutional rate 3/4	1.52	8.2 (1)	9.6 (1)
(255, 223) R-S and punctured convolutional rate 2/3	1.71	8.8 (1)	9.8 (1)(2)
Turbo code rate 1/2 (information block length = 8920 bits)	2	10.8	11.6
Basic convolutional k=7, rate 1/2	2	6.1	6.6
(255, 223) R-S and basic convolutional rate 1/2	2.28	9.4 (1)	10.8 (1)
Turbo code rate 1/4 (information block length = 8920 bits)	4	11.7	12.5

(1) Performance obtained with R-S interleaving depth $I = 5$

(2) Extrapolated value

The performance figures in Table 5-2 were obtained by simulations, where the absence of synchronization losses has been assumed.

For the basic and punctured convolutional codes (alone or concatenated) the performance values shown in the table were obtained with 8-bit quantization soft decisions.

For the turbo codes, the performance values shown in the table were obtained with a turbo decoder having the following characteristics:

- component decoders are soft-input, soft-output, "A posteriori probability" (APP) type decoders
- quantization of channel symbols is at least 6 bits/symbol
- quantization of decoder metrics is at least 8 bits
- number of decoder iterations is 10.

The coding gain figures in Table 5-2 are given for the purposes of comparison and can differ from the practical performance of a space channel.. For a given FER value, coding gain varies with the frame size, the interleaving depth, the decoding algorithm and other factors.

For further performance information, including the bit error rate (BER) performance of the various coding options over a range of signal to noise ratios, see CCSDS 130.1-G.

ECSS-E-ST-50-01 has channel coding options with code rates ranging from 1/(1.14) (Reed-Solomon only) to 1/4 (turbo code rate 1/4), and with coding gains ranging from 5dB (Reed-Solomon only) to 12dB (turbo code rate 1/4). The coding options cover, in small steps, the useful range of rates and coding gains. In general, the smaller the coding rate, the higher the coding gain.

This abundance of options has been introduced in ECSS-E-ST-50-01 to accommodate the wide variety of link conditions and constraints. Annex D of ECSS-E-ST-50-01 gives preferred coding schemes to achieve frame error rates in the range 10^{-4} to 10^{-6}. The material from Annex D is not reproduced here and the reader is recommended to consult the annex.

In choosing the appropriate coding option for a given mission, the following factors should be considered:

- Occupied bandwidth constraints on the link. Those constraints, together with the information data rate to be transmitted, can actually limit the channel coding rate and hence limit the achievable coding gain.
- Coding gain constraints. Link budget can require a minimum coding gain from the channel coding. In all cases, coding gain should always be traded-off against transmitting power, antenna gains, information data rate, etc.

Coding and decoding complexity. As can be seen from Table 5-2, turbo code rate 1/2 and basic convolutional rate 1/2 have exactly the same bandwidth of the coded channel, but the turbo code is typically 5dB better in term of coding gain. Nevertheless, the price to pay is a significantly higher coding and decoding complexity, which can prevent real-time implementation of turbo code decoding at high data rates (typically above 20 Mbps). Therefore, data rates, real time constraints and decoding complexity should be analysed before choosing the optimal coding option.

5.3 Playback of recorded data

5.3.1 Legacy systems with stored TM Transfer Frames

Clause 5.4.4. of ECSS-E-ST-50-03 specifies a way to transmit data that has been stored onboard in a legacy system. The stored data consist of TM Transfer Frames that are recorded in an on-board memory device, such as a magnetic tape recorder. The recorded frames are played back later at a convenient time, and placed in real-time TM Transfer Frames for transmission.

The clause should only be used for systems where the recorded data is stored in the form of standard TM Transfer Frames. Typically, these are the TM Transfer Frames of the real-time physical channel at the time of recording with their attached synchronization markers. For easy retrieval, each recorded conventional TM Transfer Frame is tagged with a large count number. This can be achieved, for each virtual channel, with the extended virtual channel frame count.

At playback time, the recorded TM Transfer Frames are placed into the Transfer Frame Data Field of real-time TM Transfer Frames. This process is called asynchronous insertion. A dedicated virtual channel is used for the playback data.

The asynchronous insertion may be made in either the forward or the reverse mode. If forward mode is used, then any recorded attached synchronization markers must use the alternative synchronization marker pattern of hexadecimal 352EF853. Otherwise, the presence of forward-justified attached synchronization markers in the data fields of the real-time TM Transfer Frames can cause disruptions to the frame synchronization process at the receiving end, particularly during acquisition.

At the receiving end, the real-time virtual channel used for the playback data is processed and its contents stored away for later off-line processing. In the later off-line processing, the recorded TM Transfer Frames are retrieved in the correct, forward-justified direction and processed as for a real-time physical channel.

Any Communications Link Control Word extracted from the Operational Control Field of a recorded TM Transfer Frame should not be processed by the real-time telecommand system.

5.3.2 Systems with stored AOS Transfer Frames

The header of an AOS Transfer Frame includes the Replay Flag. The flag is set to one when transmitting a frame that is replayed from onboard storage. For real-time frames the Replay Flag is set to zero. See CCSDS 732.0-B for more information on the use of the Replay Flag.

5.3.3 Current systems

Current systems usually have more options for the formats and devices used to store data on board for later transmission. For example, data can be stored in mass memory. When convenient, the data can be transferred using one of the protocols discussed in this handbook. The appropriate protocol depends on the data format used for storage,(e.g. packets, frames, or an un-structured sequence of octets or bits) and the nature of the on-board application that retrieves the data for transmission.

5.4 Idle data

5.4.1 Synchronous channel: TM and AOS

5.4.1.1 Idle data in all or part of the Transfer Frame Data Field

The TM Transfer Frames (see 4.6.2) and the AOS Transfer Frames (see 4.8.1.2) are fixed length frames. The frames are transferred synchronously over a physical channel. That is, they are transferred at a constant rate, without gaps, so that there is a regular time interval between the start of one frame and the start of the next one. When the sending end is generating the frames it must meet this timing constraint.

If there is no data available, or insufficient data available, the sending end can put idle data into all or part of the Transfer Frame Data Field. The standards define various ways this can be done, to suit the amount of data available and the timing circumstances of the virtual channel or the physical channel. So, for example, if the available packets only occupy part of the Transfer Frame Data Field, then an idle packet can be generated to occupy the remaining space in the field.

Table 5-3 shows the options for putting idle data into all or part of the Transfer Frame Data Field of a TM Transfer Frame. Table 5-4 shows the options for an AOS Transfer Frame.

Table 5-3: Idle data in a TM Transfer Frame

Sync Flag	First Header Pointer	Transfer Frame Data Field	Remarks
0	2046	- contains only idle data - the pattern of the idle data is a mission configuration parameter	- also called an OID Transfer Frame - at the receiving end, the Transfer Frame Data Field is ignored - other fields in the frame can carry useful data
0	other value	contains packets, which can include idle packets	- at the receiving end, any idle packets are ignored - in addition, other fields in the frame can carry useful data
1	undefined	contents not specified by the standard	- use of idle data in this case is not defined - in addition, other fields in the frame can carry useful data
The value of the Sync Flag is fixed for a virtual channel for the duration of a mission phase.			

Table 5-4 Idle data in an AOS Transfer Frame

Virtual Channel	Transfer Frame Data Field		Remarks
VC with VCID 63	- contains only idle data - the idle pattern is project specific		- called an Idle Frame - at the receiving end, the Transfer Frame Data Field is ignored - other fields in the frame can carry useful data
VC carrying M_PDUs (M_PDU includes the First Header Pointer and the Packet Zone)	First Header Pointer = 2046	Packet Zone contains only idle data	- at the receiving end, the Packet Zone is ignored - other fields in the frame can carry useful data
	other value for First Header Pointer	Packet Zone contains packets, which can include idle packets	- at the receiving end, any idle packets are ignored - in addition, other fields in the frame can carry useful data
VC carrying B_PDUs	Bitstream Data Zone can contain idle data as indicated by the Bitstream Data Pointer		- at the receiving end, any idle data are ignored - in addition, other fields in the frame can carry useful data
VC carrying VCA_SDUs	- contains one VCA_SDU - contents of a VCA_SDU are undefined		- use of idle data in this case is not defined - in addition, other fields in the frame can carry useful data

The use of VCID 63 is defined by the standard. For all other virtual channels, the type of data carried by the virtual channel (M_PDU, B_PDU or VCA_SDU) is fixed for the duration of a mission phase.

It is important to note that the idle data only affects the Transfer Frame Data Field. Regardless of the contents of the Transfer Frame Data Field, other fields in the frame still carry useful information:

- For a TM Transfer Frame, there can be useful data in the Transfer Frame Primary Header and in the optional Transfer Frame Secondary Header and Operational Control Field (OCF).
- For an AOS Transfer Frame, there can be useful data in the Transfer Frame Primary Header and in the optional Inset Zone and OCF.

5.4.1.2 Idle packets

As shown in Table 5-3 and Table 5-4, idle packets can be placed in the Transfer Frame Data Field of a TM Transfer Frame or in the Packet Zone of an M_PDU in an AOS Transfer Frame. Typically, one idle packet of appropriate length is generated to fill the remaining space in the field. However, there can be multiple idle packets in the field, and it also possible for a normal packet to follow an idle packet within the field. From the point of view of the sending end it can be simpler to generate many short idle packets to fill the remaining space but the packets increase the processing load in the receiving system: some telemetry systems have limits on the packet processing rate.

In current implementations, the idle packets are the Idle Packets defined for the Space Packet Protocol in CCSDS 133.0-B. The Idle Packet is a special case of the Space Packet, with the APID value set to “all ones”. The minimum length of an Idle Packet is seven octets. Therefore, in cases where the remaining space in the field is less than seven octets, the Idle Packet spills over into a subsequent frame.

A new option is being introduced, where an idle packet can be a Fill Encapsulation Packet defined for the Encapsulation Service in CCSDS 133.1-B. The Fill Encapsulation Packet is a special case of the Encapsulation Packet, with the Protocol ID value set to zero. The minimum length of a Fill Encapsulation Packet is one octet. It is expected that there will be revisions to the CCSDS Recommended Standards to clarify the use and the definition of the Fill Encapsulation Packet. Meanwhile, the availability of implementations to support this option is likely to be limited.

5.4.1.3 Pattern of idle data

The pattern of the idle data in a transfer frame or in an idle packet is project specific.

An idle data pattern of all zeroes can be simple to generate but has no bit transitions and therefore can cause problems in maintaining bit synchronization at the receiver. Data which is periodic, even if the number of ones and zeroes is about equal, can interact with the channel coding and with the modulation scheme to cause output patterns that result in receiver acquisition problems and lock problems.

Projects which do not use the pseudo-randomization defined in ECSS-E-ST-50-01 need to be especially careful in their choice of the pattern for idle data.

5.4.2 Asynchronous channel: TC and Proximity-1

The TC Transfer Frames (see 4.7.3) and the transfer frames of the Proximity-1 Space Link Protocol (see 4.9) are variable length frames. The length of the frame can be adjusted to fit the amount of data available, within the limits set by the maximum frame length.

The frames are transferred asynchronously over a physical channel. There can be gaps between the frames. Unlike the synchronous channel, there is no regular time interval between the start of one frame and the start of the next one, so the sending end does not have the same timing constraints when generating the frames. Therefore, the TC and Proximity-1 protocols do not define mechanisms to put idle data into the frames. For a typical project, the frames do not need to carry idle packets but that does not exclude their use in project-specific cases.

For telecommand, ECSS-E-ST-50-04 specifies the PLOP-2 procedure, which is the procedure for transmitting frames. It includes the specification of the variable-length idle sequence that is transmitted in the gap between frames. For the asynchronous channel, the idle sequence keeps the channel synchronized at the bit sync level, in comparison to a synchronous channel where synchronization is maintained at the frame level.

5.5 Compression

CCSDS has developed two generations of data compression standards:

- Universal lossless data compression (CCSDS 121.0-B-1)
- Image Data Compression (IDC) (CCSDS 122.0-B-1)

The two standards have been designed to offer a good ratio of compression performance to complexity, which is highly needed in the on-board space environment. The standards are described in section 4.16.

In parallel to those space related standardisation efforts, the ISO/ITU joint committee, JPEG (Joint Photographic Experts Group), has developed three widely known and used standards, aimed at general public applications on the ground (digital camera, Internet image browsing, etc.). Those standards are:

- JPEG: lossy image compression. 80% of all images exchanged on the Internet are JPEG encoded.
- JPEG-LS: lossless image compression.
- JPEG2000: lossy image compression. It provides a 30% improvement in terms of data volume w.r.t. JPEG. This relatively new standard is in the deployment phase and should replace the “old” JPEG in a five-year timeframe.

Given this context, mission planners could be tempted to overlook CCSDS standards for compression and select instead a widely ground supported

ISO/ITU standard like JPEG2000 for lossy image compression and JPEG-LS for lossless.

The rationale for using CCSDS compression standards is linked to the following points:

- CCSDS standards offer options that are especially suited for spaceborne instruments (e.g. scan-based mode of operation for push-broom imagers);
- CCSDS standards offer a very good performance / complexity ratio. In comparison, the JPEG2000 and JPEG-LS standards are designed to optimize performance at the expense of a potentially very high complexity, especially when a hardware implementation of the algorithm is needed. To give orders of magnitude, the difference in performance between JPEG2000 and CCSDS IDC is less than 1dB (PSNR) or 10% (output data volume), while the difference in complexity is in the order of a factor two (number of operations per pixel, required internal memory per pixel). See CCSDS 120.1-G for a detailed comparison.

Selection between the two CCSDS data compression standards for a given mission can be done using the following criteria:

- CCSDS universal lossless compression should be used:
  - For 1D data. A custom pre-processor (or decorrelator) can be designed if specific and stable correlation patterns exist in the input stream. Otherwise the standard nearest neighbour predictor should be used as pre-processor.
  - For 2D data (images) when lossless compression is needed and implementation complexity should be kept to a minimum.
- CCSDS image data compression (IDC) should be used:
  - For all images, either for lossy or lossless compression, when fair to high compression performance is needed. This algorithm can be implemented with all its options at very high rate (typically 60 Mpixels/s per ASIC). Implementation complexity is intermediate between CCSDS universal lossless compression and JPEG2000.

5.6 Security

The ECSS-E-ST-50 standard specifies requirements on the function and performance of the space communications system: some of the requirements are about security. For example, it includes a requirement to ensure that only telecommands from authorized sources are executed onboard the spacecraft.

There are currently no ECSS standards or CCSDS Recommended Standards that specify the security methods, such as authentication or encryption algorithms, to be used with the protocols discussed in this handbook. Among the CCSDS informational reports about security issues, the most directly relevant is CCSDS 350.0-G, *The Application of CCSDS Protocols to Secure Systems*.

5.7 Interoperability

Interoperability between agencies is provided by a cross-support capability. Cross support requires the managerial and technical commitment of one agency to provide a communications service to another. The aspects which need to be agreed in order to provide a cross support service include:

- Service view
  - Functional characteristics of services
  - Performance characteristics of services
  - Methods and/or standards for using services
  - Methods and/or standards for managing services
- Physical view
  - Physical location
  - Topology and connectivity
  - Physical media for access
- Communications view
  - Communications protocols
  - Parameter values of communications protocols
- Enterprise view
  - Policies for providing services
  - Documents necessary for using services
  - Pricing information
  - Service agreement
  - Activities for testing cross support interfaces

These aspects reflect the approach that cross support addresses more than just the interface between two elements belonging to two different agencies or organisations. Rather, it is the case that one agency contracts a service to another agency and that this service has a number of aspects, only one of which is the agreement of protocols at the interfaces between communications system elements.

The cross support points are defined by the service being contracted for. Some typical cases include:

Service	**Cross support points**
Provision of ground station network	Ground network (SLE) Space link (TM/TC)
Provision of Mars relay	Control centre to control centre (File Transfer) Orbiter to lander (Proximity-1)

It should be noted that while CCSDS develops recommendations that are commonly agreed among all participating Agencies, the actual implementation of the recommendations by individual Agencies may have variations which include the selection of options. These variations may lead to two implementations both being compliant with the CCSDS recommendation but incompatible in terms of cross-support. This problem has to a great extent been resolved in the ground segment by the implementation of SLE services. Here cross-support services have been specially agreed and implemented by all leading space Agencies and this has led to a guaranteed provision of basic space-to-earth cross-support by the majority of Agencies participating in the CCSDS. For space-to-space cross-support, the situation is not so clear as no cross-support services have been defined. CCSDS has, however, recognised the problem and is actively pursuing a solution.

In the meanwhile, projects requiring space-to-space cross-support must carefully negotiate the required services with the cross-supporting Agency.

5.8 Conformance and PICS

The ECSS E-50 series of standards primarily concern the specification of communication protocols and the associated data formats. Typically, communication takes place between two entities implementing the sending and receiving ends of the protocol, or protocol stack. The implementations may be provided by a single supplier but more commonly they are provided by different industry teams, and in the case of interoperability, under the responsibility of different space Agencies. In order for the protocols to operate as intended they must of course be implemented and tested to ensure correct implementation but of equal importance they must be compatible in terms of the selected protocol options and implementation choices.

The problem of interoperation and conformance to a protocol specification has long been recognised by ISO and has been tackled by attaching a template to ISO communications standards which may be used to record the status of a particular implementation. This template, or proforma, when completed by an implementer, is known as a Protocol Implementation Conformance Statement (PICS).

The proforma consists of a PICS Requirements List (PRL) for an implementation. The supplier of a protocol implementation generates the PICS by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The PICS can have a number of uses, including use:

- by the protocol implementer, as a check-list to reduce the risk of failure to conform to the standard through oversight;
- by the supplier and acquirer — or potential acquirer — of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- by the user — or potential user — of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs);

- by a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

The provision of PICS is now becoming a general policy within CCSDS, and PICS proforma have been provide for more recent recommendations. The provision of PICS is also under consideration by ECSS for application to the E-50 series of standards. In support of this, draft PICS proforma have been developed for three of the ECSS E-50 standards as shown in Table 5-5. For each standard, there is a proforma for a sending end implementation and a proforma for a receiving end implementation. Each annex is intended to be usable as a free-standing document so there is some repetition of introductory material across the annexes. These PICS are provided as information pending a formal decision by ECSS.

Table 5-5: Draft PICS proforma included in the annexes

Standard	Sending / Receiving	Annex containing the PICS
ECSS-E-ST-50-01 Telemetry synchronization and channel coding	sending end	Annex A
	receiving end	Annex B
ECSS-E-ST-50-03 Telemetry transfer frame protocol	sending end	Annex C
	receiving end	Annex D
ECSS-E-ST-50-04 Telecommand protocols, synchronization and channel coding	sending end	Annex E
	receiving end	Annex F

6 Rationale and architectures

6.1 Evolution of space communications standards

6.1.1 The beginning of CCSDS

Early spacecraft communicated with the ground using tone based communications where individual modulated tones on the links registered certain events, analogue values (when continuous rather than stepped tones were used) or commands. They were limited in capability but simple and robust and still may find use in extremely link budget constrained scenarios such as planetary descent phases.

To get more information back and forth, we entered the age of Pulse Code Modulation (PCM) where analogue values were represented by digital words and events and commands could be represented by specific digital codes or single bits. Along with PCM came the bonus of easy repetitive multiplexing of digital values onto the links. The links were also thus referred to by the rather more appropriate term of Time Division Multiplexing (TDM).

Data was organised into fixed length frames and the meaning of data defined by its position in the frame. This was adequate for a while but, gradually, limitations began to be apparent. The limitations are exemplified by considering non-routine communications. If a table of future commands needs to be loaded, for instance, the long upload message would need to be squeezed into part of the up going frame and, since it was too long for a single period it had to be:

- Tagged to indicate what kind of data it was
- Segmented to account for the limited and fixed length field that was available

As spacecraft got more and more sophisticated the number and variability of these non-routine messages started to multiply and, when microprocessors with up-loadable software and parameter files became common on board spacecraft, the TDM way of doing things became too cumbersome and inflexible. With increasing spacecraft and operational complexity, it dawned that, actually, the self identifying, variable length messages were now the dominant form of communications traffic with the fixed repetitive telemetry and, even more so, telecommand being sidelined.

In response to this, the major space agencies pooled resources and embarked on a programme to develop protocols which were suitable for the purpose of space communications in an era of increasing technological and operational sophistication. They also took the opportunity to establish international, multi-agency standards so that different agencies' spacecraft and ground segments could interoperate, experience could be shared and technology development cost and risk could be shared. This was the beginning of CCSDS.

6.1.2 Standards for TC and TM

It was decided that, for the spacecraft of the time, asymmetrical protocols for uplink (TC) and downlink (TM) were required. Amongst the reasons for this were:

- It was important that telecommands were delivered complete, without errors and in sequence.
- The repetitive and highly redundant telemetry and science data of the time could easily tolerate loss of data.
- Telecommand data structures were typically shorter than TM structures and the data rates correspondingly lower.
- Less complex and capable Forward Error Correction (FEC) was achievable in the uplink because of limitations on onboard processing power.

6.1.3 Later standards: AOS, Proximity-1 and CFDP

In response to a perceived need for higher rate and more symmetrical up and down links, largely driven by the International Space Station (ISS), CCSDS developed the Advance Orbiting Systems (AOS) recommendation. The high rate uplink requirement largely evaporated in the light of more realistic project expectations and the symmetrical reliability capability foundered due to lack of firm requirements. AOS is sometimes applied for its suitability to high rate downlinks where conventional TM sequence counters would roll over during a contact period.

The Proximity-1 protocol was developed for planetary explorations missions. It is particularly suited for lander to orbiter communications where power, particularly at the lander, is limited. It also, via its automated orbiter/lander link establishment mechanism and variable rate capability, makes optimum use of limited spacecraft contact times.

The CCSDS File Delivery Protocol was developed in response to the emerging complexity of onboard data storage, where file based rather than location based storage was becoming prevalent onboard. It also addresses automation of ground control procedures, automatic custody transfers and the diminishing redundancy intrinsic in spacecraft return data.

6.1.4 New developments: DTN and SOIS

For the future, the Delay/Disruption Tolerant Networking (DTN) initiative within CCSDS seeks to rationalise planetary communications by taking terrestrial networking concepts and adapting them to the sporadic, error strewn and resource limited environment inherent in space exploration.

Of the current CCSDS recommendations under development, the SOIS (Spacecraft Onboard Interface Services) is worthy of note as it is already being applied in substance to ECSS standards preparation in the area of onboard data link protocols.

In the short term, the SOIS activity attempts to standardise the communication services used on board the spacecraft for command and control. Ultimately, the SOIS services will be completed by protocols to provide a standard service and protocol stack suitable for use by all onboard applications.

The SOIS defines an architecture consisting of three main layers:

- the subnetwork layer, defining services to be provided over the onboard bus
- the transfer layer, consisting of network and transport layer protocols
- the application support layer, defining a generic set of services to satisfy the majority of onboard applications.

While the communications services are being defined within CCSDS, the protocol specification is being performed by other expert groups. For example, the ECSS-E-ST-50-13 Milbus working group will define protocols to provide the SOIS services over a Milbus, the SpaceWire working group is performing the same task for SpaceWire and mappings for other busses are expected in the future. Protocols to implement the application support layer service are either being adopted from existing protocols or will be developed as needed by the SOIS group. The basic SOIS architecture is depicted in Figure 6-1. For further information the reader is referred to the SOIS area of the CCSDS web site: www.ccsds.org.

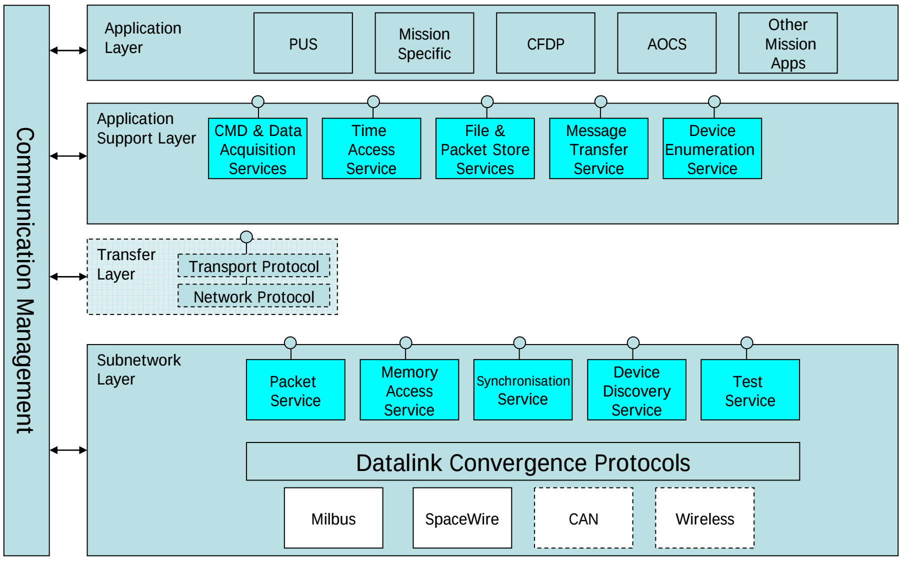

Figure 6-1: SOIS overall architecture

6.1.5 CCSDS and ECSS standards

As described in section 3.1.3, the CCSDS recommendations provide the source for the majority of the ECSS standards, either by derivation or direct adoption. This approach will continue in the future with ECSS expected to adopt all new communications-related CCSDS recommendations.

6.2 Guidelines

The following tables (Table 6-1 to Table 6-4) summarise the available ECSS E-50 standards and related CCSDS recommendations with guidance as to their applicability.

Table 6-1: Applicability of standards for space link

Ref	Title	Applicability
ECSS-E-ST-50-02	Ranging and Doppler tracking	All missions requiring Doppler tracking and ranging from earth stations
ECSS-E-ST-50-05	Radio frequency and modulation	All missions
ECSS-E-ST-50-01	Space data links - Telemetry synchronization and channel coding	Near Earth and Deep Space mission return links (for TM and AOS)
ECSS-E-ST-50-03	Space data links - Telemetry transfer frame protocol	Near Earth and Deep Space mission return links (except where AOS applies)
ECSS-E-ST-50-04	Space data links - Telecommand protocols, synchronization and channel coding	Near Earth and Deep Space mission forward links
CCSDS 732.0-B	AOS Space Data Link Protocol	Near Earth and Deep Space mission return links with very large data volumes per contact period
CCSDS 211.0-B	Proximity-1 Space Link Protocol - Data Link Layer	Inter-spacecraft links (typically lander/orbiter)
CCSDS 211.2-B	Proximity-1 Space Link Protocol - Coding and Synchronization Sublayer	
CCSDS 211.1-B	Proximity-1 Space Link Protocol - Physical Layer	

Table 6-2: Applicability of standards for Network / Transport / Application layers

Ref	Title	Applicability
CCSDS 133.0-B	Space Packet Protocol	Support to application layer protocols
CCSDS 133.1-B	Encapsulation Service	Support to IPv6 for all ECSS space links
RFC 791	Internet Protocol, Protocol Specification *[1]*	Mission specific
RFC 1883	Internet Protocol, Version 6 (IPv6) Specification	Mission specific
CCSDS 727.0-B	CCSDS File Delivery Protocol (CFDP)	File transfer and filestore manipulation Reliable return links
CCSDS 121.0-B	Lossless Data Compression	Modest compression, no data degradation
CCSDS 122.0-B	Image Data Compression	High compression, some data degradation
CCSDS 301.0-B	Time Code Formats	All missions
ECSS-E-ST-70-41	Ground systems and operations - Telemetry and telecommand packet utilization	All ESA missions
[1] now known as IPv4		

Table 6-3: Applicability of standards for management

Ref	Title	Applicability
ECSS-E-ST-50	Communications	All missions
CCSDS 320.0-B	CCSDS Global Spacecraft Identification Field: Code Assignment Control Procedures	All missions
CCSDS 135.0-B	Space Link Identifiers	All missions

Table 6-4: Applicability of standards for onboard

Ref	Title	Applicability
ECSS-E-ST-50-12	SpaceWire. Links, nodes, routers and networks	Mission specific
ECSS-E-ST-50-13	MIL 1553 Standard extension	Mission specific
ECSS-E-ST-50-14	Spacecraft discrete interfaces	All missions

7 Supporting components

7.1 Introduction

This section provides an overview of the components that are available to support the ECSS standards. It covers both flight and ground areas. It is non-exhaustive.

7.2 Flight components

ESA has a number of flight components (ASICs, IP cores) which have been developed in support of the standards outlined in this handbook. For further information refer to http://www.esa.int/TEC/Microelectronics/.

7.3 Ground components

The web portal for ESA Ground Operations Software (EGOS) provides information about all main ESA spacecraft operating systems, including SCOS-2000 (Spacecraft Operating System 2000), SIMSAT (Simulation Infrastructure for the Modelling of Satellites), ground station and communications systems, and the ESA SLE Gateway. It also documents some supporting software packages and models and gives access to papers and presentations from EGOS Workshops.

For access to the EGOS portal, go to http://www.egos.esa.int/portal/egos-web/.

7.4 Industry

The space industry supplies components that are designed to be compliant with the standards described in this handbook. The lists included here are not exhaustive and do not imply endorsement by ECSS of any supplier or product.

The ESA Industry Portal includes links to databases of ESA industry partners:

- the European Space Industry Directory
- the Small and Medium Enterprises Database
- a database of Universities and Research Centres

For access to the ESA Industry Portal, go to http://www.esa.int/home-ind/.

The CCSDS web site includes a section on industry resources. For access, go to the CCSDS home page at http://public.ccsds.org/ and select the tab "Implementations".

Annex A (informative) Draft PICS for ECSS-E-ST-50-01: Sending end

A.1 Introduction to the PICS

A.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telemetry synchronization and channel coding specified in ECSS-E-ST-50-01. It is provided as information pending a formal decision by ECSS.

Sections A.2 to A.10 constitute the proforma. The normative specification is in ECSS-E-ST-50-01.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.1.2 Format of the PICS

A.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

A.1.2.2. Mandatory or optional status

Table A-1 shows the symbols used in the 'status' column of a PRL table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table A-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional

A.1.2.3. Support provided by the implementation

For each item, the 'support' column of a PRL table contains either:

- one or more labelled boxes, with labels as shown in Table A-2, or
- the word 'Values'.

Table A-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in A.1.2.4.	

To show the conformance of an implementation, the support column of a PRL table is completed as follows:

- If an item in the table has labelled boxes, then exactly one of the given boxes in the support column should be ticked. If none of the boxes can be ticked, then the implementation is non-conformant and exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.

- If an item has the word 'Values', then the requested values should be supplied, as indicated in a note for the item. For example, item SCCB-2 in Table A-6 concerns the supported values for the Reed-Solomon interleaving depth, *I*. A list of at least one value for *I* should be entered in the support column.

A.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table A-9 contains items relating to turbo codes. If an implementation does not support turbo codes (item SCC-5 in Table A-4), then the N/A box at the top of Table A-9 is ticked and the rest of the table is ignored.

A.2 Identification

Table A-3 contains the items to identify the implementation.

Table A-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telemetry synchronization and channel coding specified in ECSS-E-ST-50-01		
Item	Description	Information
SI1	Date of Statement	
SI2	PICS serial number	
SI3	Supplier	
SI4	Contact details	
SI5	Implementation name	
SI6	Implementation version	
SI7	Other information for full identification, e.g. versions for operating systems	
SI8	Special configuration details	
SI9	Protocol version	
SI10	Protocol addenda and amendments	

A.3 Channel coding

This section specifies the items for the channel coding options supported by the implementation.

Table A-4: Supported channel codes

Item	Description	Clause	Notes	Status	Support	Remarks
SCC-1	Convolutional code (not concatenated with Reed-Solomon code)	5	See A.4	O	Yes [ ] No [ ]	
SCC-2	Reed-Solomon code (not concatenated with convolutional code)	6	See A.5	O	Yes [ ] No [ ]	
SCC-3	Concatenation of Reed-Solomon code and convolutional code	5 6	See A.6	O	Yes [ ] No [ ]	
SCC-4	(255,239) Reed-Solomon code and 4D-8PSK-TCM	6.4	See A.7	O	Yes [ ] No [ ]	
SCC-5	Turbo code	7	See A.8	O	Yes [ ] No [ ]	
SCC-6	No channel coding		See A.9	O	Yes [ ] No [ ]	
SCC-7	At least one of the options SCC-1 to SCC-6		Note 1	M	Yes [ ]	
Note 1: A conformant implementation supports one or more of the options SCC-1 to SCC-6.						

A.4 Convolutional code

This section specifies the items for the convolutional code, when it is not concatenated with the Reed-Solomon code.

Table A-5: Items for convolutional code

If Item SCC-1 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SCCA-1	Convolutional code rate 1/2 as defined in clause 5.3	5.3		O	Yes [ ] No [ ]	
SCCA-2	Punctured convolutional code as defined in clause 5.4	5.4		O	Yes [ ] No [ ]	
SCCA-3	At least one of the options SCCA-1 and SCCA-2		Note 1	M	Yes [ ]	
SCCA-4	Supported code rates for punctured convolutional code	5.4.a	Note 2	O	Values:	
SCCA-5	If differential encoding (i.e. conversion from NRZ-L to NRZ-M) is used, then the conversion is performed at the input to the convolutional encoder	5.2.c	Note 3 Note 4	O	Yes [ ] No [ ] N/A[ ]	
SCCA-6	The ASM is attached to each transfer frame before the convolutional encoding	8.3.a 8.4		M	Yes [ ]	
SCCA-7	Transfer frame lengths supported		Note 5	M	Values:	
SCCA-8	Pseudo-randomization of the transfer frame, before attachment of the ASM	9.3 9.4	Note 6	O	Yes [ ] No [ ]	

Note 1: If the implementation supports SCC-1, then it supports one or both of the options SCCA-1 and SCCA-2. If the implementation does not support SCC-1, see the instructions at the top of this table.

Note 2: If SCCA-2 is supported, a list of at least one code rate is entered in the support column. If SCCA-2 is not supported, 'N/A' is entered in the support column.

Note 3: This item is not mandatory but it is recommended in ECSS-E-ST-50-01, "Convolutional coding – General", which contains a 'should' recommendation.

Note 4: If the implementation does not support the conversion from NRZ-L to NRZ-M, then support for this item is not applicable.

Note 5: The values entered in the support column show the frame lengths, measured in octets. The values can be entered as a range or as a list. Note that ECSS-E-ST-50-03 defines a maximum length of 2048 octets for TM Transfer Frames.

Note 6: Use of pseudo-randomization by a mission or project is optional. Support for item SCCA-8 implies that the implementation supports the convolutional code with and without pseudo-randomization.

A.5 Reed-Solomon code

This section specifies the items for the Reed-Solomon code, when it is not concatenated with a convolutional code. This section does not apply to the Reed-Solomon code with 4D-8PSK-TCM specified in ECSS-E-ST-50-01, Clause "Reed-Solomon with E=8".

Table A-6: Items for Reed-Solomon code

If Item SCC-2 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SCCB-1	(255,223) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2	6.3.1 6.3.2		M	Yes [ ]	
SCCB-2	Supported values for the interleaving depth, *I*	6.3.3.a	Note 1	M	Values:	
SCCB-3	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
SCCB-4	Supported shortened lengths for each supported interleaving depth	6.3.6	Note 2	O	Values:	
SCCB-5	The ASM is attached to each R-S codeblock after the R-S encoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
SCCB-6	Pseudo-randomization of the R-S codeblock after the R-S encoding, before attachment of the ASM	9.3 9.4	Note 3	O	Yes [ ] No [ ]	

Note 1: A list of at least one value for *I* is entered in the support column.

Note 2: If SCCB-3 is supported, the values entered in the support column provide details of the shortened lengths. If SCCB-3 is not supported, 'N/A' is entered in the support column.

Note 3: Use of pseudo-randomization by a mission or project is optional. Support for item SCCB-6 implies that the implementation supports the Reed-Solomon code with and without pseudo-randomization.

A.6 Concatenation of Reed-Solomon code and convolutional code

This section specifies the items for the concatenated code, consisting of the Reed-Solomon code with a convolutional code.

Table A-7: Items for concatenated R-S and convolutional code

If Item SCC-3 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SCCC-1	(255,223) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2	6.3.1 6.3.2		M	Yes [ ]	
SCCC-2	Supported values for the interleaving depth, *I*	6.3.3.a	Note 1	M	Values:	
SCCC-3	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
SCCC-4	Supported shortened lengths for each supported interleaving depth	6.3.6	Note 2	O	Values:	
SCCC-5	The ASM is attached to each R-S codeblock, after the R-S encoding, before the convolutional encoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
SCCC-6	Convolutional code rate 1/2 as defined in clause 5.3	5.3		O	Yes [ ] No [ ]	
SCCC-7	Punctured convolutional code as defined in clause 5.4	5.4		O	Yes [ ] No [ ]	
SCCC-8	At least one of the options SCCC-6 and SCCC-7		Note 3	M	Yes [ ]	
SCCC-9	Supported code rates for punctured convolutional code	5.4.a	Note 4	O	Values:	
SCCC-10	If differential encoding (i.e. conversion from NRZ-L to NRZ-M) is used, then the conversion is performed at the input to the convolutional encoder	5.2.c	Note 5 Note 6	O	Yes [ ] No [ ] N/A[ ]	
SCCC-11	Pseudo-randomization of the R-S codeblock after the R-S encoding, before attachment of the ASM	9.3 9.4	Note 7	O	Yes [ ] No [ ]	

Note 1: A list of at least one value for *I* is entered in the support column.

Note 2: If SCCC-3 is supported, the values entered in the support column provide details of the shortened lengths. If SCCC-3 is not supported, 'N/A' is entered in the support column.

Note 3: If the implementation supports SCC-3, then it supports one or both of the options SCCC-6 and SCCC-7. If the implementation does not support SCC-3, see the instructions at the top of this table.

Note 4: If SCCC-7 is supported, a list of at least one code rate is entered in the support column. If SCCC-7 is not supported, 'N/A' is entered in the support column.

Note 5: This item is not mandatory but it is recommended in ECSS-E-ST-50-01 "Convolutional coding – General", which contains a 'should' recommendation.

Note 6: If the implementation does not support the conversion from NRZ-L to NRZ-M, then support for this item is not applicable.

Note 7: Use of pseudo-randomization by a mission or project is optional. Support for item SCCC-11 implies that the implementation supports the concatenated code with and without pseudo-randomization.

A.7 Reed-Solomon code and 4D-8PSK-TCM

This section specifies the items for the Reed-Solomon code, when it is used with 4D-8PSK-TCM as specified in ECSS-E-ST-50-01, Clause "Reed-Solomon with E=8".

Table A-8: Items for Reed-Solomon code and 4D-8PSK-TCM

If Item SCC-4 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SCCD-1	(255,239) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2 and modified by clause 6.4	6.3.1 6.3.2 6.4		M	Yes [ ]	
SCCD-2	Interleaving depth, *I*, is 8	6.4.2.c		M	Yes [ ]	
SCCD-3	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
SCCD-4	Supported shortened lengths for interleaving depth *I*=8	6.3.6	Note 1	O	Values:	
SCCD-5	The ASM is attached to each R-S codeblock after the R-S encoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
SCCD-6	Pseudo-randomization of the R-S codeblock, as specified in Clause 9. The pseudo-randomization occurs after the R-S encoding, before attachment of the ASM	9.3 9.4	Note 2 Note 3	O	Yes [ ] No [ ]	

Note 1: If SCCD-3 is supported, the values entered in the support column provide details of the shortened lengths. If SCCD-3 is not supported, 'N/A' is entered in the support column.

Note 2: If the implementation supports pseudo-randomization, but the pseudo-randomizer differs from the specification in Clause 9, then 'No' should be ticked in the support column. Details can optionally be provided.

Note 3: Use of pseudo-randomization by a mission or project is optional. Support for item SCCD-6 implies that the implementation supports the (255,239) Reed-Solomon code with and without pseudo-randomization.

A.8 Turbo code

This section specifies the items for the turbo code.

Table A-9: Items for turbo code

If Item SCC-5 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SCCE-1	Turbo code as defined in clause 7.3	7.3		M	Yes [ ]	
SCCE-2	Supported nominal code rates, *r*	7.3.2.b	Note 1	M	Values:	
SCCE-3	Supported information block lengths, *k*	7.3.2.c	Note 2	M	Values:	
SCCE-4	The ASM is attached to each turbo codeblock after the turbo encoding	8.3.b 8.4 8.5.b		M	Yes [ ]	
SCCE-5	After the turbo encoder, differential encoding (i.e. NRZ-M signalling) is not used.	7.2.b	Note 3	O	Yes [ ] No [ ]	
SCCE-6	Pseudo-randomization of the turbo codeblock after the turbo encoding, before attachment of the ASM	9.3 9.4	Note 4	O	Yes [ ] No [ ]	

Note 1: A list of at least one nominal code rate is entered in the support column.

Note 2: A list of supported information block lengths is entered in the support column. If the implementation supports different information block lengths depending on the nominal code rate, then details should be provided.

Note 3: This item is not mandatory but it is recommended in ECSS-E-ST-50-01, "Turbo coding – General", which contains a 'should' recommendation.

Note 4: Use of pseudo-randomization by a mission or project is optional. Support for item SCCE-6 implies that the implementation supports the turbo code with and without pseudo-randomization.

A.9 Uncoded

This section specifies the items when no channel coding is used.

Table A-10: Items when no channel coding is used

If Item SCC-6 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SCCF-1	The ASM is attached to each transfer frame	8.3.a 8.4		M	Yes [ ]	
SCCF-2	Transfer frame lengths supported		Note 1	M	Values:	
SCCF-3	Pseudo-randomization of the transfer frame, before attachment of the ASM	9.3 9.4	Note 2	O	Yes [ ] No [ ]	

Note 1: The values entered in the support column show the frame lengths, measured in octets. The values can be entered as a range or as a list. Note that ECSS-E-ST-50-03 defines a maximum length of 2048 octets for TM Transfer Frames.

Note 2: Use of pseudo-randomization by a mission or project is optional. Support for item SCCF-3 implies that the implementation supports uncoded frames with and without pseudo-randomization.

A.10 External items

Table A-11 contains external items, related to higher layers and sublayers (how the synchronization and channel coding is used and managed) . These items can therefore be outside the scope of an implementation of the telemetry synchronization and channel coding. For conformance to the items in Table A-11, it is sufficient that the implementation does not prevent support for the items by other system elements.

Table A-11: External items

Item	Description	Clause	Notes	Status	Support (Note 1)	Remarks
SCEX-1	For the Reed-Solomon code, the interleaving depth and the length of the virtual fill on a physical channel are fixed for a mission phase.	6.3.3.b	Note 2	M	Yes [ ] N/A[ ]	
SCEX-2	If the (255,239) Reed-Solomon code specified in clause 6.4 is used, then the physical channel uses 4D-8PSK-TCM	6.4	Note 3	M	Yes [ ] N/A[ ]	
SCEX-3	The presence or absence of pseudo-randomization is fixed for a physical channel.	9.1.2.a		M	Yes [ ]	

Note 1: In this table, 'Yes' means that the implementation does not prevent support of the item.

Note 2: If the implementation does not include any support for Reed-Solomon code, then this item is not applicable.

Note 3: If the implementation does not support the Reed-Solomon code specified in clause 6.4, then this item is not applicable.

Annex B (informative) Draft PICS for ECSS-E-ST-50-01: Receiving end

B.1 Introduction to the PICS

B.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telemetry synchronization and channel coding specified in ECSS-E-ST-50-01. It is provided as information pending a formal decision by ECSS.

Sections B.2 to B.10 constitute the proforma. The normative specification is in ECSS-E-ST-50-01.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

B.1.2 Format of the PICS

B.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

B.1.2.2. Mandatory or optional status

Table B-1 shows the symbols used in the 'status' column of a PRL table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table B-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional

B.1.2.3. Support provided by the implementation

For each item, the 'support' column of a PRL table contains either:

- one or more labelled boxes, with labels as shown in Table B-2, or
- the word 'Values'.

Table B-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in B.1.2.4.	

To show the conformance of an implementation, the support column of a PRL table is completed as follows:

- If an item in the table has labelled boxes, then exactly one of the given boxes in the support column should be ticked. If none of the boxes can be ticked, then the implementation is non-conformant and exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.
- If an item has the word 'Values', then the requested values should be supplied, as indicated in a note for the item. For example, item RCCB-2 in

Table B-6 concerns the supported values for the Reed-Solomon interleaving depth, *I*. A list of at least one value for *I* should be entered in the support column.

B.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table B-9 contains items relating to turbo codes. If an implementation does not support turbo codes (item RCC-5 in Table B-4), then the N/A box at the top of Table B-9 is ticked and the rest of the table is ignored.

B.2 Identification

Table B-3 contains the items to identify the implementation.

Table B-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telemetry synchronization and channel coding specified in ECSS-E-ST-50-01		
Item	Description	Information
RI1	Date of Statement	
RI2	PICS serial number	
RI3	Supplier	
RI4	Contact details	
RI5	Implementation name	
RI6	Implementation version	
RI7	Other information for full identification, e.g. versions for operating systems	
RI8	Special configuration details	
RI9	Protocol version	
RI10	Protocol addenda and amendments	

B.3 Channel coding

This section specifies the items for the channel coding options supported by the implementation.

Table B-4: Supported channel codes

Item	Description	Clause	Notes	Status	Support	Remarks
RCC-1	Convolutional code (not concatenated with Reed-Solomon code)	5	See B.4	O	Yes [ ] No [ ]	
RCC-2	Reed-Solomon code (not concatenated with convolutional code)	6	See B.5	O	Yes [ ] No [ ]	
RCC-3	Concatenation of Reed-Solomon code and convolutional code	5 6	See B.6	O	Yes [ ] No [ ]	
RCC-4	(255,239) Reed-Solomon code and 4D-8PSK-TCM	6.4	See B.7	O	Yes [ ] No [ ]	
RCC-5	Turbo code	7	See B.8	O	Yes [ ] No [ ]	
RCC-6	No channel coding		See B.9	O	Yes [ ] No [ ]	
RCC-7	At least one of the options RCC-1 to RCC-6		Note 1	M	Yes [ ]	
Note 1: A conformant implementation supports one or more of the options RCC-1 to RCC-6.						

B.4 Convolutional code

This section specifies the items for the convolutional code, when it is not concatenated with the Reed-Solomon code.

Table B-5: Items for convolutional code

If Item RCC-1 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RCCA-1	Convolutional code rate 1/2 as defined in clause 5.3	5.3		O	Yes [ ] No [ ]	
RCCA-2	Punctured convolutional code as defined in clause 5.4	5.4		O	Yes [ ] No [ ]	
RCCA-3	At least one of the options RCCA-1 and RCCA-2		Note 1	M	Yes [ ]	
RCCA-4	Supported code rates for punctured convolutional code	5.4.a	Note 2	O	Values:	
RCCA-5	Soft bit decisions with at least 3-bit quantization are used for the decoder.	5.2.a		M	Yes [ ]	
RCCA-6	If the link uses differential encoding, then the conversion from NRZ-M to NRZ-L is performed at the output of the convolutional decoder	5.2.c	Note 3 Note 4	O	Yes [ ] No [ ] N/A[ ]	
RCCA-7	The ASM is removed from each transfer frame after the convolutional decoding	8.3.a 8.4		M	Yes [ ]	
RCCA-8	Transfer frame lengths supported		Note 5	M	Values:	
RCCA-9	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus that of the frame			M	Yes [ ]	
RCCA-10	Pseudo-derandomization of the transfer frame, after removal of the ASM	9.3 9.4	Note 6	O	Yes [ ] No [ ]	

Note 1: If the implementation supports RCC-1, then it supports one or both of the options RCCA-1 and RCCA-2. If the implementation does not support RCC-1, see the instructions at the top of this table.

Note 2: If RCCA-2 is supported, a list of at least one code rate is entered in the support column. If RCCA-2 is not supported, 'N/A' is entered in the support column.

Note 3: This item is not mandatory but it is recommended in ECSS-E-ST-50-01 "Convolutional coding – General", which contains a 'should' recommendation.

Note 4: If the implementation does not support the conversion from NRZ-L to NRZ-M, then support for this item is not applicable.

Note 5: The values entered in the support column show the frame lengths, measured in octets. The values can be entered as a range or as a list. Note that ECSS-E-ST-50-03 defines a maximum length of 2048 octets for TM Transfer Frames.

Note 6: Use of pseudo-randomization by a mission or project is optional. Support for item RCCA-10 implies that the implementation supports the convolutional code with and without pseudo-derandomization.

B.5 Reed-Solomon code

This section specifies the items for the Reed-Solomon code, when it is not concatenated with a convolutional code. This section does not apply to the Reed-Solomon code with 4D-8PSK-TCM specified in ECSS-E-ST-50-01, Clause "Reed-Solomon with E=8".

Table B-6: Items for Reed-Solomon code

If Item RCC-2 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RCCB-1	(255,223) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2	6.3.1 6.3.2		M	Yes [ ]	
RCCB-2	Supported values for the interleaving depth, *I*	6.3.3.a	Note 1	M	Values:	
RCCB-3	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
RCCB-4	Supported shortened lengths for each supported interleaving depth	6.3.6	Note 2	O	Values:	
RCCB-5	The ASM is removed from each R-S codeblock before the R-S decoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
RCCB-6	Codeblock synchronization of the Reed-Solomon decoder is achieved by synchronization of the ASM associated with each codeblock	6.3.8		M	Yes [ ]	
RCCB-7	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus the length of the transmitted codeblock	6.3.5	Note 3	M	Yes [ ]	
RCCB-8	The ambiguity between true and complemented data is resolved so that only true data is provided to the Reed-Solomon decoder.	6.3.9		M	Yes [ ]	
RCCB-9	Pseudo-derandomization of the R-S codeblock after removal of the ASM, before the R-S decoding	9.3 9.4	Note 4	O	Yes [ ] No [ ]	

Note 1: A list of at least one value for *I* is entered in the support column.

Note 2: If RCCB-3 is supported, the values entered in the support column provide details of the shortened lengths. If RCCB-3 is not supported, 'N/A' is entered in the support column.

Note 3: The length of the transmitted codeblock depends on the interleaving depth and the amount of virtual fill.

Note 4: Use of pseudo-randomization by a mission or project is optional. Support for item RCCB-9 implies that the implementation supports the Reed-Solomon code with and without pseudo-derandomization.

B.6 Concatenation of Reed-Solomon code and convolutional code

This section specifies the items for the concatenated code, consisting of the Reed-Solomon code with a convolutional code.

Table B-7: Items for concatenated R-S and convolutional code

If Item RCC-3 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RCCC-1	Convolutional code rate 1/2 as defined in clause 5.3	5.3		O	Yes [ ] No [ ]	
RCCC-2	Punctured convolutional code as defined in clause 5.4	5.4		O	Yes [ ] No [ ]	
RCCC-3	At least one of the options RCCC-1 and RCCC-2		Note 1	M	Yes [ ]	
RCCC-4	Supported code rates for punctured convolutional code	5.4.a	Note 2	O	Values:	
RCCC-5	Soft bit decisions with at least 3-bit quantization are used for the convolutional decoder.	5.2.a		M	Yes [ ]	
RCCC-6	If the link uses differential encoding, then the conversion from NRZ-M to NRZ-L is performed at the output of the convolutional decoder	5.2.c	Note 3 Note 4	O	Yes [ ] No [ ] N/A[ ]	
RCCC-7	(255,223) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2	6.3.1 6.3.2		M	Yes [ ]	
RCCC-8	Supported values for the interleaving depth, *I*	6.3.3.a	Note 5	M	Values:	
RCCC-9	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
RCCC-10	Supported shortened lengths for each supported interleaving depth	6.3.6	Note 6	O	Values:	
RCCC-11	The ASM is removed from each R-S codeblock, after the convolutional decoding, before the R-S decoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
RCCC-12	Codeblock synchronization of the Reed-Solomon decoder is achieved by synchronization of the ASM associated with each codeblock	6.3.8		M	Yes [ ]	

: Items for concatenated R-S and convolutional code (Cont'd)

If Item RCC-3 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RCCC-13	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus the length of the transmitted codeblock	6.3.5	Note 7	M	Yes [ ]	
RCCC-14	The ambiguity between true and complemented data is resolved so that only true data is provided to the Reed-Solomon decoder	6.3.9		M	Yes [ ]	
RCCC-15	Pseudo-derandomization of the R-S codeblock after removal of the ASM, before the R-S decoding	9.3 9.4	Note 8	O	Yes [ ] No [ ]	

Note 1: If the implementation supports RCC-3, then it supports one or both of the options RCCC-1 and RCCC-2. If the implementation does not support RCC-3, see the instructions at the top of this table.

Note 2: If RCCC-2 is supported, a list of at least one code rate is entered in the support column. If RCCC-2 is not supported, 'N/A' is entered in the support column.

Note 3: This item is not mandatory but it is recommended in ECSS-E-ST-50-01, "Convolutional coding – General", which contains a 'should' recommendation.

Note 4: If the implementation does not support the conversion from NRZ-L to NRZ-M, then support for this item is not applicable.

Note 5: A list of at least one value for *I* is entered in the support column.

Note 6: If RCCC-9 is supported, the values entered in the support column provide details of the shortened lengths. If RCCC-9 is not supported, 'N/A' is entered in the support column.

Note 7: The length of the transmitted codeblock depends on the interleaving depth and the amount of virtual fill.

Note 8: Use of pseudo-randomization by a mission or project is optional. Support for item RCCC-15 implies that the implementation supports the concatenated code with and without pseudo-derandomization.

B.7 Reed-Solomon code and 4D-8PSK-TCM

This section specifies the items for the Reed-Solomon code, when it is used with 4D-8PSK-TCM as specified in ECSS-E-ST-50-01 Clause "Reed-Solomon with E=8".

Table B-8: Items for Reed-Solomon code and 4D-8PSK-TCM

If Item RCC-4 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RCCD-1	(255,239) Reed-Solomon code as defined in clauses 6.3.1 and 6.3.2 and modified by clause 6.4	6.3.1 6.3.2 6.4		M	Yes [ ]	
RCCD-2	Interleaving depth, *I*, is 8	6.4.2.c		M	Yes [ ]	
RCCD-3	Shortened codeblocks and virtual fill	6.3.6		O	Yes [ ] No [ ]	
RCCD-4	Supported shortened lengths for interleaving depth *I*=8	6.3.6	Note 1	O	Values:	
RCCD-5	The ASM is removed from each R-S codeblock before the R-S decoding	8.3.a 8.4 8.5.a		M	Yes [ ]	
RCCD-6	Codeblock synchronization of the Reed-Solomon decoder is achieved by synchronization of the ASM associated with each codeblock	6.3.8		M	Yes [ ]	
RCCD-7	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus the length of the transmitted codeblock	6.3.5	Note 2	M	Yes [ ]	
RCCD-8	The ambiguity between true and complemented data is resolved so that only true data is provided to the Reed-Solomon decoder.	6.3.9		M	Yes [ ]	
RCCD-9	Pseudo-derandomization of the R-S codeblock, as specified in Clause 9. The pseudo-derandomization occurs after removal of the ASM, before the R-S decoding	9.3 9.4	Note 3 Note 4	O	Yes [ ] No [ ]	

Note 1: If RCCD-3 is supported, the values entered in the support column provide details of the shortened lengths. If RCCD-3 is not supported, 'N/A' is entered in the support column.

Note 2: The length of the transmitted codeblock depends on the amount of virtual fill.

Note 3: If the implementation supports pseudo-derandomization, but the pseudo-derandomizer differs from the specification in Clause 9, then 'No' should be ticked in the support column. Details can optionally be provided.

Note 4: Use of pseudo-randomization by a mission or project is optional. Support for item RCCD-9 implies that the implementation supports the (255,239) Reed-Solomon code with and without pseudo-derandomization.

B.8 Turbo code

This section specifies the items for the turbo code.

Table B-9: Items for turbo code

If Item RCC-5 is not supported, this table is not applicable. In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RCCE-1	Turbo code as defined in clause 7.3	7.3		M	Yes [ ]	
RCCE-2	Supported nominal code rates, *r*	7.3.2.b	Note 1	M	Values:	
RCCE-3	Supported information block lengths, *k*	7.3.2.c	Note 2	M	Values:	
RCCE-4	The ASM is acquired in the channel symbol domain (i.e. the code symbol domain of the turbo code)	8.2.3.b		M	Yes [ ]	
RCCE-5	The ASM is removed from each turbo codeblock before the turbo decoding	8.3.b 8.4 8.5.b		M	Yes [ ]	
RCCE-6	Codeblock synchronization of the turbo decoder is achieved by synchronization of the ASM associated with each turbo codeblock	7.3.7		M	Yes [ ]	
RCCE-7	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus that of the turbo codeblock.	7.3.7	Note 3	M	Yes [ ]	
RCCE-8	Pseudo-derandomization of the turbo codeblock after removal of the ASM, before the turbo decoding	9.3 9.4	Note 4	O	Yes [ ] No [ ]	

Note 1: A list of at least one nominal code rate is entered in the support column.

Note 2: A list of supported information block lengths is entered in the support column. If the implementation supports different information block lengths depending on the nominal code rate, then details should be provided.

Note 3: The length of the turbo codeblock depends on the information block length and the nominal code rate.

Note 4: Use of pseudo-randomization by a mission or project is optional. Support for item RCCE-8 implies that the implementation supports the turbo code with and without pseudo-derandomization.

B.9 Uncoded

This section specifies the items when no channel coding is used.

Table B-10: Items when no channel coding is used

If Item RCC-6 is not supported, this table is not applicable.   **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RCCF-1	The ASM is removed from each transfer frame	8.3.a 8.4		M	Yes [ ]	
RCCF-2	Transfer frame lengths supported		Note 1	M	Values:	
RCCF-3	Frame synchronizers are set to expect an ASM at a recurrence interval equal to the length of the ASM plus that of the frame			M	Yes [ ]	
RCCF-4	Pseudo-derandomization of the transfer frame, after removal of the ASM	9.3 9.4	Note 2	O	Yes [ ] No [ ]	

Note 1: The values entered in the support column show the frame lengths, measured in octets. The values can be entered as a range or as a list. Note that ECSS-E-ST-50-03 defines a maximum length of 2048 octets for TM Transfer Frames.

Note 2: Use of pseudo-randomization by a mission or project is optional. Support for item RCCE-4 implies that the implementation supports uncoded frames with and without pseudo-derandomization.

B.10 External items

Table B-11 contains external items, related to higher layers and sublayers (how the synchronization and channel coding is used and managed). These items can therefore be outside the scope of an implementation of the telemetry synchronization and channel coding. For conformance to the items in Table B-11, it is sufficient that the implementation does not prevent support for the items by other system elements.

Table B-11: External items

Item	Description	Clause	Notes	Status	Support (Note 1)	Remarks
RCEX-1	The presence or absence of pseudo-randomization is managed. The presence or absence is not signalled in the telemetry but is known a priori by the receiving system.	9.1.2.b		M	Yes [ ]	
Note 1: In this table, 'Yes' means that the implementation does not prevent support of the item.						

Annex C (informative) Draft PICS for ECSS-E-ST-50-03: Sending end

C.1 Introduction to the PICS

C.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telemetry transfer frame protocol specified in ECSS-E-ST-50-03. It is provided as information pending a formal decision by ECSS.

Sections C.2 to C.6 constitute the proforma. The normative specification is in ECSS-E-ST-50-03.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

C.1.2 Format of the PICS

C.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

C.1.2.2. Mandatory or optional status

Table C-1 shows the symbols used in the 'status' column of a PRL table.

The status of an item can be conditional on support for other items. In this case, the status column contains a numbered conditional symbol, such as C1, and the associated conditions are defined at the end of the table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table C-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional
C*n*	conditional

C.1.2.3. Support provided by the implementation

The 'support' column of a PRL table contains one or more labelled boxes. Table C-2 shows the meaning of the labels.

Table C-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
Restricted	The implementation provides restricted support
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in C.1.2.4.	

In general, if the item is mandatory, then the only box in the support column is 'Yes'. If there is a 'Restricted' box, then the table also includes a note, specifying the extra information to be supplied.

To show the conformance of an implementation, the PRL tables are completed as follows:

- For each item in the table, exactly one of the given boxes in the support column should be ticked. Any additional information or parameters should be supplied as indicated in the notes for the item.
- If none of the boxes can be ticked, then the implementation is non-conformant. Exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.

C.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table C-8 contains items relating to the extended virtual channel frame count. If an implementation does not support the extended virtual channel frame count (item STFB-9 in Table C-7), then the N/A box at the top of Table C-8 is ticked and the rest of the table is ignored.

C.2 Identification

PICS identification is shown in Table C-3.

Table C-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telemetry transfer frame protocol specified in ECSS-E-ST-50-03		
Item	**Description**	**Information**
SI1	Date of Statement	
SI2	PICS serial number	
SI3	Supplier	
SI4	Contact details	
SI5	Implementation name	
SI6	Implementation version	
SI7	Other information for full identification, e.g. versions for operating systems	
SI8	Special configuration details	
SI9	Protocol version	
SI10	Protocol addenda and amendments	

C.3 TM Transfer Frame format

C.3.1 General

This section specifies the items for the TM Transfer Frames generated by the implementation.

Table C-4: Major fields in TM Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
STF-1	Transfer Frame Primary Header	5.1.a 5.2	See C.3.2	M	Yes [ ]	
STF-2	Transfer Frame Secondary Header	5.1.a 5.3	See C.3.3	O	Yes [ ] No [ ]	
STF-3	Transfer Frame Data Field	5.1.a 5.4	See C.3.4	M	Yes [ ]	
STF-4	Operational Control Field	5.1.a 5.5	See C.3.5	O	Yes [ ] No [ ]	
STF-5	Frame Error Control Field	5.1.a 5.6	See C.3.6	O	Yes [ ] No [ ]	

C.3.2 Transfer Frame Primary Header

This section specifies the items for the Transfer Frame Primary Header of the TM Transfer Frames generated by the implementation.

Table C-5: Items for Transfer Frame Primary Header

Item	Description	Clause	Notes	Status	Support	Remarks
STFA-1	Transfer Frame Primary Header is present in all frames	5.2.1.a		M	Yes [ ]	
STFA-2	Transfer Frame Primary Header contains all specified fields in the specified bit positions	5.2		M	Yes [ ]	
STFA-3	Transfer Frame Version Number is set to ‘00’	5.2.2.2.c		M	Yes [ ]	
STFA-4	All TM Transfer Frames with the same Master Channel Identifier on a physical channel constitute a master channel	5.1.i		M	Yes [ ]	
STFA-5	A master channel has between one and eight virtual channels	5.1.j		M	Yes [ ]	

Table C-5: Items for Transfer Frame Primary Header (Cont'd)

Item	Description	Clause	Notes	Status	Support	Remarks
STFA-6	Virtual Channel Identifier provides the identification of the virtual channel to which the TM Transfer Frame belongs	5.2.3.c		M	Yes [ ]	
STFA-7	Operational Control Field Flag is set to the specified value, depending on the presence or absence of the Operational Control Field in the frame	5.2.4.c		M	Yes [ ]	
STFA-8	Master Channel Frame Count contains a sequential binary count (modulo 256) of each TM Transfer Frame transmitted within the master channel	5.2.5.c		M	Yes [ ]	
STFA-9	Unless there is a major system reset, the value of the Master Channel Frame Count is not reset before it reaches 255	5.2.5.d	Note 1	M	Yes [ ]	
STFA-10	Virtual Channel Frame Count contains a sequential binary count (modulo 256) of each TM Transfer Frame transmitted within the virtual channel of a master channel	5.2.6.c		M	Yes [ ]	
STFA-11	Unless there is a major system reset, the value of the Virtual Channel Frame Count is not reset before it reaches 255	5.2.6.d	Note 2	M	Yes [ ]	
STFA-12	Transfer Frame Secondary Header Flag is set to the specified value, depending on the presence or absence of the Transfer Frame Secondary Header in the frame	5.2.7.2.c		M	Yes [ ]	
STFA-13	The implementation can generate frames with Synchronization Flag set to '0'. In this case, the Transfer Frame Data Field contains octet-synchronized and forward-ordered packets or idle data	5.2.7.3.c	Note 3	O	Yes [ ] No [ ]	
STFA-14	The implementation can generate frames with Synchronization Flag set to '1'. In this case, formatting of the Transfer Frame Data Field is otherwise than in STFA-13	5.2.7.3.c	Note 4	O	Yes [ ] No [ ]	
STFA-15	At least one of the options STFA-13 and STFA-14	5.2.7.3.c	Note 5	M	Yes [ ]	

Note 1: Implementation to specify the conditions which cause the Master Channel Frame Count to be reset before reaching 255.

Note 2: Implementation to specify the conditions which cause the Virtual Channel Frame Count to be reset before reaching 255.

Note 3: See Table C-6 for additional items when Synchronization Flag = '0'.

Note 4: For frames with Synchronization Flag = '1', the values of the Packet Order Flag, Segment Length Identifier and First Header Pointer are undefined.

Note 5: A conformant implementation supports one or both of the options STFA-13 and STFA-14.

Table C-6: Items for Transfer Frame Primary Header when Synchronization Flag = '0'

If Item STFA-13 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFAX-1	Packet Order Flag is set to '0'	5.2.7.4.c		M	Yes [ ]	
STFAX-2	Segment Length Identifier is set to '11'	5.2.7.5.c		M	Yes [ ]	
STFAX-3	First Header Pointer contains information on the data within the Transfer Frame Data Field	5.2.7.6.c		M	Yes [ ]	
STFAX-4	If at least one packet starts in the Transfer Frame Data Field, the First Header Pointer contains the location of the first octet of the first packet that starts in the Transfer Frame Data Field. The locations of the octets in the Transfer Frame Data Field are numbered in ascending order starting with '0'.	5.2.7.6.d 5.2.7.6.e		M	Yes [ ]	
STFAX-5	If no packet starts in the Transfer Frame Data Field, the First Header Pointer is set to '11111111111'	5.2.7.6.f		M	Yes [ ]	
STFAX-6	If the Transfer Frame Data Field contains only idle data, the First Header Pointer is set to '11111111110'	5.2.7.6.g		M	Yes [ ]	

C.3.3 Transfer Frame Secondary Header

This section specifies the items for the Transfer Frame Secondary Header of the TM Transfer Frames generated by the implementation.

Table C-7: Items for frames with Transfer Frame Secondary Header

If Item STF-2 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFB-1	Presence of the Transfer Frame Secondary Header is optional	5.3.1.a	Note 1	O	Yes [ ] No [ ]	
STFB-2	Transfer Frame Secondary Header follows, without gap, the Transfer Frame Primary Header	5.3.1.a		M	Yes [ ]	
STFB-3	Presence of the Transfer Frame Secondary Header is signalled by the Transfer Frame Secondary Header Flag	5.3.1.b	See STFA-12	M	Yes [ ]	
STFB-4	Transfer Frame Secondary Header consists of an integral number of octets between a minimum of 2 octets and a maximum of 64 octets	5.3.1.c	Note 2	M	Yes [ ] Restricted [ ]	
STFB-5	Transfer Frame Secondary Header is associated with either a master channel or a virtual channel	5.3.1.d	Note 3	M	Yes [ ] Restricted [ ]	
STFB-6	Transfer Frame Secondary Header contains all specified fields in the specified positions	5.3		M	Yes [ ]	
STFB-7	Transfer Frame Secondary Header Version Number is set to '00'	5.3.2.2.c		M	Yes [ ]	
STFB-8	Value contained in the Transfer Frame Secondary Header Length is set as specified in 5.3.2.3.c	5.3.2.3.c		M	Yes [ ]	
STFB-9	Use of the Transfer Frame Secondary Header to provide an extended virtual channel frame count as specified in 5.3.4	5.3.4	Note 4 Note 5	O	Yes [ ] Restricted [ ] No [ ]	

Note 1: Support implies that the implementation can generate frames with and without the Transfer Frame Secondary Header. If not supported, then the implementation cannot generate frames without the Transfer Frame Secondary Header.

Note 2: Full support implies that the implementation supports all legal lengths for the Transfer Frame Secondary Header. If support is restricted, implementation to specify the lengths supported.

Note 3: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the Transfer Frame Secondary Header with a physical channel.

Note 4: See Table C-8 for additional items for extended virtual channel frame count.

Note 5: Implementation to specify the details if support for the item is restricted. Example: the implementation cannot support the simultaneous use of the extended virtual channel frame count on some virtual channels with other uses of the Transfer Frame Secondary Header on other virtual channels.

Table C-8: Items for frames with extended virtual channel frame count

If Item STFB-9 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFBX-1	Length of the Transfer Frame Secondary Header is 32 bits	5.3.4.2.a		M	Yes [ ]	
STFBX-2	Transfer Frame Secondary Header Data Field contains the 24-bit extension to the virtual channel frame count	5.3.4.2.b		M	Yes [ ]	
STFBX-3	The extension to the virtual channel frame count is a binary count of the roll-overs of the 8-bit value contained in the Virtual Channel Frame Count in the Transfer Frame Primary Header	5.3.4.2.c		M	Yes [ ]	
STFBX-4	Use of the extended virtual channel frame count is associated with either a master channel or a virtual channel	5.3.4.2.d	Note 1	M	Yes [ ] Restricted [ ]	
Note 1: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the extended virtual channel frame count with a physical channel.						

C.3.4 Transfer Frame Data Field

This section specifies the items for the Transfer Frame Data Field of the TM Transfer Frames generated by the implementation.

Table C-9: Items for Transfer Frame Data Field

Item	Description	Clause	Notes	Status	Support	Remarks
STFC-1	Transfer Frame Data Field is always present in a TM Transfer Frame	5.4.2.a		M	Yes [ ]	
STFC-2	Transfer Frame Data Field follows, without gap, either the Transfer Frame Secondary Header (if present) or the Transfer Frame Primary Header (otherwise)	5.4.2.b		M	Yes [ ]	
STFC-3	Length of the Transfer Frame Data Field consists of an integral number of octets and is constrained by the length of the total TM Transfer Frame	5.4.2.c		M	Yes [ ]	
STFC-4	Use of the Transfer Frame Data Field to carry asynchronously inserted data as specified in 5.4.4	5.4.4	Note 1 Note 2	O	Yes [ ] No [ ]	

Item	Description	Clause	Notes	Status	Support	Remarks
Note 1: See Table C-10 for additional items for asynchronously inserted data. Note 2: The specification in 5.4.4 is included in ECSS-E-ST-50-03 for legacy reasons.						

Table C-10: Items for frames with asynchronously inserted data as specified in clause 5.4.4

If Item STFC-4 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
STFCX-1	The stored data are in the form of standard TM Transfer Frames	5.4.4.2.a		M	Yes [ ]	
STFCX-2	At playback time, the recorded TM Transfer Frames are placed into the Transfer Frame Data Field of real-time TM Transfer Frames with the Synchronization Flag set to '1'	5.4.4.2.b		M	Yes [ ]	
STFCX-3	The asynchronous insertion can be made in either the forward or the reverse mode	5.4.4.2.c	Note 1	M	Yes [ ] Restricted [ ]	
STFCX-4	If forward insertion mode is used, then any recorded attached synchronization markers use the alternative synchronization marker pattern of hexadecimal 352EF853	5.4.4.2.d		C1	Yes [ ] N/A [ ]	
STFCX-5	A dedicated virtual channel is used for the playback data	5.4.4.2.e		M	Yes [ ]	
Note 1: Full support implies that the implementation supports forward mode and reverse mode. If support is restricted, implementation to specify the mode supported. Conditional status C1: If implementation includes support for forward mode in item STFCX-3, then mandatory, else not applicable.						

C.3.5 Operational Control Field

This section specifies the items for the Operational Control Field of the TM Transfer Frames generated by the implementation.

Table C-11: Items for frames with Operational Control Field

If Item STF-4 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFD-1	Presence of the Operational Control Field is optional	5.5.1.a	Note 1	O	Yes [ ] No [ ]	
STFD-2	Operational Control Field occupies the four octets following, without gap, the Transfer Frame Data Field	5.5.1.a		M	Yes [ ]	
STFD-3	Presence of the Operational Control Field is signalled by the Operational Control Field Flag	5.5.1.b	See STFA-7	M	Yes [ ]	
STFD-4	Operational Control Field is associated with either a master channel or a virtual channel	5.5.1.c	Note 2	M	Yes [ ] Restricted [ ]	
STFD-5	Bit 0 of the Operational Control Field contains a Type Flag which indicates the contents of the field	5.5.1.e 5.5.2.a 5.5.2.b		M	Yes [ ]	
STFD-6	Operational Control Field contains a Type-1-Report or a Type-2 Report	5.5.3.a 5.5.4.a	Note 3	M	Yes [ ] Restricted [ ]	
STFD-7	Type Flag is set to '0' if the Operational Control Field holds a Type-1-Report	5.5.2.c		C1	Yes [ ] N/A [ ]	
STFD-8	Type Flag is set to '1' if the Operational Control Field holds a Type-2-Report	5.5.2.c		C2	Yes [ ] N/A [ ]	
STFD-9	Type Flag can vary between TM Transfer Frames on the same virtual channel	5.5.2.d		C3	Yes [ ] N/A [ ]	
STFD-10	A Type-1-Report contains a Communications Link Control Word	5.5.3.b		C1	Yes [ ] N/A [ ]	
STFD-11	The first bit of a Type-2-Report (i.e. bit 1 of the Operational Control Field) indicates the use of the report	5.5.4.b		C2	Yes [ ] N/A [ ]	
STFD-12	The value of the first bit of a Type-2-Report can vary between TM Transfer Frames on the same virtual channel	5.5.4.c		C2	Yes [ ] N/A [ ]	

Note 1: Support implies that the implementation can generate frames with and without the Operational Control Field. If not supported, then the implementation cannot generate frames without the Operational Control Field.

Note 2: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the Operational Control Field with a physical channel.

Note 3: Full support implies that the implementation supports Type-1-Report and Type-2 Report. If support is restricted, implementation to specify the report type supported.

Conditional status C1: If implementation includes support for Type-1 Report in item STFD-6, then mandatory, else not applicable.

Conditional status C2: If implementation includes support for Type-2 Report in item STFD-6, then mandatory, else not applicable.

Conditional status C3: If implementation has full support for item STFD-6, then mandatory, else not applicable.

C.3.6 Frame Error Control Field

This section specifies the items for the Frame Error Control Field of the TM Transfer Frames generated by the implementation.

Table C-12: Items for frames with Frame Error Control Field

If Item STF-5 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFE-1	Presence of the Frame Error Control Field is optional	5.6.1.a	Note 1	O	Yes [ ] No [ ]	
STFE-2	Frame Error Control Field occupies the two octets following, without gap, either the Operational Control Field (if present) or the Transfer Frame Data Field (otherwise)	5.6.1.a		M	Yes [ ]	
STFE-4	Value of the Frame Error Control Field is generated as specified in the encoding procedure	5.6.2.a 5.6.2.b		M	Yes [ ]	
Note 1: Support implies that the implementation can generate frames with and without the Frame Error Control Field. If not supported, then the implementation cannot generate frames without the Frame Error Control Field.						

C.4 TM Transfer Frame length

This section specifies the items for the length of the TM Transfer Frames generated by the implementation.

Table C-13: Items for the length of the TM Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
SPL-1	TM Transfer Frame has length up to 2048 octets	5.1.b	Note 1	M	Yes [ ] Restricted [ ]	
Note 1: Full support implies the implementation can generate TM Transfer Frames of any legal length. If support is restricted, implementation to specify the frame lengths supported.						

C.5 Packet processing function

This section specifies the items for the packet processing function. The function is used when generating the Transfer Frame Data Fields of frames with Synchronization Flag set to ‘0’.

Table C-14: Items for packet processing function

If Item STFA-13 is not supported, this table is not applicable. In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SPPF-1	A packet handled by the packet processing function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.3.3.a	Note 1	M	Yes [ ] Restricted [ ]	
SPPF-2	A packet handled by the packet processing function is conformant with the definition of the related data structure	5.4.3.3.a		M	Yes [ ]	
SPPF-3	The packet processing function is conducted independently for each virtual channel	5.4.3.4.a		M	Yes [ ]	
SPPF-4	The packet processing function places packets contiguously into the Transfer Frame Data Field	5.4.3.4.b		M	Yes [ ]	
SPPF-5	If the length of a packet exceeds the available space in the Transfer Frame Data Field, the packet processing function places the remainder of the packet into the Transfer Frame Data Fields of one or more subsequent TM Transfer Frames for the same virtual channel	5.4.3.4.c		M	Yes [ ]	
SPPF-6	The packet processing function sets the First Header Pointer as specified in clause 5.2.7.6	5.4.3.4.d		M	Yes [ ]	
SPPF-7	Packets with different Packet Version Numbers can be transmitted within a virtual channel	5.4.3.4.e		C1	Yes [ ] N/A [ ]	
SPPF-8	A Transfer Frame Data Field containing only idle data can be created	5.4.3.4.f		O	Yes [ ] No [ ]	
SPPF-9	One or more idle packets can be created to fill space in a Transfer Frame Data Field	5.4.3.4.g		M	Yes [ ]	
SPPF-10	An idle packet is either an idle space packet as specified in CCSDS 133.0-B-1 clause 4.1, or a fill encapsulation packet as specified in CCSDS 133.1-B-1 clause 4.2	5.4.3.3.b	Note 2	M	Yes [ ] Restricted [ ]	

Note 1: Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.

Note 2: Full support implies the implementation can generate idle space packets and fill encapsulation packets as idle packets. If support is restricted, implementation to specify the idle packets supported.

Conditional status C1: If implementation has support for at least two Packet Version Numbers for item SPPF-1, then mandatory, else not applicable.

C.6 External items

Table C-15 contains external items, related to higher layers and sublayers (how the telemetry transfer frame protocol is used and managed) and to lower layers and sublayers (channel coding and physical channel properties). These items can therefore be outside the scope of an implementation of the telemetry transfer frame protocol. For conformance to the items in Table C-15, it is sufficient that the implementation does not prevent support for the items by other system elements.

Table C-15: External items

Item	Description	Clause	Notes	Status	Support (Note 1)	Remarks
SPEX-1	Length of TM Transfer Frame is constant for a mission phase for a physical channel	5.1.c		M	Yes [ ]	
SPEX-2	Length of TM Transfer Frame is consistent with ECSS-E-ST-50-01	5.1.d		M	Yes [ ]	
SPEX-3	TM Transfer Frames are transferred over a physical channel at a constant rate	5.1.e		M	Yes [ ]	
SPEX-4	The same telemetry channel coding options are applied to all TM Transfer Frames of a physical channel	5.1.f		M	Yes [ ]	
SPEX-5	On a physical channel which carries TM Transfer Frames, all the frames have the same Transfer Frame Version Number	5.1.k		M	Yes [ ]	
SPEX-6	Spacecraft Identifier provides the identification of the spacecraft which is associated with the data contained in the TM Transfer Frame	5.2.2.3.c		M	Yes [ ]	
SPEX-7	Spacecraft Identifier is static for a master channel throughout all mission phases	5.2.2.3.d		M	Yes [ ]	
SPEX-8	The value of the Operational Control Field Flag is static within the associated master channel or virtual channel throughout a mission phase	5.2.4.d		M	Yes [ ]	
SPEX-9	Value of the Transfer Frame Secondary Header Flag is static within the associated master channel or virtual channel throughout a mission phase	5.2.7.2.d 5.2.7.2.e		M	Yes [ ]	
SPEX-10	Value of the Synchronization Flag is static within a virtual channel throughout a mission phase	5.2.7.3.d		M	Yes [ ]	

Table C-15: External items (Cont'd)

Item	Description	Clause	Notes	Status	Support (Note 1)	Remarks
SPEX-11	If the Transfer Frame Secondary Header is used, then it has fixed length within the associated channel throughout a mission phase	5.3.1.e		C1	Yes [ ] N/A [ ]	
SPEX-12	If the Transfer Frame Secondary Header is used, then it carries fixed length data defined at mission level	5.3.1.g		C1	Yes [ ] N/A [ ]	
SPEX-13	If the Transfer Frame Secondary Header is used, then the value of the Transfer Frame Secondary Header Length is static within the associated channel throughout a mission phase	5.3.2.3.d		C1	Yes [ ] N/A [ ]	
SPEX-14	If the Transfer Frame Secondary Header is used, then the Transfer Frame Secondary Header Data Field contains the Transfer Frame Secondary Header data	5.3.3.c		C1	Yes [ ] N/A [ ]	
SPEX-15	If the extended virtual channel frame count is used, then the use is static within the associated channel throughout a mission phase	5.3.4.2.e		C2	Yes [ ] N/A [ ]	
SPEX-16	If the Operational Control Field is used, then it occurs within every TM Transfer Frame transmitted through the associated channel throughout a mission phase	5.5.1.d		C3	Yes [ ] N/A [ ]	
SPEX-17	If the TM Transfer Frame is not Reed-Solomon encoded, then the Frame Error Control Field is present	5.6.1.b	Note 2	C4	Yes [ ] N/A [ ]	
SPEX-18	If the Frame Error Control Field is used, then it occurs within every TM Transfer Frame transmitted within the same physical channel throughout a mission phase	5.6.1.c		C4	Yes [ ] N/A [ ]	

Note 1: In this table, Yes means that the implementation does not prevent support of the item.

Note 2: If the TM Transfer Frame is Reed-Solomon encoded, then the Frame Error Control Field is optional.

Conditional status C1: If implementation has support for item STF-2, then mandatory, else not applicable.

Conditional status C2: If implementation has support for item STFB-9, then mandatory, else not applicable.

Conditional status C3: If implementation has support for item STF-4, then mandatory, else not applicable.

Conditional status C4: If implementation has support for item STF-5, then mandatory, else not applicable.

Annex D (informative) Draft PICS for ECSS-E-ST-50-03: Receiving end

D.1 Introduction to the PICS

D.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telemetry transfer frame protocol specified in ECSS-E-ST-50-03. It is provided as information pending a formal decision by ECSS.

Sections D.2 to D.5 constitute the proforma. The normative specification is in ECSS-E-ST-50-03.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

D.1.2 Format of the PICS

D.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

D.1.2.2. Mandatory or optional status

Table D-1 shows the symbols used in the 'status' column of a PRL table.

The status of an item can be conditional on support for other items. In this case, the status column contains a numbered conditional symbol, such as C1, and the associated conditions are defined at the end of the table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table D-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional
C*n*	conditional

D.1.2.3. Support provided by the implementation

The 'support' column of a PRL table contains one or more labelled boxes. Table D-2 shows the meaning of the labels.

Table D-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
Restricted	The implementation provides restricted support
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in D.1.2.4.	

In general, if the item is mandatory, then the only box in the support column is 'Yes'. If there is a 'Restricted' box, then the table also includes a note, specifying the extra information to be supplied.

To show the conformance of an implementation, the PRL tables are completed as follows:

- For each item in the table, exactly one of the given boxes in the support column should be ticked. Any additional information or parameters should be supplied as indicated in the notes for the item.
- If none of the boxes can be ticked, then the implementation is non-conformant. Exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.

D.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table D-8 contains items relating to the extended virtual channel frame count. If an implementation does not support the extended virtual channel frame count (item RTFB-9 in Table D-7), then the N/A box at the top of Table D-8 is ticked and the rest of the table is ignored.

D.2 Identification

Table D-3 shows the items to identify the implementation.

Table D-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telemetry transfer frame protocol specified in ECSS-E-ST-50-03		
Item	**Description**	**Information**
RI1	Date of Statement	
RI2	PICS serial number	
RI3	Supplier	
RI4	Contact details	
RI5	Implementation name	
RI6	Implementation version	
RI7	Other information for full identification, e.g. versions for operating systems	
RI8	Special configuration details	
RI9	Protocol version	
RI10	Protocol addenda and amendments	

D.3 TM Transfer Frame format

D.3.1 General

This section specifies the items for the TM Transfer Frames handled by the implementation.

Table D-4: Major fields in TM Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
RTF-1	Transfer Frame Primary Header	5.1.a 5.2	See D.3.2	M	Yes [ ]	
RTF-2	Transfer Frame Secondary Header	5.1.a 5.3	See D.3.3	O	Yes [ ] No [ ]	
RTF-3	Transfer Frame Data Field	5.1.a 5.4	See D.3.4	M	Yes [ ]	
RTF-4	Operational Control Field	5.1.a 5.5	See D.3.5	O	Yes [ ] No [ ]	
RTF-5	Frame Error Control Field	5.1.a 5.6	See D.3.6	O	Yes [ ] No [ ]	

D.3.2 Transfer Frame Primary Header

This section specifies the items for the Transfer Frame Primary Header of the TM Transfer Frames handled by the implementation.

Table D-5: Items for Transfer Frame Primary Header

Item	Description	Clause	Notes	Status	Support	Remarks
RTFA-1	Transfer Frame Primary Header is present in all frames	5.2.1.a		M	Yes [ ]	
RTFA-2	Transfer Frame Primary Header fields are found in the specified bit positions	5.2		M	Yes [ ]	
RTFA-3	Transfer Frame Version Number is set to '00'	5.2.2.2.c		M	Yes [ ]	
RTFA-4	All TM Transfer Frames with the same Master Channel Identifier on a physical channel constitute a master channel	5.1.i		M	Yes [ ]	
RTFA-5	A master channel has between one and eight virtual channels	5.1.j		M	Yes [ ]	
RTFA-6	Virtual Channel Identifier provides the identification of the virtual channel to which the TM Transfer Frame belongs	5.2.3.c		M	Yes [ ]	
RTFA-7	Operational Control Field Flag indicates the presence or absence of the Operational Control Field in the frame	5.2.4.c		M	Yes [ ]	
RTFA-8	Master Channel Frame Count contains a sequential binary count (modulo 256) of each TM Transfer Frame transmitted within the master channel	5.2.5.c		M	Yes [ ]	
RTFA-9	Virtual Channel Frame Count contains a sequential binary count (modulo 256) of each TM Transfer Frame transmitted within the virtual channel of a master channel	5.2.6.c		M	Yes [ ]	
RTFA-10	Transfer Frame Secondary Header Flag indicates the presence or absence of the Transfer Frame Secondary Header in the frame	5.2.7.2.c		M	Yes [ ]	
RTFA-11	The implementation can handle frames with Synchronization Flag set to '0'. In this case, the Transfer Frame Data Field contains octet-synchronized and forward-ordered packets or idle data	5.2.7.3.c	Note 1	O	Yes [ ] No [ ]	
RTFA-12	The implementation can handle frames with Synchronization Flag set to '1'. In this case, formatting of the Transfer Frame Data Field is otherwise than in RTFA-11	5.2.7.3.c	Note 2	O	Yes [ ] No [ ]	
RTFA-13	At least one of the options RTFA-11 and RTFA-12	5.2.7.3.c	Note 3	M	Yes [ ]	

Note 1: See Table D-6 for additional items when Synchronization Flag = '0'.

Note 2: For frames with Synchronization Flag = '1', the values of the Packet Order Flag, Segment Length Identifier and First Header Pointer are undefined.

Note 3: A conformant implementation supports one or both of the options RTFA-11 and RTFA-12.

Table D-6: Items for Transfer Frame Primary Header when Synchronization Flag = ‘0’

If Item RTFA-11 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RTFAX-1	First Header Pointer contains information on the data within the Transfer Frame Data Field	5.2.7.6.c		M	Yes [ ]	
RTFAX-2	If at least one packet starts in the Transfer Frame Data Field, the First Header Pointer contains the location of the first octet of the first packet that starts in the Transfer Frame Data Field. The locations of the octets in the Transfer Frame Data Field are numbered in ascending order starting with ‘0’.	5.2.7.6.d 5.2.7.6.e		M	Yes [ ]	
RTFAX-3	If the First Header Pointer is set to ‘11111111111’, no packet starts in the Transfer Frame Data Field	5.2.7.6.f		M	Yes [ ]	
RTFAX-4	If the First Header Pointer is set to ‘11111111110’, the Transfer Frame Data Field contains only idle data	5.2.7.6.g		M	Yes [ ]	

D.3.3 Transfer Frame Secondary Header

This section specifies the items for the Transfer Frame Secondary Header of the TM Transfer Frames handled by the implementation.

Table D-7: Items for frames with Transfer Frame Secondary Header

If Item RTF-2 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RTFB-1	Presence of the Transfer Frame Secondary Header is optional	5.3.1.a	Note 1	O	Yes [ ] No [ ]	
RTFB-2	Transfer Frame Secondary Header follows, without gap, the Transfer Frame Primary Header	5.3.1.a		M	Yes [ ]	
RTFB-3	Presence of the Transfer Frame Secondary Header is signalled by the Transfer Frame Secondary Header Flag	5.3.1.b	See RTFA-10	M	Yes [ ]	
RTFB-4	Transfer Frame Secondary Header consists of an integral number of octets between a minimum of 2 octets and a maximum of 64 octets	5.3.1.c	Note 2	M	Yes [ ] Restricted [ ]	
RTFB-5	Transfer Frame Secondary Header is associated with either a master channel or a virtual channel	5.3.1.d	Note 3	M	Yes [ ] Restricted [ ]	
RTFB-7	Transfer Frame Secondary Header Version Number is set to '00'	5.3.2.2.c		M	Yes [ ]	
RTFB-8	Value contained in the Transfer Frame Secondary Header Length is set as specified in 5.3.2.3.c	5.3.2.3.c		M	Yes [ ]	
RTFB-9	Use of the Transfer Frame Secondary Header to provide an extended virtual channel frame count as specified in 5.3.4	5.3.4	Note 4 Note 5	O	Yes [ ] Restricted [ ] No [ ]	

Note 1: Support implies that the implementation can handle frames with and without the Transfer Frame Secondary Header. If not supported, then the implementation cannot handle frames without the Transfer Frame Secondary Header.

Note 2: Full support implies that the implementation supports all legal lengths for the Transfer Frame Secondary Header. If support is restricted, implementation to specify the lengths supported.

Note 3: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the Transfer Frame Secondary Header with a physical channel.

Note 4: See Table D-8 for additional items for extended virtual channel frame count.

Note 5: Implementation to specify the details if support for the item is restricted. Example: the implementation cannot support the simultaneous use of the extended virtual channel frame count on some virtual channels with other uses of the Transfer Frame Secondary Header on other virtual channels.

Table D-8: Items for frames with extended virtual channel frame count

If Item RTFB-9 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RTFBX-1	The extension to the virtual channel frame count is a binary count of the roll-overs of the 8-bit value contained in the Virtual Channel Frame Count in the Transfer Frame Primary Header	5.3.4.2.c		M	Yes [ ]	
RTFBX-2	Use of the extended virtual channel frame count is associated with either a master channel or a virtual channel	5.3.4.2.d	Note 1	M	Yes [ ] Restricted [ ]	
Note 1: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the extended virtual channel frame count with a physical channel.						

D.3.4 Transfer Frame Data Field

This section specifies the items for the Transfer Frame Data Field of the TM Transfer Frames handled by the implementation.

Table D-9: Items for Transfer Frame Data Field

Item	Description	Clause	Notes	Status	Support	Remarks
RTFC-1	Transfer Frame Data Field is always present in a TM Transfer Frame	5.4.2.a		M	Yes [ ]	
RTFC-2	Transfer Frame Data Field follows, without gap, either the Transfer Frame Secondary Header (if present) or the Transfer Frame Primary Header (otherwise)	5.4.2.b		M	Yes [ ]	
RTFC-3	Use of the Transfer Frame Data Field to carry asynchronously inserted data as specified in 5.4.4	5.4.4	Note 1 Note 2	O	Yes [ ] No [ ]	
Note 1: See Table D-10 for additional items for asynchronously inserted data. Note 2: The specification in 5.4.4 is included in ECSS-E-ST-50-03 for legacy reasons.						

Table D-10: Items for frames with asynchronously inserted data as specified in clause 5.4.4

If Item RTFC-3 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RTFCX-1	The asynchronous insertion can be made in either the forward or the reverse mode	5.4.4.2.c	Note 1	M	Yes [ ] Restricted [ ]	
RTFCX-2	If forward insertion mode is used, then any recorded attached synchronization markers use the alternative synchronization marker pattern of hexadecimal 352EF853	5.4.4.2.d		C1	Yes [ ] N/A [ ]	
RTFCX-3	The real-time virtual channel used for the playback data is processed and its contents stored away for later off-line processing	5.4.4.2.f		M	Yes [ ]	
RTFCX-4	In the later off-line processing, the recorded TM Transfer Frames are retrieved in the correct, forward-justified direction and processed as for a real-time physical channel	5.4.4.2.g		M	Yes [ ]	
RTFCX-5	Any Communications Link Control Word extracted from the Operational Control Field of a recorded TM Transfer Frame is not processed by the real-time telecommand system	5.4.4.2.h		M	Yes [ ]	
Note 1: Full support implies that the implementation supports forward mode and reverse mode. If support is restricted, implementation to specify the mode supported. Conditional status C1: If implementation includes support for forward mode in item RTFCX-1, then mandatory, else not applicable.						

D.3.5 Operational Control Field

This section specifies the items for the Operational Control Field of the TM Transfer Frames handled by the implementation.

Table D-11: Items for frames with Operational Control Field

If Item RTF-4 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RTFD-1	Presence of the Operational Control Field is optional	5.5.1.a	Note 1	O	Yes [ ] No [ ]	
RTFD-2	Operational Control Field occupies the four octets following, without gap, the Transfer Frame Data Field	5.5.1.a		M	Yes [ ]	
RTFD-3	Operational Control Field is associated with either a master channel or a virtual channel	5.5.1.c	Note 2	M	Yes [ ] Restricted [ ]	
RTFD-4	Operational Control Field can contain a Type-1-Report or a Type-2 Report, as signalled by the Type Flag	5.5.3.a 5.5.4.a 5.5.2.c	Note 3	M	Yes [ ] Restricted [ ]	
RTFD-5	Type Flag can vary between TM Transfer Frames on the same virtual channel	5.5.2.d		C1	Yes [ ] N/A [ ]	
RTFD-6	The value of the first bit of a Type-2-Report can vary between TM Transfer Frames on the same virtual channel	5.5.4.c		C2	Yes [ ] N/A [ ]	

Note 1: Support implies that the implementation can handle frames with and without the Operational Control Field. If not supported, then the implementation cannot handle frames without the Operational Control Field.

Note 2: Implementation to specify the details if support for the item is restricted. Example: the implementation can only associate the Operational Control Field with a physical channel.

Note 3: Full support implies that the implementation supports Type-1-Report and Type-2 Report. If support is restricted, implementation to specify the report type supported.

Conditional status C1: If implementation has full support for item RTFD-4, then mandatory, else not applicable.

Conditional status C2: If implementation includes support for Type-2 Report in item RTFD-4, then mandatory, else not applicable.

D.3.6 Frame Error Control Field

This section specifies the items for the Frame Error Control Field of the TM Transfer Frames handled by the implementation.

Table D-12: Items for frames with Frame Error Control Field

If Item RTF-5 is not supported, this table is not applicable. In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RTFE-1	Presence of the Frame Error Control Field is optional	5.6.1.a	Note 1	O	Yes [ ] No [ ]	
RTFE-2	Frame Error Control Field occupies the two octets following, without gap, either the Operational Control Field (if present) or the Transfer Frame Data Field (otherwise)	5.6.1.a		M	Yes [ ]	
RTFE-3	Frame Error Control Field is decoded as specified in the decoding procedure	5.6.3.a		M	Yes [ ]	
RTFE-4	Frame Error Control Field is not used for error correction	5.6.3.b		M	Yes [ ]	
Note 1: Support implies that the implementation can handle frames with and without the Frame Error Control Field. If not supported, then the implementation cannot handle frames without the Frame Error Control Field.						

D.4 TM Transfer Frame length

This section specifies the items for the length of the TM Transfer Frames handled by the implementation.

Table D-13: Items for the length of the TM Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
RPL-1	TM Transfer Frame has length up to 2048 octets	5.1.b	Note 1	M	Yes [ ] Restricted [ ]	
Note 1: Full support implies the implementation can handle TM Transfer Frames of any legal length. If support is restricted, implementation to specify the frame lengths supported.						

D.5 Packet extraction function

This section specifies the items for the packet extraction function. The function is used when handling the Transfer Frame Data Fields of frames which have the Synchronization Flag set to '0'.

Table D-14: Items for packet extraction function

If Item RTFA-11 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RPEF-1	A packet handled by the packet extraction function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.3.3.a	Note 1	M	Yes [ ] Restricted [ ]	
RPEF-2	Packets with different Packet Version Numbers can be handled within a virtual channel	5.4.3.4.e		C1	Yes [ ] N/A [ ]	
RPEF-3	The packet extraction function is conducted independently for each virtual channel	5.4.3.5.a		M	Yes [ ]	
RPEF-4	The packet extraction function can handle the First Header Pointer values specified in clause 5.2.7.6	5.4.3.5.b		M	Yes [ ]	
RPEF-5	The packet extraction function extracts the packets from a Transfer Frame Data Field using the value of the First Header Pointer together with the values contained in the length fields of the packet headers	5.4.3.5.b		M	Yes [ ]	
RPEF-6	A Transfer Frame Data Field containing idle data is discarded	5.4.3.5.c		M	Yes [ ]	
RPEF-7	Any idle packets extracted from Transfer Frame Data Fields are discarded	5.4.3.5.d		M	Yes [ ]	
RPEF-8	An idle packet is either an idle space packet as specified in CCSDS 133.0-B-1 clause 4.1, or a fill encapsulation packet as specified in CCSDS 133.1-B-1 clause 4.2	5.4.3.3.b	Note 2	M	Yes [ ] Restricted [ ]	

Note 1: Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.

Note 2: Full support implies the implementation can handle idle space packets and fill encapsulation packets as idle packets. If support is restricted, implementation to specify the idle packets supported.

Conditional status C1: If implementation has support for at least two Packet Version Numbers for item RPEF-1, then mandatory, else not applicable.

Annex E (informative) Draft PICS for ECSS-E-ST-50-04: Sending end

E.1 Introduction to the PICS

E.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telecommand protocols, synchronization and channel coding specified in ECSS-E-ST-50-04. It is provided as information pending a formal decision by ECSS.

Sections E.2 to E.7 constitute the proforma. The normative specification is in ECSS-E-ST-50-04.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

E.1.2 Format of the PICS

E.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

E.1.2.2. Mandatory or optional status

Table E-1 shows the symbols used in the 'status' column of a PRL table.

The status of an item can be conditional on support for other items. In this case, the status column contains a numbered conditional symbol, such as C1, and the associated conditions are defined at the end of the table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table E-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional
C*n*	conditional

E.1.2.3. Support provided by the implementation

The 'support' column of a PRL table contains one or more labelled boxes. Table E-2 shows the meaning of the labels.

Table E-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
Restricted	The implementation provides restricted support
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in E.1.2.4.	

In general, if the item is mandatory, then the only box in the support column is 'Yes'. If there is a 'Restricted' box, then the table also includes a note, specifying the extra information to be supplied.

To show the conformance of an implementation, the PRL tables are completed as follows:

- For each item in the table, exactly one of the given boxes in the support column should be ticked. Any additional information or parameters should be supplied as indicated in the notes for the item.
- If none of the boxes can be ticked, then the implementation is non-conformant. Exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.

E.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table E-7 contains items relating to the segmentation function. If an implementation does not support the segmentation function (item SSM-1 in Table E-5), then the N/A box at the top of Table E-7 is ticked and the rest of the table is ignored.

E.2 Identification

Table E-3 shows the items to identify the implementation.

Table E-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a sending end implementation of the telecommand protocols specified in ECSS-E-ST-50-04		
Item	**Description**	**Information**
SI1	Date of Statement	
SI2	PICS serial number	
SI3	Supplier	
SI4	Contact details	
SI5	Implementation name	
SI6	Implementation version	
SI7	Other information for full identification, e.g. versions for operating systems	
SI8	Special configuration details	
SI9	Protocol version	
SI10	Protocol addenda and amendments	

E.3 Supported layers and sublayers

This section specifies the items for the layers and sublayers supported by the sending-end implementation.

Table E-4: Layers and sublayers supported

Item	Description	Clause	Notes	Status	Support	Remarks
SLAY-1	Segmentation sublayer	5	See E.4	O	Yes [ ] No [ ]	
SLAY-2	Transfer sublayer	6 7	See E.5	O	Yes [ ] No [ ]	
SLAY-3	Synchronization and channel coding sublayer	8	See E.6	O	Yes [ ] No [ ]	
SLAY-4	Physical layer elements within the scope of ECSS-E-ST-50-04	9	See E.7	O	Yes [ ] No [ ]	
SLAY-5	At least one of the options SLAY-1 to SLAY-4		Note 1	M	Yes [ ]	
Note 1: A conformant implementation supports one or more of the options SLAY-1 to SLAY-4.						

If the implementation does not include the segmentation sublayer (item SLAY-1), then section E.4 does not apply. In this case, leave all tables and boxes in section E.4 empty and tick here ……………………………………. N/A [ ]

E.4 Segmentation sublayer

E.4.1 Introduction

Section E.4 contains the PICS tables for the segmentation sublayer of a sending-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the segmentation sublayer, then all tables and boxes in section E.4 should be ignored. Table E-4 contains the items for the layers and sublayers supported by the implementation.

E.4.2 Elements of the segmentation sublayer

Table E-5 shows the main items for segmentation sublayer.

Table E-5: Main items for segmentation sublayer

Item	Description	Clause	Notes	Status	Support	Remarks
SSM-1	Segmentation function	5.5	See E.4.4	O	Yes [ ] No [ ]	
SSM-2	MAP multiplexing	5.6	See E.4.7	O	Yes [ ] No [ ]	
SSM-3	Blocking function for packets	5.4	See E.4.5, E.4.6	O	Yes [ ] No [ ]	
SSM-4	At least one of the options SSM-1 to SSM-3		Note 1	M	Yes [ ]	
SSM-5	TC Segment data structure, including support for virtual channels that carry TC Segments	5.2 5.1	See E.4.3 Note 2	C1	Yes [ ] No [ ]	
SSM-6	Support for virtual channels that do not carry TC Segments	5.1	Note 2	C2	Yes [ ] No [ ]	
SSM-7	Within a physical channel, there can be some virtual channels that carry TC Segments and some that do not carry TC Segments	5.1		C3	Yes [ ] N/A[ ]	
SSM-8	Transfer notification	5.3	See E.4.8	M	Yes [ ]	

Note 1: A conformant implementation of the segmentation sublayer supports one or more of the options SSM-1 to SSM-3.

Note 2: Each data unit generated by the segmentation sublayer belongs to a virtual channel. Within a virtual channel:
- all the data units are TC Segments, or,
- none of the data units are TC Segments.

Conditional status C1: If implementation includes support for item SSM-1 or item SSM-2, then mandatory, else optional.

Conditional status C2: If implementation does not include support for item SSM-5, then mandatory, else optional.

Conditional status C3: If implementation includes support for items SSM-5 and SSM-6, then mandatory, else not applicable.

E.4.3 TC Segments

The TC Segment contains fields to support the segmentation and MAP multiplexing functions of the segmentation sublayer. This section specifies the items for the TC Segment data structures generated by the implementation.

Table E-6: Items for TC Segment data structure

If Item SSM-5 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SSTC-1	Length of a TC Segment is an integer number of octets	5.2.1.b		M	Yes [ ]	
SSTC-2	Length of a TC Segment does not exceed the maximum length accepted by the next lower sublayer below the segmentation sublayer.	5.2.1.a	Note 1	M	Yes [ ]	
SSTC-3	Lengths of the TC Segments can vary	5.2.1.a	Note 2	M	Yes [ ]	
SSTC-4	The implementation can generate TC Segments with lengths in the range of 2 to 1017 octets, corresponding to TC Transfer Frame lengths up to 1024 octets	6.2.2.8	Note 3 Note 4 Note 5	M	Yes [ ] Restricted [ ]	
SSTC-5	TC Segments do not share a virtual channel with other data structures	5.2.1.d	Note 6	M	Yes [ ]	
SSTC-6	A TC Segment consists of a 1-octet Segment Header followed by a variable-length Segment Data Field	5.2.1.c	Note 3	M	Yes [ ]	
SSTC-7	Segment Header contains both specified fields in the specified positions	5.2.2		M	Yes [ ]	
SSTC-8	Sequence Flags field provides information on the segmentation status of the data in the TC Segment	5.2.2.2.c		M	Yes [ ]	
SSTC-9	MAP Identifier provides the identification of the MAP to which the TC Segment belongs	5.2.2.3.c		M	Yes [ ]	
SSTC-10	If MAP multiplexing is not in use for a virtual channel, then all TC Segments on the virtual channel have the same value for the MAP Identifier.	5.2.2.3.d		M	Yes [ ]	

Note 1: The maximum length can vary for different virtual channels.

Note 2: A conformant implementation supports the generation of variable-length TC Segments.

Note 3: If the implementation supports the packet assembly controller (see Table E-8), the implementation can also generate TC Segments with a length of 1 octet. In a 1-octet TC Segment, the Segment Data Field is absent.

Note 4: Full support implies that the implementation supports all values in the range. If the implementation is restricted to TC Segment lengths up to 239 or 241 octets, then the "Restricted" box should be ticked. Any other restrictions make the implementation non-conformant.

Note 5: If the next lower sublayer below the segmentation sublayer is the transfer sublayer, then the length of the Transfer Frame Data Field of the TC Transfer Frame is the same as the length of the TC Segment.

Note 6: See Note 2 of Table E-5.

E.4.4 Segmentation

This section specifies the items for segmentation. At the sending end, segmentation is performed by the segmenting function.

Table E-7: Items for segmenting function

If Item SSM-1 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SSSG-1	Segmenting function is conducted independently for each MAP	5.5.2.a		M	Yes [ ]	
SSSG-2	Within a virtual channel, there can be some MAPs with the segmenting function enabled and some MAPs with the segmenting function not enabled	5.5		M	Yes [ ]	
SSSG-3	Segmenting function segments a user data unit as specified in clause 5.5.2.b	5.5.2.b		M	Yes [ ]	
SSSG-4	User data units input to the segmenting function can be any variable-length octet-aligned data units	5.2.1	Note 1 Note 2	M	Yes [ ]	
SSSG-6	Support for the sending-end for the packet assembly controller (PAC) defined in clause 5.5.4	5.5.4	Note 3	O	Yes [ ] No [ ]	

Note 1: A user data unit input to the segmentation sublayer from the next higher layer or sublayer is a variable-length octet-aligned data unit. Support implies that the segmenting function can handle any type of user data unit input to the segmentation sublayer.

Note 2: An implementation can set an upper limit for the length of a user data unit handled by the segmenting function. The upper limit should be specified.

Note 3: See Table E-8 for additional items when the packet assembly controller is supported.

Table E-8: Items for packet assembly controller (PAC)

If Item SSSG-4 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SSPA-1	Use of the PAC is optional		Note 1	O	Yes [ ] No [ ]	
SSPA-2	PAC uses a pair of MAP identifier values as specified in clause 5.5.4.2	5.5.4.2		M	Yes [ ]	
SSPA-3	A control segment (MAP reset) generated by the implementation conforms to the specification in clause 5.5.4.6	5.5.4.6	Note 2	M	Yes [ ]	
Note 1:	Support implies that the implementation can support segmentation for MAPs with the PAC and for MAPs without the PAC. If not supported, then the implementation cannot support segmentation for MAPs without the PAC.					
Note 2:	ECSS-E-ST-50-04 specifies the contents of a PAC status report but does not specify how the report is transported to the appropriate entity at the sending end. It also does not specify any resulting behaviour at the sending end, such as the decision to send a control segment.					

E.4.5 Blocking function on a virtual channel that carries TC Segments

This section specifies the items for the blocking function, when it is being used on a virtual channel that carries TC Segments. In this case, the blocking function operates within a MAP.

Table E-9: Items for blocking function on a MAP

If Item SSM-3 or item SSM-5 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SSBM-1	Blocking function is conducted independently for each MAP	5.4.2.a		M	Yes [ ]	
SSBM-2	A virtual channel can have some MAPs where the blocking function is enabled and some MAPs where the blocking function is not enabled	5.4.2.c		M	Yes [ ]	
SSBM-3	If the blocking function is enabled for a MAP, then all user data units for that MAP are packets with the specified packet properties	5.4.2.d		M	Yes [ ]	
SSBM-4	A packet input to the blocking function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.4	Note 1	M	Yes [ ] Restricted [ ]	
SSBM-5	Blocking function blocks packets as specified in clause 5.4.5	5.4.5		M	Yes [ ]	
SSBM-6	The length of the data unit resulting from blocking does not exceed the maximum length of the Segment Data Field of a TC Segment.	5.4.2.b		M	Yes [ ]	
SSBM-7	If the blocking function blocks multiple packets into a data unit, then the Sequence Flags of the TC Segment which carries the data unit are set to indicate no segmentation	5.4.2.e	Note 2	M	Yes [ ]	
SSBM-8	Mechanisms for ensuring transmission of all packets, and any additional constraints	5.4.1	Note 3	O	Yes [ ] No [ ]	
Note 1:	Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.					
Note 2:	A data unit containing multiple packets is not divided into multiple segments by the segmenting function					
Note 3:	The mechanisms and additional constraints are implementation dependent. For example, an implementation can set a limit for the delay imposed on a packet while waiting for more packets to be blocked with it. Details should be provided.					

E.4.6 Blocking function on a virtual channel that does not carry TC Segments

This section specifies the items for the blocking function, when it is being used on a virtual channel that does not carry TC Segments. In this case, the blocking function operates within a virtual channel.

Table E-10: Items for blocking function on a virtual channel

If Item SSM-3 or item SSM-6 is not supported, this table is not applicable. In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SSBL-1	Blocking function is conducted independently for each virtual channel	5.4.3.a		M	Yes [ ]	
SSBL-2	A physical channel can have some virtual channels where the blocking function is enabled and some virtual channels where the blocking function is not enabled	5.4.3.c		M	Yes [ ]	
SSBL-3	If the blocking function is enabled for a virtual channel, then all user data units for that virtual channel are packets with the specified packet properties	5.4.3.d		M	Yes [ ]	
SSBL-4	A packet input to the blocking function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.4	Note 1	M	Yes [ ] Restricted [ ]	
SSBL-5	Blocking function blocks packets as specified in clause 5.4.5	5.4.5		M	Yes [ ]	
SSBL-6	The length of the data unit resulting from blocking does not exceed the maximum length of the data units accepted by the next lower sublayer below the segmentation sublayer	5.4.2.b	Note 2	M	Yes [ ]	
SSBL-7	Mechanisms for ensuring transmission of all packets, and any additional constraints	5.4.1	Note 3	O	Yes [ ] No [ ]	

Note 1: Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.

Note 2: The maximum length can vary for different virtual channels.

Note 3: The mechanisms and additional constraints are implementation dependent. For example, an implementation can set a limit for the delay imposed on a packet while waiting for more packets to be blocked with it. Details should be provided.

E.4.7 MAP multiplexing

TC Segments support multiple MAPs on a virtual channel. The MAP multiplexing takes place within the segmentation sublayer. This section specifies the items for the MAP multiplexing.

Table E-11: Items for MAP multiplexing

If Item SSM-2 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SSMX-1	The multiplexer selects a TC Segment depending on the algorithm in use	5.6	Note 1	M	Yes [ ]	
Note 1:	The implementation should supply details of the multiplexing algorithms supported and the means for setting and selecting priorities where applicable.					

E.4.8 Transfer notification

The sequence-controlled service of the COP-1 protocol of the transfer sublayer uses acknowledgements to obtain positive confirmation that data units have arrived at the transfer layer at the receiving end. The segmentation sublayer at the sending end takes the information obtained from COP-1 confirm responses and makes it available to the next higher layer or sublayer. This section specifies the items for the transfer notifications.

Table E-12: Items for transfer notification

Item	Description	Clause	Notes	Status	Support	Remarks
SSTN-1	If a user data unit of the segmentation sublayer is transferred using the sequence-controlled service, then the segmentation sublayer at the sending end provides the next higher layer or sublayer with information about the delivery of the user data unit as specified in clause 5.3.2	5.3.2		M	Yes [ ]	

If the implementation does not include the transfer sublayer (item SLAY-2), then section E.5 does not apply. In this case, leave all tables and boxes in section E.5 empty and tick here ……………………………………………….. N/A [ ]

E.5 Transfer sublayer

E.5.1 Introduction

Section E.5 contains the PICS tables for the transfer sublayer of a sending-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the transfer sublayer, then all tables and boxes in section E.5 should be ignored. Table E-4 contains the items for the layers and sublayers supported by the implementation.

E.5.2 Elements of the transfer sublayer

Table E-13 shows the main items for transfer sublayer.

Table E-13: Main items for transfer sublayer

Item	Description	Clause	Notes	Status	Support	Remarks
STM-1	TC Transfer Frame data structure	6.2	See E.5.3	M	Yes [ ]	
STM-2	CLCW (communications link control word) data structure	6.3	See E.5.4	M	Yes [ ]	
STM-3	FOP-1 (sending end of COP-1 protocol)	7	See E.5.5 Note 1	M	Yes [ ] Restricted [ ]	
STM-4	Frame header procedure	6.4	See E.5.6	M	Yes [ ]	
STM-5	Frame error control procedure	6.5	See E.5.6	M	Yes [ ]	
STM-6	Virtual channel multiplexing	6.9	See E.5.6 and E.5.7	M	Yes [ ]	
STM-7	Multiple virtual channels on a physical channel		Note 2	M	Yes [ ]	

Note 1: Support includes full support for the sequence-controlled and expedited services of FOP-1 as defined in Clause 7.
- If support is limited to the expedited service, then the “Restricted” box should be ticked and the implementation should specify details of the FOP-1 support it provides. The PICS tables in section E.5 are intended for a transfer sublayer implementation with full FOP-1 support.
- If support is otherwise limited, then the implementation in non-conformant.

Note 2: The implementation should supply details of the number of virtual channels supported.

E.5.3 TC Transfer Frame data structure

This section specifies the items for the TC Transfer Frames generated by the implementation.

Table E-14: Items for TC Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
STTF-1	TC Transfer Frame contains all specified fields in the specified positions	6.2		M	Yes [ ]	
STTF-2	All fields of the Transfer Frame Primary Header are set to the specified values	6.2.2		M	Yes [ ]	
STTF-3	Spacecraft Identifier is static throughout all mission phases	6.2.2.6.c		M	Yes [ ]	
STTF-4	A single physical channel can carry virtual channels with different Spacecraft Identifiers	6.2.2.6		O	Yes [ ] No [ ]	
STTF-5	Value contained in the Frame Length field can vary	6.2.2.8.d	Note 1	M	Yes [ ]	
STTF-6	Value contained in the Frame Length field is in the range of 7 to 1023, corresponding to TC Transfer Frame lengths of 8 to 1024 octets	6.2.2.8	Note 2	M	Yes [ ] Restricted [ ]	
STTF-7	In a type-D frame, the Transfer Frame Data Field contains a frame data unit	6.2.3.d	Note 3	M	Yes [ ]	
STTF-8	In a type-C frame, the Transfer Frame Data Field contains exactly one COP-1 control command	6.2.3.e 7.8.2.a		M	Yes [ ]	
STTF-9	The COP-1 control command in a type-C frame conforms to the specification	7.8		M	Yes [ ]	
STTF-10	Value of the Frame Error Control Field is generated as specified in the encoding procedure	6.2.4.2.a 6.2.4.2.b		M	Yes [ ]	

Note 1: A conformant implementation supports the generation of variable-length TC Transfer Frames.

Note 2: Full support implies that the implementation supports all values in the range. If the implementation is restricted to TC Transfer Frame lengths up to 256 octets, then the "Restricted" box should be ticked. Any other restrictions make the implementation non-conformant.

Note 3: The transfer sublayer accepts variable-length data units from the next higher sublayer at the sending end. Within the transfer sublayer, these data units are referred to as frame data units (FDUs).

E.5.4 CLCW data structure

The operation of the COP-1 protocol uses the communications link control word (CLCW), which is passed, via telemetry, from the receiving end to the sending end of the transfer sublayer. This section specifies the items for the CLCWs handled by the implementation.

Table E-15: Items for CLCW

Item	Description	Clause	Notes	Status	Support	Remarks
STCL-1	CLCW contains all specified fields in the specified bit positions	6.3.1.b		M	Yes [ ]	
STCL-2	Mission-specific use of the Status Field	6.3.4		O	Yes [ ] No [ ]	
STCL-3	CLCWs are extracted from telemetry frames and delivered to the FOP-1 of the appropriate virtual channel	6.3.1.c 6.3.1.d 6.3.1.e 6.3.6	Note 1	M	Yes [ ]	
STCL-4	Validation of fields in a CLCW as specified in the first part of clause 7.9.2.2	7.9.2.2	Note 2	M	Yes [ ]	

Note 1: Some of the actions in item STCL-3 can be outside the scope of an implementation of the transfer sublayer. For conformance to item STCL-3, it is sufficient that the implementation supports the actions that are in the scope of the implementation.

Note 2: A field can be validated before the CLCW is passed to FOP-1 (item STCL-4) or it can be validated within FOP-1 (item STFO-10 in Table E-16). This is implementation dependent.

E.5.5 Sending end of COP-1 protocol, FOP-1

This section specifies the items for FOP-1, the sending end of the COP-1 protocol.

Table E-16: Items for FOP-1

Item	Description	Clause	Notes	Status	Support	Remarks
STFO-1	Transfer sublayer has an instance of FOP-1 for each virtual channel	6.1.2.2	Note 1	M	Yes [ ]	
STFO-2	FOP-1 maintains the variables and data structures listed in clause 7.2.2.1	7.2.2.1		M	Yes [ ]	
STFO-3	FOP-1 interface for management of the sequence-controlled service	7.4.2	Note 2 Note 3	M	Yes [ ]	
STFO-4	FOP-1 interface for FDU transfer on the sequence-controlled service	7.4.3	Note 2 Note 4	M	Yes [ ]	
STFO-5	FOP-1 interface for FDU transfer on the expedited service	7.4.4	Note 2 Note 4	M	Yes [ ]	
STFO-6	Signals for the lower interface of FOP-1	7.6	Note 2 Note 5	M	Yes [ ]	
STFO-7	FOP-1 is executed as specified by the FOP-1 state table, including all states and events, and the associated sequences of actions	State table and 7.9.2.4 to 7.9.2.19		M	Yes [ ]	
STFO-8	Execution of the FOP-1 state table as specified in the general rules for state tables	7.9.1		M	Yes [ ]	
STFO-9	Modulo 256 arithmetic for FOP-1 is handled as specified in clause 7.9.2.3	7.9.2.3		M	Yes [ ]	
STFO-10	FOP-1 validates fields in an incoming CLCW and uses the fields to determine the event	7.9.2.2	Note 6	M	Yes [ ]	

Note 1: In the descriptions of remaining items in this table, the phrase "within a virtual channel" or "for each virtual channel" is not repeated.

Note 2: Interactions on the interfaces of FOP-1 are defined as signals which are passed across the interfaces. Associated parameters are specified in an abstract sense and specify the information to be made available. The manner in which the signals and parameters are passed across the interfaces is implementation dependent.

Note 3: For the management of the sequence-controlled service, FOP-1 interfaces to a "management entity". ECSS-E-ST-50-04 does not specify the management entity. Item relating to the management entity are contained in Table E-17.

Note 4: For FDU transfer, FOP-1 interfaces to the "next higher sublayer". In a typical scenario, this is the segmentation sublayer.

Note 5: The lower interface is to the procedures below FOP-1 in the transfer sublayer.

Note 6: See also item STCL-3 in Table E-15.

Table E-17: Items for management of FOP-1

If the management entity (see Note 3 of Table E-16) is not included in the implementation, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
STFM-1	Value of the FOP_Sliding_Window_-Width, K, is set according to the rules in clause 7.2.2.12.2	7.2.2.12.2		M	Yes [ ]	
STFM-2	Following receipt of an Alert notification, the management entity performs recovery actions to try to ensure that data are not lost, duplicated or erroneous.	7.4.2.5.2.c		O	Yes [ ] No [ ]	
STFM-3	In implementations where FARM-1 uses the same single back-end buffer for FDUs carried by type-AD and type-BD frames, the sending-end management entity terminates any ongoing AD service before starting a BD service on the same virtual channel.	7.4.4.2.b		M	Yes [ ]	

E.5.6 Procedures below FOP-1

This section specifies the items which apply collectively to the procedures below FOP-1 in the transfer sublayer.

The frame header procedure and the frame error control procedure ensure that the Transfer Frame Primary Header and the Frame Error Control Field are completed for each TC Transfer Frame generated by the transfer sublayer. Items for these fields are contained in Table E-14. The virtual channel multiplexing items are in Table E-19.

Table E-18: Items for procedures below FOP-1

Item	Description	Clause	Notes	Status	Support	Remarks
STLO-1	Signals for the lower interface of FOP-1	7.6	Note 1	M	Yes [ ]	
STLO-2	On receipt of an Abort request the procedures delete any type-BC or type-AD frames on that virtual channel that are awaiting transmission within the procedures	7.6.3.b	Note 2	O	Yes [ ] No [ ]	
STLO-3	Passing an Abort request to sublayers and layers below the transfer sublayer	7.6.3.b	Note 3	O	Yes [ ] No [ ]	
Note 1: See item STFO-6 in Table E-16. Note 2: Any action by the procedures, sublayers and layers below FOP-1 following receipt of an Abort request is implementation dependent. Note 3: The manner in which the Abort request is passed to the sublayers and layers below the transfer sublayer is implementation dependent.						

E.5.7 Virtual channel multiplexing

TC Transfer Frames support multiple virtual channels on a physical channel. The virtual channel multiplexing takes place within the transfer sublayer. This section specifies the items for the virtual channel multiplexing.

Table E-19: Items for virtual channel multiplexing

Item	Description	Clause	Notes	Status	Support	Remarks
STVX-1	The multiplexer selects a frame depending on the algorithm in use	6.9.2	Note 1	M	Yes [ ]	
Note 1: The implementation should supply details of the multiplexing algorithms supported and the means for setting and selecting priorities where applicable.						

If the implementation does not include the synchronization and channel coding sublayer (item SLAY-3), then section E.6 does not apply. In this case, leave all tables and boxes in section E.6 empty and tick here .... N/A [ ]

E.6 Synchronization and channel coding sublayer

E.6.1 Introduction

Section E.6 contains the PICS tables for the synchronization and channel coding sublayer of a sending-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the synchronization and channel coding sublayer, then all tables and boxes in section E.6 should be ignored. Table E-4 contains the items for the layers and sublayers supported by the implementation.

E.6.2 Elements of the synchronization and channel coding sublayer

Table E-20: Main items for synchronization and channel coding sublayer functions

Item	Description	Clause	Notes	Status	Support	Remarks
SCM-1	CLTU data structure, including BCH codeblock data structure	8.2 8.3 8.6	See E.6.3	M	Yes [ ]	
SCM-2	Pseudo-randomization procedure	8.4	See E.6.4	M	Yes [ ]	
SCM-3	BCH encoding procedure	8.5	See E.6.5	M	Yes [ ]	
SCM-4	Handling of Abort request from FOP-1	7.6.3	See E.6.6 Note 1	O	Yes [ ] No [ ]	
Note 1: Any action by the sublayers and layers below FOP-1 following receipt of an Abort request is implementation dependent. The manner in which the Abort request is passed from FOP-1 in the transfer sublayer to the synchronization and channel coding sublayer is implementation dependent.						

E.6.3 CLTU and BCH codeblock

This section specifies the items for the Command Link Transmission Units (CLTUs) generated by the implementation. A CLTU contains one or more BCH codeblocks and this section also specifies the items for a BCH codeblock.

Table E-21: Items for Command Link Transmission Unit (CLTU)

Item	Description	Clause	Notes	Status	Support	Remarks

Item	Description	Clause	Notes	Status	Support	Remarks
SCCL-1	CLTU contains all specified fields in the specified positions	8.3		M	Yes [ ]	
SCCL-2	Start Sequence contains the specified bit pattern (hexadecimal EB90)	8.3.2.c		M	Yes [ ]	
SCCL-3	Encoded Data field consists of an integral number of BCH codeblocks	8.3.3.c		M	Yes [ ]	
SCCL-4	Encoded Data field contains at least one BCH codeblock	8.3.3.d		M	Yes [ ]	
SCCL-5	If the input data to the synchronization and channel coding sublayer from the layer above consists of TC Transfer Frames, then the input data for the BCH codeblocks of a CLTU consists of a single TC Transfer Frame.	8.3.3.e	Note 1	M	Yes [ ]	
SCCL-6	TC Transfer Frame (or other input data) accepted by the synchronization and channel coding sublayer has length up to 1024 octets	6.2.2.8	Note 2	M	Yes [ ] Restricted [ ]	
SCCL-7	Tail Sequence has a length of eight octets	8.3.4.b		M	Yes [ ]	
SCCL-8	The first seven octets of the Tail Sequence field each contain the bit pattern 11000101 (hexadecimal C5)	8.3.4.c		M	Yes [ ]	
SCCL-9	The final octet of the Tail Sequence field contains the bit pattern 01111001 (hexadecimal 79)	8.3.4.d		M	Yes [ ]	

Note 1: In this case, a CLTU carries exactly one TC Transfer Frame.

Note 2: Full support implies the synchronization and channel coding sublayer implementation can generate a CLTU for a TC Transfer Frame of any legal length. If support is restricted, implementation to specify the frame lengths supported.

Table E-22: Items for BCH codeblocks of a CLTU

Item	Description	Clause	Notes	Status	Support	Remarks
SCBS-1	BCH codeblock has a length of 8 octets (64 bits)	8.2.1.a		M	Yes [ ]	
SCBS-2	BCH codeblock contains all specified fields in the specified positions	8.2		M	Yes [ ]	
SCBS-4	Information field contains data bits from the input data, which is randomized as defined in clause 8.4	8.2.2.c		M	Yes [ ]	
SCBS-5	If the input data do not fit exactly within an integral number of BCH codeblocks then fill bits are added	8.6.1		M	Yes [ ]	
SCBS-6	If a CLTU contains fill bits, then the fill bits are present only in the last BCH codeblock within a CLTU	8.6.2.a		M	Yes [ ]	
SCBS-7	If a BCH codeblock contains fill bits, the fill bits are in the last octets of the Information field of the codeblock.	8.6.2.b		M	Yes [ ]	
SCBS-8	If a BCH codeblock contains fill bits, the length of the fill bits is an integral number of octets.	8.6.2.c		M	Yes [ ]	
SCBS-9	The pattern of the fill consists of a sequence of alternating "ones" and "zeroes" starting with a "zero".	8.6.2.d		M	Yes [ ]	
SCBS-12	The Parity bits consist of the complements of the seven parity check bits, P_0 through P_6	8.2.3.1.c		M	Yes [ ]	
SCBS-15	The Filler bit is always set to zero	8.2.3.2.c		M	Yes [ ]	

E.6.4 Pseudo-randomization procedure

This section specifies the items for the pseudo-randomization procedure.

Table E-23: Items for pseudo-randomization procedure

Item	Description	Clause	Notes	Status	Support	Remarks
SCPR-1	The defined randomizer is used in all CLTUs passed to the physical layer	8.4.2.a		M	Yes [ ]	
SCPR-2	Randomization is applied to the input data in a codeblock, before the encoding procedure for the codeblock is executed	8.4.2.b		M	Yes [ ]	
SCPR-3	The random sequence is generated using the specified polynomial	8.4.3.a		M	Yes [ ]	
SCPR-4	The bit transition generator is compatible with the logical diagram shown in the requirement	8.4.3.b		M	Yes [ ]	
SCPR-5	For each CLTU, the bit transition generator is preset to the "all-ones" state and then is exclusively ORed, bit by bit, with the input data until the end of the input data is reached	8.4.4		M	Yes [ ]	
SCPR-6	Randomization of the fill bits added after the end of the input data to complete the last codeblock of the CLTU	8.4.4		O	Yes [ ] No [ ]	
SCPR-7	The randomization is not applied to the Error Control field at the end of each BCH codeblock	8.4.4		M	Yes [ ]	

E.6.5 BCH encoding procedure

This section specifies the items for the BCH encoding procedure, which uses a modified Bose-Chaudhuri-Hocquenghem (BCH) code.

Table E-24: Items for BCH encoding procedure

Item	Description	Clause	Notes	Status	Support	Remarks
SCBE-1	The BCH encoding uses a (63,56) modified BCH code with the specified generator polynomial to produce the seven parity check bits, P_0 through P_6	8.5	Note 1	M	Yes [ ]	
SCBE-2	The code generator is compatible with the logical diagram shown in clause 8.5	8.5	Note 2	M	Yes [ ]	
SCBE-3	The shift registers for the code generator are initialized to zero for each codeblock	8.5		M	Yes [ ]	

Note 1: The BCH encoding is included within the code generator in the logical diagram shown in clause 8.5.

Note 2: In addition to the generation of the seven parity check bits, P_0 through P_6, the code generator in the diagram also inverts the parity bits for insertion in the Parity bits field, and it generates the zero bit for the Filler bit field.

E.6.6 Handling of Abort request from FOP-1

This section specifies the items for handling an Abort request from FOP-1.

Table E-25: Items for Abort request

If Item SCM-4 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
SPAB-1	On receipt of an Abort request the synchronization and channel coding sublayer deletes any type-BC or type-AD frames on that virtual channel that are awaiting transmission within the synchronization and channel coding sublayer	7.6.3.b	Note 1	O	Yes [ ] No [ ]	
SPAB-2	Passing an Abort request to layers below the synchronization and channel coding sublayer	7.6.3.b	Note 2	O	Yes [ ] No [ ]	

Note 1: Any action by the sublayers and layers below FOP-1 following receipt of an Abort request is implementation dependent.

Note 2: The manner in which the Abort request is passed to the layers below the synchronization and channel coding sublayer is implementation dependent.

If the implementation does not include the physical layer (item SLAY-4), then section E.7 does not apply. In this case, leave all tables and boxes in section E.7 empty and tick here ……………………………………………...…. N/A [ ]

E.7 Physical layer

E.7.1 Introduction

Section E.7 contains the PICS tables for the physical layer of a sending-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the physical layer, then all tables and boxes in section E.7 should be ignored. Table E-4 contains the items for the layers and sublayers supported by the implementation.

Note that section E.7 only concerns the physical layer elements that are within the scope of ECSS-E-ST-50-04. All other elements of the physical layer are outside the scope of this PICS.

E.7.2 Elements of the physical layer

Table E-26 shows the main items for the physical layer.

Table E-26: Main items for physical layer

Item	Description	Clause	Notes	Status	Support	Remarks
SPM-1	Acquisition sequence	9.2.1	See E.7.3	M	Yes [ ]	
SPM-2	Idle sequence	9.2.3	See E.7.3	M	Yes [ ]	
SPM-3	CLTU	9.2.2	See E.7.3	M	Yes [ ]	
SPM-4	Carrier Modulation Modes CMM-1 to CMM-4	9.3.2	See E.7.4	M	Yes [ ]	
SPM-5	PLOP-2 physical layer operation procedure	9.3.4.3	See E.7.5 Note 1	O	Yes [ ] No [ ]	
SPM-6	PLOP-1 physical layer operation procedure	9.3.4.4	See E.7.6	O	Yes [ ] No [ ]	
SPM-7	At least one of the options SPM-5 and SPM-6		Note 2	M	Yes [ ]	
SPM-8	Handling of Abort request from FOP-1	7.6.3	See E.7.7 Note 3	O	Yes [ ] No [ ]	

Note 1: Although PLOP-2 is optional, clause 9.3.4.2 states that PLOP-2 should be used in preference to PLOP-1.

Note 2: A conformant implementation supports one or both of the options SPM-5 and SPM-6.

Note 3: Any action by the sublayers and layers below FOP-1 following receipt of an Abort request is implementation dependent. The manner in which the Abort request is passed from FOP-1 in the transfer sublayer to the physical layer is implementation dependent.

E.7.3 Data structures

This section specifies the items for the acquisition sequences and idle sequences generated by the implementation. It also specifies the items for the Command Link Transmission Units (CLTUs) handled by the implementation.

Table E-27: Items for acquisition sequence and idle sequence

Item	Description	Clause	Notes	Status	Support	Remarks
SPAI-1	Length of the acquisition sequence is at least 128 bits	9.2.1.a		M	Yes [ ]	
SPAI-2	Pattern of the acquisition sequence is alternating "ones" and "zeroes", starting with either a "one" or a "zero"	9.2.1.b		M	Yes [ ]	
SPAI-3	Length of the idle sequence is at least 8 bits	9.2.3.a		M	Yes [ ]	
SPAI-4	Maximum length of the idle sequence is an unconstrained number of bits	9.2.3.a		M	Yes [ ]	
SPAI-5	Pattern of the idle sequence is alternating "ones" and "zeroes", starting with either a "one" or a "zero"	9.2.3.b		M	Yes [ ]	

Table E-28: Items for Command Link Transmission Unit (CLTU)

Item	Description	Clause	Notes	Status	Support	Remarks
SPCL-1	CLTUs handled by the physical layer are randomized as specified in clause 8.4	9.2.2	Note 1	M	Yes [ ]	
SPCL-2	A CLTU handled by the physical layer has length up to 1186 octets (147 BCH codeblocks), resulting from a TC Transfer Frame with length up to 1024 octets	6.2.2.8	Note 2	M	Yes [ ] Restricted [ ]	

Note 1: The specification of the physical layer assumes that the CLTU is the data structure (symbol sequence) furnished from the next higher sublayer.

Note 2: Full support implies the physical layer implementation can handle a CLTU for a TC Transfer Frame of any legal length. If support is restricted, implementation to specify the lengths supported. Lengths can be expressed as frame lengths (octets), as CLTU lengths (octets), or as limits on the number of BCH codeblocks in a CLTU.

E.7.4 Carrier Modulation Modes

This section specifies the items for the Carrier Modulation Modes (CMMs) used during the physical layer operation procedures. The CMMs consist of different states of data modulation upon the RF carrier which creates the physical telecommand channel.

Table E-29: Items for Carrier Modulation Modes (CMMs)

Item	Description	Clause	Notes	Status	Support	Remarks
SPCM-1	CMM-1 is unmodulated carrier only, which is defined as the state in which no telecommand modulation is present	9.3.2.1		M	Yes [ ]	
SPCM-2	In operating the acquisition procedures, the sending end uses information about the lock status of the receiving spacecraft transponders. The information is carried by the No RF Available Flag in the communications link control word (CLCW).	9.3.2.2		M	Yes [ ]	
SPCM-3	CMM-2 is carrier modulated with acquisition sequence	9.3.2.1		M	Yes [ ]	
SPCM-4	CMM-3 is carrier modulated with telecommand data (e.g. CLTU)	9.3.2.1		M	Yes [ ]	
SPCM-5	CMM-4 is carrier modulated with idle sequence	9.3.2.1		M	Yes [ ]	
SPCM-6	In CMM-2, CMM-3 and CMM-4, the sending end uses information about the quality of the received bit stream. The information is carried by the No Bit Lock Flag in the CLCW.	9.3.2.3		M	Yes [ ]	

E.7.5 PLOP-2

This section specifies the items for the physical layer operation procedure PLOP-2.

Table E-30: Items for PLOP-2

If Item SPM-5 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SPPA-1	PLOP-2 procedure invokes the sequence of CMMs shown in the figure in clause 9.3.4.3	9.3.4.3		M	Yes [ ]	
SPPA-2	An idle sequence of at least one octet is inserted between each CLTU	9.3.4.3	Note 1	M	Yes [ ]	
SPPA-3	Maximum length of the idle sequence between each CLTU is unconstrained and is dictated by the timing of the telecommand operations (e.g. when no CLTU is available)	9.3.4.3		M	Yes [ ]	
Note 1: Item SPPA-2 is included within item SPPA-1, but is shown separately here, because the one-octet minimum for the idle between CLTUs is not included in all PLOP-2 specifications.						

E.7.6 PLOP-1

This section specifies the items for the physical layer operation procedure PLOP-2.

Table E-31: Items for PLOP-1

If Item SPM-6 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SPPB-1	PLOP-1 procedure invokes the sequence of CMMs shown in the figure in clause 9.3.4.4	9.3.4.4		M	Yes [ ]	

E.7.7 Handling of Abort request from FOP-1

This section specifies the items for handling an Abort request from FOP-1.

Table E-32: Items for Abort request

If Item SPM-8 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
SPAB-1	On receipt of an Abort request the physical layer deletes the CLTUs for any type-BC or type-AD frames on that virtual channel that are awaiting transmission within the physical layer	7.6.3.b		O	Yes [ ] No [ ]	
SPAB-2	If a frame (in the form of a CLTU) is currently being transmitted, the frame is not deleted and the transmission is allowed to continue to the end of the CLTU.	7.6.3.c		M	Yes [ ]	

Annex F (informative) Draft PICS for ECSS-E-ST-50-04: Receiving end

F.1 Introduction to the PICS

F.1.1 Purpose

This annex contains a draft proforma of the Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telecommand protocols, synchronization and channel coding specified in ECSS-E-ST-50-04. It is provided as information pending a formal decision by ECSS.

Sections F.2 to F.7 constitute the proforma. The normative specification is in ECSS-E-ST-50-04.

The proforma consists of a PICS Requirements List (PRL) for an implementation. The PICS is generated by completing the PRL in accordance with the instructions. The PICS states which capabilities and options of the standard have been implemented.

The following can use the PICS:

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- the user or potential user of the implementation, as a basis for initially checking the possibility of interoperating with another implementation;
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

F.1.2 Format of the PICS

F.1.2.1. Overview of the tables

The first part of the PICS consists of a table of identification information.

The remainder of the PICS consists of a set of PRL tables. Each item in the tables has:

- a unique item identifier
- a description of the item
- reference to the clause(s) in the standard where the requirement(s) can be found
- explanatory notes
- status (mandatory or optional)
- support information
- a column for implementation remarks.

F.1.2.2. Mandatory or optional status

Table F-1 shows the symbols used in the 'status' column of a PRL table.

The status of an item can be conditional on support for other items. In this case, the status column contains a numbered conditional symbol, such as C1, and the associated conditions are defined at the end of the table.

The mandatory status of an item applies only to an implementation. It does not imply that it is mandatory for a project or mission to use the related protocol features.

Table F-1: Symbols for item status

Symbol	Meaning
M	mandatory
O	optional
C*n*	conditional

F.1.2.3. Support provided by the implementation

The 'support' column of a PRL table contains one or more labelled boxes. Table F-2 shows the meaning of the labels.

Table F-2: Labelled boxes for support

Label	Meaning
Yes	The implementation provides full support
No	Not supported by the implementation
Restricted	The implementation provides restricted support
N/A	Not applicable
The 'N/A' label also appears at the top of tables of conditional items, as described in F.1.2.4.	

In general, if the item is mandatory, then the only box in the support column is 'Yes'. If there is a 'Restricted' box, then the table also includes a note, specifying the extra information to be supplied.

To show the conformance of an implementation, the PRL tables are completed as follows:

- For each item in the table, exactly one of the given boxes in the support column should be ticked. Any additional information or parameters should be supplied as indicated in the notes for the item.
- If none of the boxes can be ticked, then the implementation is non-conformant. Exception information is supplied by entering a numbered reference X*n* in the support column. The *n* is an identifier to an accompanying rationale for the nonconformance.

F.1.2.4. Tables of conditional items

There are some PRL tables where all the items are conditional on support for a given optional item. This is indicated at the top of the table, and an 'N/A' box is provided which applies to the table as a whole.

If the implementation does support the given optional item, then the table should be completed as described above.

If the implementation does not support the optional item, then:

- the N/A box at the top of the table should be ticked, and
- all the other boxes in the table should be left blank.

For example, Table F-6 contains items relating to the segmentation function. If an implementation does not support the segmentation function (item RSM-1 in Table F-5), then the N/A box at the top of Table F-6 is ticked and the rest of the table is ignored.

F.2 Identification

Table F-3 contains the items to identify the implementation.

Table F-3: PICS identification

Protocol Implementation Conformance Statement (PICS) for a receiving end implementation of the telecommand protocols specified in ECSS-E-ST-50-04		
Item	**Description**	**Information**
RI1	Date of Statement	
RI2	PICS serial number	
RI3	Supplier	
RI4	Contact details	
RI5	Implementation name	
RI6	Implementation version	
RI7	Other information for full identification, e.g. versions for operating systems	
RI8	Special configuration details	
RI9	Protocol version	
RI10	Protocol addenda and amendments	

F.3 Supported layers and sublayers

This section specifies the items for the layers and sublayers supported by the receiving-end implementation.

Table F-4: Layers and sublayers supported

Item	Description	Clause	Notes	Status	Support	Remarks
RLAY-1	Segmentation sublayer	5	See F.4	O	Yes [ ] No [ ]	
RLAY-2	Transfer sublayer	6 7	See F.5	O	Yes [ ] No [ ]	
RLAY-3	Synchronization and channel coding sublayer	8	See F.6	O	Yes [ ] No [ ]	
RLAY-4	Physical layer elements within the scope of ECSS-E-ST-50-04	9	See F.7	O	Yes [ ] No [ ]	
RLAY-5	At least one of the options RLAY-1 to RLAY-4		Note 1	M	Yes [ ]	
Note 1: A conformant implementation supports one or more of the options RLAY-1 to RLAY-4.						

If the implementation does not include the segmentation sublayer (item RLAY-1), then section F.4 does not apply. In this case, leave all tables and boxes in section F.4 empty and tick here ………………………………………… N/A [ ]

F.4 Segmentation sublayer

F.4.1 Introduction

Section F.4 contains the PICS tables for the segmentation sublayer of a receiving-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the segmentation sublayer, then all tables and boxes in section F.4 should be ignored. Table F-4 contains the items for the layers and sublayers supported by the implementation.

F.4.2 Elements of the segmentation sublayer

Table F-5 shows the main items for segmentation sublayer.

Table F-5: Main items for segmentation sublayer

Item	Description	Clause	Notes	Status	Support	Remarks
RSM-1	Reassembly function	5.5	See F.4.5	O	Yes [ ] No [ ]	
RSM-2	MAP demultiplexing	5.6	See F.4.4	O	Yes [ ] No [ ]	
RSM-3	Deblocking function for packets	5.4	See F.4.6, F.4.7	O	Yes [ ] No [ ]	
RSM-4	At least one of the options RSM-1 to RSM-3		Note 1	M	Yes [ ]	
RSM-5	TC Segment data structure, including support for virtual channels that carry TC Segments	5.2 5.1	See F.4.3 Note 2	C1	Yes [ ] No [ ]	
RSM-6	Support for virtual channels that do not carry TC Segments	5.1	Note 2	C2	Yes [ ] No [ ]	
RSM-7	Within a physical channel, there can be some virtual channels that carry TC Segments and some that do not carry TC Segments	5.1		C3	Yes [ ] N/A[ ]	

Note 1: A conformant implementation of the segmentation sublayer supports one or more of the options RSM-1 to RSM-3.

Note 2: Each data unit received by the segmentation sublayer from the layer or sublayer below belongs to a virtual channel. Within a virtual channel:
- all the data units are TC Segments, or,
- none of the data units are TC Segments.

Conditional status C1: If implementation includes support for item RSM-1 or item RSM-2, then mandatory, else optional.

Conditional status C2: If implementation does not include support for item RSM-5, then mandatory, else optional.

Conditional status C3: If implementation includes support for items RSM-5 and RSM-6, then mandatory, else not applicable.

F.4.3 TC Segments

The TC Segment contains fields to support the segmenting/reassembly and MAP multiplexing/demultiplexing functions of the segmentation sublayer. This section specifies the items for the TC Segment data structures handled by the implementation.

Table F-6: Items for TC Segment data structure

If Item RSM-5 is not supported, this table is not applicable.
In this case, leave boxes below empty and tick here……………………………….. N/A [ ]

Item	Description	Clause	Notes	Status	Support	Remarks
RSTC-1	Length of a TC Segment is an integer number of octets	5.2.1.b		M	Yes [ ]	
RSTC-2	Lengths of the TC Segments can vary	5.2.1.a	Note 1	M	Yes [ ]	
RSTC-3	The implementation can handle TC Segments with lengths in the range of 2 to 1017 octets, corresponding to TC Transfer Frame lengths up to 1024 octets	6.2.2.8	Note 2 Note 3 Note 4	M	Yes [ ] Restricted [ ]	
RSTC-4	TC Segments do not share a virtual channel with other data structures	5.2.1.d	Note 5	M	Yes [ ]	
RSTC-5	A TC Segment consists of a 1-octet Segment Header followed by a variable-length Segment Data Field	5.2.1.c	Note 2	M	Yes [ ]	
RSTC-6	Segment Header contains both specified fields in the specified positions	5.2.2		M	Yes [ ]	
RSTC-7	Sequence Flags field provides information on the segmentation status of the data in the TC Segment	5.2.2.2.c		M	Yes [ ]	
RSTC-8	MAP Identifier provides the identification of the MAP to which the TC Segment belongs	5.2.2.3.c		M	Yes [ ]	

Note 1: A conformant implementation supports the handling of variable-length TC Segments.

Note 2: If the implementation supports the packet assembly controller (see Table F-9), the implementation can also handle TC Segments with a length of 1 octet. In a 1-octet TC Segment, the Segment Data Field is absent.

Note 3: Full support implies that the implementation supports all values in the range. If the implementation is restricted to TC Segment lengths up to 239 or 241 octets, then the "Restricted" box should be ticked. Any other restrictions make the implementation non-conformant.

Note 4: If the next lower sublayer below the segmentation sublayer is the transfer sublayer, then the length of the TC Segment is the same as the length of the Transfer Frame Data Field of the TC Transfer Frame.

Note 5: See Note 2 of Table F-5.

F.4.4 MAP demultiplexing

TC Segments support multiple MAPs on a virtual channel. The MAP demultiplexing takes place within the segmentation sublayer. This section specifies the items for the MAP demultiplexing.

Table F-7: Items for MAP demultiplexing

If Item RSM-2 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RSMX-1	As a result of the demultiplexing, the TC Segment is processed by the segmentation sublayer functions for the appropriate MAP	5.6		M	Yes [ ]	

F.4.5 Reassembly function

This section specifies the items for the reassembly function, which recreates user data units that have been broken into segments by the segmenting function at the sending end..

Table F-8: Items for reassembly function

If Item RSM-1 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RSSG-1	Reassembly function is conducted independently for each MAP	5.5.3.a		M	Yes [ ]	
RSSG-2	Within a virtual channel, there can be some MAPs with the reassembly function enabled and some MAPs with the reassembly function not enabled	5.5		M	Yes [ ]	
RSSG-3	Reassembly function reassembles the Segment Data Fields from a sequence of TC Segments to recreate the original user data unit, as specified in clause 5.5.3	5.5.3		M	Yes [ ]	
RSSG-4	User data units reassembled by the to the reassembly function can be any variable-length octet-aligned data units	5.2.1	Note 1 Note 2	M	Yes [ ]	
RSSG-5	If a failure occurs during reassembly, any resulting incomplete or faulty user data unit is delivered to the next higher layer	5.5.3		O	Yes [ ] No [ ]	
RSSG-6	Packet assembly controller (PAC) defined in clause 5.5.4	5.5.4	Note 3	O	Yes [ ] No [ ]	

Note 1: A user data unit delivered by the segmentation sublayer to the next higher layer or sublayer is a variable-length octet-aligned data unit. Support implies that the reassembly function can handle any type of user data unit delivered by the segmentation sublayer.

Note 2: An implementation can set an upper limit for the length of a user data unit handled by the reassembly function. The upper limit should be specified.

Note 3: See Table F-9 for additional items when the packet assembly controller is supported.

Table F-9: Items for packet assembly controller (PAC)

If Item RSSG-5 is not supported, this table is not applicable.   **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RSPA-1	Use of the PAC is optional		Note 1	O	Yes [ ] No [ ]	
RSPA-2	PAC uses a pair of MAP identifier values as specified in clause 5.5.4.2	5.5.4.2		M	Yes [ ]	
RSPA-3	PAC reassembly function uses the Sequence Flags in the Segment Header of each TC Segment to reassemble the user data units. It does not make use of any length fields contained in the user data units.	5.5.4.3	Note 2 Note 3	M	Yes [ ]	
RSPA-4	PAC has a lockout state, with behaviour as specified in clause 5.5.4.4	5.5.4.4		M	Yes [ ]	
RSPA-5	PAC generates a status report as specified in clause 5.5.4.5	5.5.4.5	Note 4	M	Yes [ ]	
RSPA-6	PAC handles a control segment and MAP reset as specified in clauses 5.5.4.6 and 5.5.4.7	5.5.4.6 5.5.4.7		M	Yes [ ]	

Note 1: Support implies that the implementation can support reassembly for MAPs with the PAC and for MAPs without the PAC. If not supported, then the implementation cannot support reassembly for MAPs without the PAC.

Note 2: The reassembly function is not limited to packets or other data structures with known length fields.

Note 3: Any verification that the length of a reassembled data unit is consistent with the value in a length field is outside the scope of ECSS-E-ST-50-04.

Note 4: ECSS-E-ST-50-04 specifies the contents of a PAC status report but does not specify the format of the status information nor how the report is transported to the appropriate entity at the sending end.

F.4.6 Deblocking function on a virtual channel that carries TC Segments

This section specifies the items for the deblocking function, when it is being used on a virtual channel that carries TC Segments. In this case, the deblocking function operates within a MAP.

Table F-10: Items for deblocking function on a MAP

If Item RSM-3 or item RSM-5 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RSBM-1	Deblocking function is conducted independently for each MAP	5.4.2.a		M	Yes [ ]	
RSBM-2	A virtual channel can have some MAPs where the deblocking function is enabled and some MAPs where the deblocking function is not enabled	5.4.2.c		M	Yes [ ]	
RSBM-3	If the deblocking function is enabled for a MAP, then all user data units on that MAP are packets with the specified packet properties	5.4.2.d		M	Yes [ ]	
RSBM-4	A packet handled by the deblocking function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.4	Note 1	M	Yes [ ] Restricted [ ]	
RSBM-5	Deblocking function extracts packets as specified in clause 5.4.6	5.4.6		M	Yes [ ]	
Note 1: Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.						

F.4.7 Deblocking function on a virtual channel that does not carry TC Segments

This section specifies the items for the deblocking function, when it is being used on a virtual channel that does not carry TC Segments. In this case, the deblocking function operates within a virtual channel.

Table F-11: Items for deblocking function on a virtual channel

If Item RSM-3 or item RSM-6 is not supported, this table is not applicable. **In this case, leave boxes below empty and tick here……………………………….. N/A [ ]**

Item	Description	Clause	Notes	Status	Support	Remarks
RSBL-1	Deblocking function is conducted independently for each virtual channel	5.4.3.a		M	Yes [ ]	
RSBL-2	A physical channel can have some virtual channels where the deblocking function is enabled and some virtual channels where the deblocking function is not enabled	5.4.3.c		M	Yes [ ]	
RSBL-3	If the deblocking function is enabled for a virtual channel, then all user data units on that virtual channel are packets with the specified packet properties	5.4.3.d		M	Yes [ ]	
RSBL-4	A packet handled by the deblocking function has a defined Packet Version Number as specified in CCSDS 135.0-B-3 clause 7.6	5.4.4	Note 1	M	Yes [ ] Restricted [ ]	
RSBL-5	Deblocking function extracts packets as specified in clause 5.4.6	5.4.6		M	Yes [ ]	
Note 1: Full support implies the implementation can handle packets with all the specified Packet Version Numbers. If support is restricted, implementation to specify the Packet Version Numbers supported.						

If the implementation does not include the transfer sublayer (item RLAY-2), then section F.5 does not apply. In this case, leave all tables and boxes in section F.5 empty and tick here …………………………………………………….. N/A [ ]

F.5 Transfer sublayer

F.5.1 Introduction

Section F.5 contains the PICS tables for the transfer sublayer of a receiving-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the transfer sublayer, then all tables and boxes in section F.5 should be ignored. Table F-4 contains the items for the layers and sublayers supported by the implementation.

F.5.2 Elements of the transfer sublayer

Table F-12 shows the main items for transfer sublayer.

Table F-12: Main items for transfer sublayer

Item	Description	Clause	Notes	Status	Support	Remarks
RTM-1	TC Transfer Frame data structure	6.2	See F.5.3	M	Yes [ ]	
RTM-2	CLCW (communications link control word) data structure	6.3	See F.5.4	M	Yes [ ]	
RTM-3	Procedures to process a candidate frame	6.6 6.7 6.8	See F.5.5	M	Yes [ ]	
RTM-4	Virtual channel demultiplexing	6.9	See F.5.6	M	Yes [ ]	
RTM-5	FARM-1 (receiving end of COP-1 protocol)	7	See F.5.7 Note 1	M	Yes [ ]	
RTM-6	Multiple virtual channels on a physical channel		Note 2	M	Yes [ ]	

Note 1: Support includes full support for the sequence-controlled and expedited services of FARM-1 as defined in Clause 7. Otherwise, the implementation in non-conformant.

Note 2: The implementation should supply details of the number of virtual channels supported.

F.5.3 TC Transfer Frame data structure

This section specifies the items for the TC Transfer Frames handled by the implementation.

Table F-13: Items for TC Transfer Frame

Item	Description	Clause	Notes	Status	Support	Remarks
RTTF-1	TC Transfer Frame contains all specified fields in the specified positions	6.2.		M	Yes [ ]	
RTTF-2	The fields of the Transfer Frame Primary Header contain the specified values	6.2.2 6.8.2	Note 1	M	Yes [ ]	
RTTF-3	A single physical channel can carry virtual channels with different Spacecraft Identifiers	6.2.2.6		O	Yes [ ] No [ ]	
RTTF-4	Virtual Channel Identifier, together with the Spacecraft Identifier, identifies the virtual channel to which the frame belongs	6.2.2.7.c		M	Yes [ ]	
RTTF-5	Value contained in the Frame Length field can vary	6.2.2.8.d	Note 2	M	Yes [ ]	
RTTF-6	Value contained in the Frame Length field is in the range of 7 to 1023, corresponding to TC Transfer Frame lengths of 8 to 1024 octets	6.2.2.8	Note 3	M	Yes [ ] Restricted [ ]	
RTTF-7	In a type-D frame, the Transfer Frame Data Field contains a frame data unit	6.2.3.d	Note 4	M	Yes [ ]	
RTTF-8	In a type-C frame, the Transfer Frame Data Field contains exactly one COP-1 control command	6.2.3.e 7.8.2.a		M	Yes [ ]	
RTTF-9	The COP-1 control command in a type-C frame is an Unlock or a Set V(R) command as specified	7.8		M	Yes [ ]	
RTTF-10	Frame Error Control Field is decoded as specified in the decoding procedure	6.2.4.3.a		M	Yes [ ]	
RTTF-11	Frame Error Control Field is not used for error correction	6.2.4.3.b		M	Yes [ ]	

Note 1: Frame header validation procedure validates the values in fields in the Transfer Frame Primary Header as specified in clause 6.8.2. Other fields in the Transfer Frame Primary Header are used by FARM-1.

Note 2: A conformant implementation supports the handling of variable-length TC Transfer Frames.

Note 3: Full support implies that the implementation supports all values in the range. If the implementation is restricted to TC Transfer Frame lengths up to 256 octets, then the “Restricted” box should be ticked. Any other restrictions make the implementation non-conformant.

Note 4: The transfer sublayer delivers variable-length data units to the next higher sublayer at the receiving end. Within the transfer sublayer, these data units are referred to as frame data units (FDUs).

F.5.4 CLCW data structure

The operation of the COP-1 protocol uses the communications link control word (CLCW), which is passed, via telemetry, from the receiving end to the sending end of the transfer sublayer. This section specifies the items for the CLCWs generated by the implementation.

Table F-14: Items for CLCW

Item	Description	Clause	Notes	Status	Support	Remarks
RTCL-1	A CLCW contains all specified fields in the specified bit positions	6.3.1.b		M	Yes [ ]	
RTCL-2	All fields of a CLCW are set to the specified values	6.3.1	Note 1	M	Yes [ ]	
RTCL-3	Mission-specific use of the Status Field	6.3.4		O	Yes [ ] No [ ]	
RTCL-4	CLCWs are generated for each active virtual channel	6.3.1.e		M	Yes [ ]	
RTCL-5	CLCWs are transmitted in telemetry frames	6.3.1.b	Note 2	M	Yes [ ]	

Note 1: FARM-1 supplies the values for Lockout Flag, Wait Flag, Retransmit Flag, FARM B Counter and Report Value. The physical layer supplies the values for No RF Available Flag and No Bit Lock Flag (see items RPL-1 and RPL-2 in Table F-23).

Note 2: Some of the actions to perform item RTCL-5 can be outside the scope of an implementation of the transfer sublayer. For conformance to item RTCL-5, it is sufficient that the implementation supports the actions that are in the scope of the implementation.

F.5.5 Procedures to process a candidate frame

The transfer sublayer receives candidate transfer frames from the layer or sublayer below. This section specifies the items for the procedures which process a candidate transfer frame:

- Frame delimiting and fill removal procedure
- Frame error control procedure
- Frame header validation procedure

Table F-15: Items for procedures to process a candidate frame

Item	Description	Clause	Notes	Status	Support	Remarks
RTCF-1	When the frame delimiting and fill removal procedure receives data from the next lower sublayer, it also receives information to delimit the candidate frame	8.7 6.6	Note 1	M	Yes [ ]	
RTCF-2	Frame delimiting and fill removal procedure processes a candidate frame as specified in clause 6.6	6.6		M	Yes [ ]	
RTCF-3	Frame error control procedure processes a candidate frame as specified in clause 6.7	6.7		M	Yes [ ]	
RTCF-4	Frame header validation procedure processes a candidate frame as specified in clause 6.8	6.8		M	Yes [ ]	
RTCF-5	Valid Frame Arrived Indication when the procedures have a frame to deliver to FARM-1	7.7	Note 2	M	Yes [ ]	

Note 1: The manner in which data is transferred from the next lower sublayer is implementation dependent. The receiving-end channel logic in the synchronization and channel coding sublayer provides information about the start and end of a CLTU.

Note 2: Interactions on the interfaces of FARM-1 are defined as signals which are passed across the interfaces. Associated parameters are specified in an abstract sense and specify the information to be made available. The manner in which the signals and parameters are passed across the interfaces is implementation dependent.

F.5.6 Virtual channel demultiplexing

TC Transfer Frames support multiple virtual channels on a physical channel. The virtual channel demultiplexing takes place within the transfer sublayer. This section specifies the items for the virtual channel demultiplexing.

Table F-16: Items for virtual channel demultiplexing

Item	Description	Clause	Notes	Status	Support	Remarks
RTVX-1	As a result of the demultiplexing, the TC Transfer Frame is processed by the FARM-1 instance for the appropriate virtual channel	6.9.3		M	Yes [ ]	

F.5.7 FARM-1 (receiving end of COP-1 protocol)

This section specifies the items for FARM-1, the receiving end of the COP-1 protocol.

Table F-17: Items for FARM-1

Item	Description	Clause	Notes	Status	Support	Remarks
RTFA-1	Transfer sublayer has an instance of FARM-1 for each virtual channel	6.1.2.3	Note 1	M	Yes [ ]	
RTFA-2	FARM-1 maintains the variables and data structures listed in clause 7.2.3.1	7.2.3.1		M	Yes [ ]	
RTFA-3	FDU Arrived Indication when FARM-1 has an FDU to deliver to the next higher sublayer	7.5.5	Note 2	M	Yes [ ]	
RTFA-4	Buffer management mechanism as specified in clause 7.5.2	7.5.2		M	Yes [ ]	
RTFA-5	Buffer management in the wait system as specified in clause 7.5.3	7.5.3		M	Yes [ ]	
RTFA-6	Multiple output buffers (back-end buffers) for a virtual channel	7.5	Note 3	O	Yes [ ] No [ ]	
RTFA-7	Buffer management for a FARM-1 implementation which provides only one back-end buffer for a virtual channel	7.5.4		C1	Yes [ ] N/A [ ]	
RTFA-8	Valid Frame Arrived Indication when the procedures below FARM-1 have a frame to deliver to FARM-1	7.7	Note 2	M	Yes [ ]	
RTFA-9	FARM-1 is executed as specified by the FARM-1 state table, including all states and events, and the additional provisions	State table and 7.9.3.3 to 7.9.3.6		M	Yes [ ]	
STFO-8	Execution of the FARM-1 state table as specified in the general rules for state tables	7.9.1		M	Yes [ ]	
STFO-9	Modulo 256 arithmetic for FARM-1 is handled as specified in clause 7.9.3.2	7.9.3.2		M	Yes [ ]	

Note 1: In the descriptions of remaining items in this table, the phrase “within a virtual channel” or “for each virtual channel” is not generally repeated.

Note 2: Interactions on the interfaces of FARM-1 are defined as signals which are passed across the interfaces. Associated parameters are specified in an abstract sense and specify the information to be made available. The manner in which the signals and parameters are passed across the interfaces is implementation dependent.

Note 3: If the implementation provides only one FARM-1 back-end buffer for a virtual channel, then item RTFA-7 is mandatory.

Conditional status C1: If implementation includes support for item RTFA-6, then not applicable, else mandatory

If the implementation does not include the synchronization and channel coding sublayer (item RLAY-3), then section F.6 does not apply. In this case, leave all tables and boxes in section F.6 empty and tick here ………… N/A [ ]

F.6 Synchronization and channel coding sublayer

F.6.1 Introduction

SectionF.6 contains the PICS tables for the synchronization and channel coding sublayer of a receiving-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the synchronization and channel coding sublayer, then all tables and boxes in section F.6 should be ignored. Table E-4 contains the items for the layers and sublayers supported by the implementation.

F.6.2 Elements of synchronization and channel coding sublayer

Table F-18 shows the main items for synchronization and channel coding sublayer.

Table F-18: Main items for synchronization and channel coding sublayer

Item	Description	Clause	Notes	Status	Support	Remarks
RCM-1	CLTU data structure, including BCH codeblock data structure	8.2 8.3	See F.6.3	M	Yes [ ]	
RCM-2	Receiving-end channel logic	8.7	See F.6.4	M	Yes [ ]	
RCM-3	BCH decoding procedure	8.8	See F.6.5	M	Yes [ ]	
RCM-4	Pseudo-derandomization procedure	8.4	See F.6.6	M	Yes [ ]	

F.6.3 CLTU and BCH codeblock

This section specifies the items for the Command Link Transmission Units (CLTUs) and the BCH codeblocks handled by the implementation.

Table F-19: Items for Command Link Transmission Unit (CLTU) and BCH codeblock

Item	Description	Clause	Notes	Status	Support	Remarks
RCCB-1	A CLTU handled by the synchronization and channel coding sublayer has length up to 1186 octets (147 BCH codeblocks), resulting from a TC Transfer Frame with length up to 1024 octets	6.2.2.8	Note 1	M	Yes [ ] Restricted [ ]	
RCCB-2	BCH codeblock has a length of 8 octets (64 bits)	8.2.1.a		M	Yes [ ]	
Note 1: Full support implies the synchronization and channel coding sublayer implementation can handle a CLTU for a TC Transfer Frame of any legal length. If support is restricted, implementation to specify the lengths supported. Lengths can be expressed as frame lengths (octets), as CLTU lengths (octets), or as limits on the number of BCH codeblocks in a CLTU.						

F.6.4 Receiving-end channel logic

This section specifies the items for the receiving-end channel logic.

Table F-20: Items for receiving-end channel logic

Item	Description	Clause	Notes	Status	Support	Remarks
RCCL-1	Receiving-end channel logic has a means to receive data from the physical layer. The data includes the bits from the demodulated incoming bitstream, together with information about gain or loss of bit lock.	8.7	Note 1	M	Yes [ ]	
RCCL-2	Receiving-end channel logic implements the states, events and associated actions as specified in clause 8.7, including its tables and notes	8.7		M	Yes [ ]	
RCCL-3	When the receiving-end channel logic is searching for the Start Sequence, a bit sequence which matches the Start Sequence bit pattern with a one bit error is recognized as a Start Sequence	8.3.2.d		M	Yes [ ]	
RCCL-4	When the receiving-end channel logic transfers data to the next higher sublayer, it also provides information about the boundaries of the CLTUs	8.7 6.6	Note 2	M	Yes [ ]	

Note 1: The manner in which the data is received from the physical layer is implementation dependent.

Note 2: The manner in which data is transferred to the next higher sublayer is implementation dependent. The frame delimiting and fill removal procedure in the transfer sublayer uses information about the start and end of a CLTU to detect the start of a TC Transfer Frame and to remove fill data.

F.6.5 BCH decoding procedure

This section specifies the items for the BCH decoding procedure.

Table F-21: Items for BCH decoding procedure

Item	Description	Clause	Notes	Status	Support	Remarks
RCBD-1	BCH codeblocks are decoded in error-correcting mode.	8.8.2	Note 1	M	Yes [ ]	

Note 1: The error-correcting mode is also known as single-error correction (SEC).

F.6.6 Pseudo-derandomization procedure

This section specifies the items for the pseudo-derandomization procedure.

Table F-22: Items for pseudo-derandomization procedure

Item	Description	Clause	Notes	Status	Support	Remarks
RCPR-1	The defined derandomizer is used for the data from all CLTUs received from the physical layer	8.4.2.a		M	Yes [ ]	
RCPR-2	Derandomization is applied to the successfully decoded data from a codeblock, after the decoding procedure for the codeblock is executed	8.4.2.b		M	Yes [ ]	
RCPR-3	The random sequence is generated using the specified polynomial	8.4.3.a		M	Yes [ ]	
RCPR-4	The bit transition generator is compatible with the logical diagram shown in the requirement	8.4.3.b		M	Yes [ ]	
RCPR-5	For each CLTU, the bit transition generator is preset to the "all-ones" state and then is exclusively ORed, bit by bit, with the successfully decoded data until the end of the CLTU is reached	8.4.4	Note 1 Note 2	M	Yes [ ]	
RCPR-6	The derandomization is not applied to the Error Control field at the end of each BCH codeblock	8.4.4		M	Yes [ ]	

Note 1: The end of a CLTU is indicated by the event E4 or the event E2 specified in clause 8.7.

Note 2: Any fill octets added to the last codeblock of the CLTU are derandomized, even if they were not randomized at the sending end.

If the implementation does not include the physical layer (item RLAY-4), then section F.7 does not apply. In this case, leave all tables and boxes in section F.7 empty and tick here ……………………………………………...…. N/A [ ]

F.7 Physical layer

Section F.7 contains the PICS tables for the physical layer of a receiving-end implementation of the telecommand protocols specified in ECSS-E-ST-50-04. If the implementation does not include the physical layer, then all tables and boxes in this section should be ignored. Table F-4 contains the items for the layers and sublayers supported by the implementation.

Note that section F.7 only concerns the physical layer elements that are within the scope of ECSS-E-ST-50-04. All other elements of the physical layer are outside the scope of this PICS.

Table F-23: Items for physical layer

Item	Description	Clause	Notes	Status	Support	Remarks
RPL-1	Provide the information specified in clause 6.3.8, about the status of the receiving spacecraft transponders. The information is carried to the sending end in the No RF Available Flag in the communications link control word (CLCW).	6.3.8		M	Yes [ ]	
RPL-2	Provide the information specified in clause 6.3.9, about the status of the receiving spacecraft transponders. The information is carried to the sending end in the No Bit Lock Flag in the communications link control word (CLCW).	6.3.9		M	Yes [ ]	
RPL-3	Provide data to the receiving end of the synchronization and channel coding sublayer logic defined in clause 8.7. The data includes the bits from the demodulated incoming bitstream, together with information about gain or loss of bit lock.	8.7	Note 1	M	Yes [ ]	
Note 1: The manner in which the data is passed to the synchronization and channel coding sublayer is implementation dependent.						

Annex G (informative) False decoding probabilities for concatenated R-S *E*=8 with 4D-8PSK-TCM

G.1 Introduction

ECSS-E-ST-50-01 includes the option for Reed-Solomon (R-S) codes with *E*=8, restricted to channels where the modulation scheme is 4D-8PSK-TCM. In that case, interleaving depth *I*=8 is mandatory. See section 4.5.3 of this handbook.

For *E*=8, there is an increase in the probability of decoding errors, compared to codes with *E*=16. A decoding error occurs when the decoder finds a codeword other than the transmitted codeword (i.e. when the error pattern is within a distance less than *E* symbols of a nonzero R-S codeword).

This annex provides performance information on the false decoding probabilities for the R-S code *E*=8 with 4D-8PSK-TCM.

G.2 False decoding probabilities

The use of R-S codes in concatenation offers very good BER and FER performances. It is also known that these codes have error detection capabilities in addition to their error correction capabilities. Nevertheless, as for other block codes, R-S codes can exhibit some false decoding probabilities, in so far as the decoder considers the correction with respect to the nearest codeword (in terms of Hamming distance).

Some classical computations of these false decoding probabilities have already been proposed in the literature (for example in *Digital Communications*, J.G. Proakis) but they have been developed and computed for non-concatenated links. The specific case of R-S concatenation with 4D-8PSK-TCM needs some more explanation, because 4D-8PSK-TCM deals with 8- or 10-bit symbols, so errors at the output of the TCM decoder are quite specific. We detail here the calculations and give some results showing that, for high rate TM links operating points, the false decoding probabilities remain very low, and are negligible in terms of contribution to overall BER.

$$\mathit{FalseDecoding\,Probability} \le \sum_{j=2t+1}^{n} \binom{n}{j} p^{j} (1-p)^{n-j} \quad \text{(from Proakis)}$$

with p being the outer code symbol error probability at the output of the inner code decoder

Considering *p* being the Word (byte) Error Rate which is given here (from CCSDS 413.0-G), it is possible to compute a more close approximation (see Proakis for details) with the following formula:

$$FalseDecoding\ Probability \cong BER_{inner} \sum_{j=2t}^{n} \binom{n}{j} WER_{inner}{}^{j} \left(1 - WER_{inner}\right)^{n-j}$$

It is then possible to compute the Bit Error Rate and the Frame Error Rate and also the False Decoding Probability with respect to the inner WER (byte). We provide also some curves showing that for mean BER less than 10^{-9} or FER less than $5*10^{-7}$ which correspond to targeted application (transmission of compressed image data), the False Decoding Probability remains below 10^{-19}. This is negligible as far as it corresponds to one failure per 10000 years for a 700 Mb/s transmission of 1 hour per day.

Table G-1: 4D-8PSK-TCM 2 b/s/Hz + RS(254,238), I=8

WER_inner	1.0E-2	7.0E-3	5.0E-3	3.0E-3	1.8E-3	1.0E-3	7.0E-4
BER_inner	3.4E-3	2.4E-3	1.7E-3	9.9E-4	5.8E-4	3.2E-4	2.2E-4
BER_global	1.5E-5	1.1E-6	8.4E-8	1.3E-9	1.7E-11	9.9E-14	4.2E-15
FER_global	7.9E-3	6.2E-4	5.2E-5	8.2E-7	1.1E-8	6.8E-11	3.2E-12
False Decoding Proba.	3.2E-11	1.4E-13	7.3E-16	1.9E-19	4.2E-23	2.3E-27	5.6E-30

Table G-2: 4D-8PSK-TCM 2.5 b/s/Hz + RS(254,238), I=8

WER_inner	1.0E-2	7.0E-3	5.0E-3	3.0E-3	1.8E-3	1.0E-3	7.0E-4
BER_inner	3.3E-3	2.3E-3	1.7E-3	1.0E-3	6.0E-4	3.3E-4	2.3E-4
BER_global	1.5E-5	1.1E-6	8.4E-8	1.3E-9	1.7E-11	1.0E-13	4.5E-15
FER_global	4.7E-3	4.9E-4	4.3E-5	9.5E-7	1.3E-8	8.3E-11	3.8E-12
False Decoding Proba.	3.1E-11	1.4E-13	7.3E-16	1.9E-19	4.3E-23	2.4E-27	5.9E-30

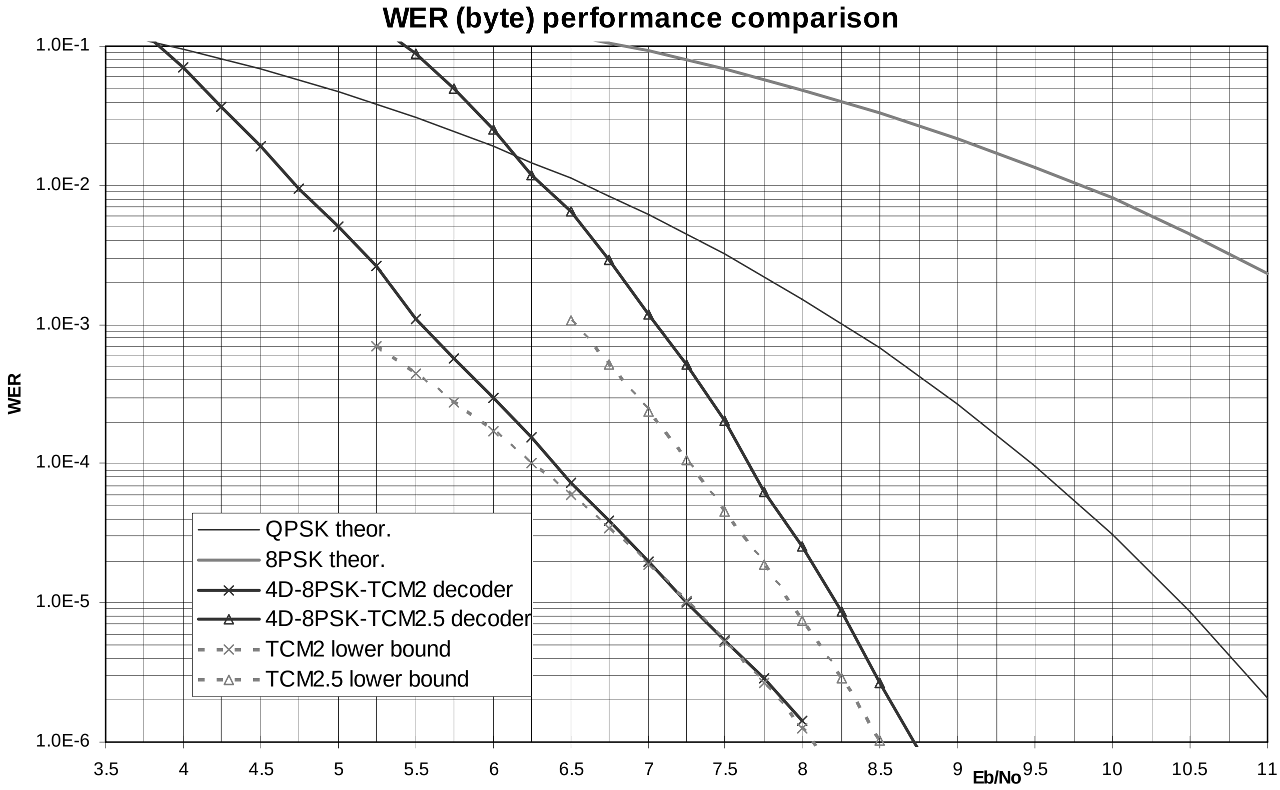

Figure G-1: WER (byte) Performance Comparison

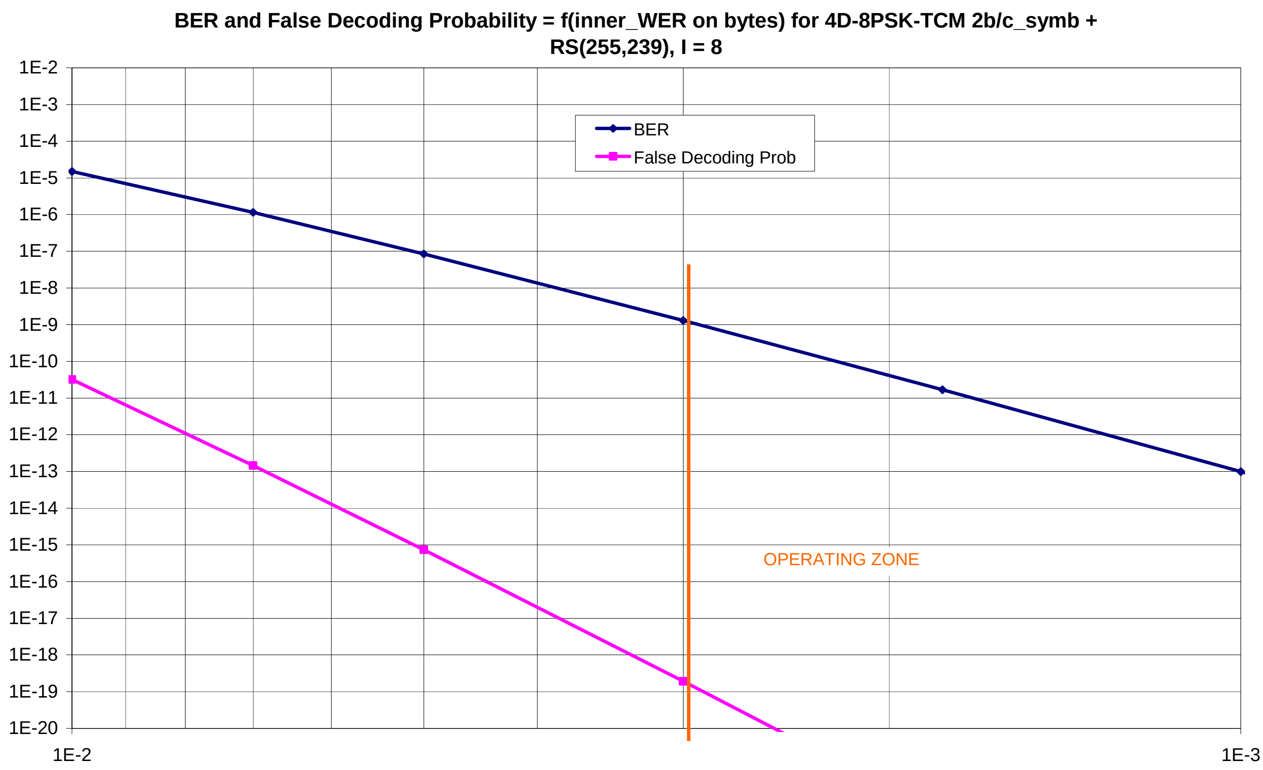

Figure G-2: For 4D-8PSK-TCM 2 b/s/Hz + RS(254,238), I=8

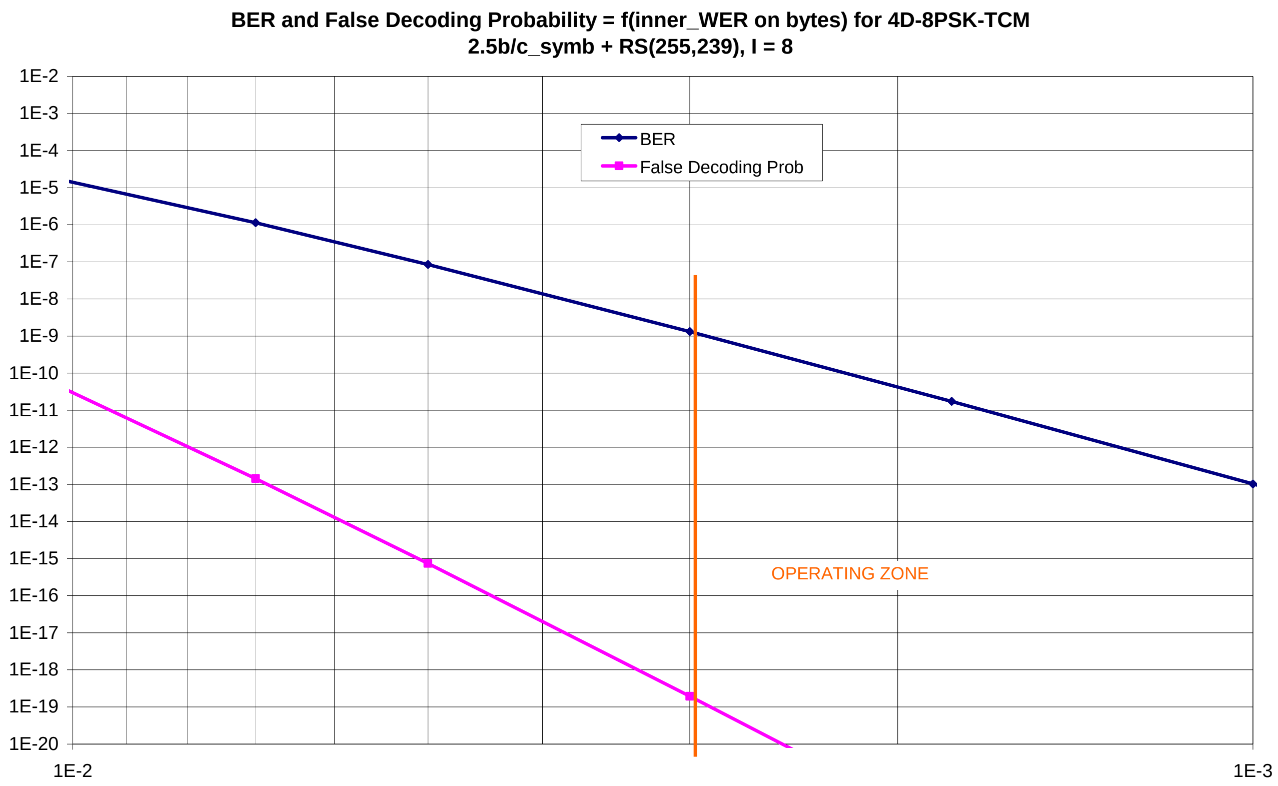

Figure G-3: For 4D-8PSK-TCM 2.5 b/s/Hz + RS(254,238), I=8

References

This handbook uses the following method when referencing ECSS and CCSDS documents:

- The general form of the reference is used in most cases and is the form shown in this section. The general form does not include the issue number or letter: for example, CCSDS-131.0-B or ECSS-E-ST-50-01. The reader can use the most recent issue of the document.
- Reference to a specific issue is used where necessary. In this case, the issue number or letter is shown: for example, CCSDS-131.0-B-1 or ECSS-E-ST-50-01.

Published ECSS standards can be downloaded from the ECSS web site at http://www.ecss.nl.

CCSDS documents can be downloaded from the CCSDS web site at http://www.ccsds.org.

CCSDS 120.0-G	Lossless Data Compression – Informational Report
CCSDS 120.1-G	Image Data Compression – Informational Report
CCSDS 121.0-B	Lossless Data Compression
CCSDS 122.0-B	Image Data Compression
CCSDS 130.1-G	TM Synchronization and Channel Coding – Summary of Concept and Rationale
CCSDS 131.0-B	TM Synchronization and Channel Coding
CCSDS 132.0-B	TM Space Data Link Protocol
CCSDS 133.0-B	Space Packet Protocol
CCSDS 133.1-B	Encapsulation Service
CCSDS 135.0-B	Space Link Identifiers
CCSDS 211.0-B	Proximity-1 Space Link Protocol - Data Link Layer
CCSDS 211.1-B	Proximity-1 Space Link Protocol - Physical Layer
CCSDS 211.2-B	Proximity-1 Space Link Protocol - Coding and Synchronization Sublayer
CCSDS 230.1-G	TC Synchronization and Channel Coding – Summary of Concept and Rationale
CCSDS 231.0-B	TC Synchronization and Channel Coding
CCSDS 232.0-B	TC Space Data Link Protocol

CCSDS 232.1-B	Communications Operation Procedure-1
CCSDS 301.0-B	Time Code Formats
CCSDS 320.0-B	CCSDS Global Spacecraft Identification Field: Code Assignment Control Procedures
CCSDS 350.0-G	The Application of CCSDS Protocols to Secure Systems
CCSDS 401.0-B	Radio Frequency and Modulation Systems – Part 1 Earth Stations and Spacecraft
CCSDS 413.0-G	Bandwidth-Efficient Modulations: Summary of Definition, Implementation and Performance
CCSDS 701.0-B-3-S	Advanced Orbiting Systems, Networks and Data Links: Architectural Specification, Issue 3, June 2001 *[obsolete document with silver status]*
CCSDS 714.0-B	Space Communications Protocol Specification (SCPS) – Transport Protocol (SCPS-TP)
CCSDS 720.1-G	CCSDS File Delivery Protocol (CFDP) – Part 1: Introduction and Overview
CCSDS 720.2-G	CCSDS File Delivery Protocol (CFDP) – Part 2: Implementers Guide
CCSDS 720.3-G	CCSDS File Delivery Protocol (CFDP) – Part 3: Interoperability Testing Final Report
CCSDS 727.0-B	CCSDS File Delivery Protocol (CFDP)
CCSDS 732.0-B	AOS Space Data Link Protocol
CCSDS 740.0-G	Mars Mission Protocol Profiles – Purpose and Rationale
CCSDS 910.0-G	Space Link Extension Services – Executive Summary
ECSS-E-ST-50	Space engineering - Communications
ECSS-E-ST-50-01	Space data links - Telemetry synchronization and channel coding
ECSS-E-ST-50-02	Ranging and Doppler tracking
ECSS-E-ST-50-03	Space data links - Telemetry transfer frame protocol
ECSS-E-ST-50-04	Space data links - Telecommand protocols, synchronization and channel coding
ECSS-E-ST-50-05	Radio frequency and modulation
ECSS-E-ST-50-11	SpaceWire RMAP protocol
ECSS-E-ST-50-12	SpaceWire - Links, nodes, routers and networks
ECSS-E-ST-50-13	MIL 1553 Standard extension
ECSS-E-ST-50-14	Spacecraft discrete interfaces
ECSS-E-ST-70-41	Ground systems and operations - Telemetry and telecommand packet utilization

PSS-04-151	ESA Telecommand decoder specification, Issue 1, September 1993 *[obsolete document]*
RFC 791	Internet Protocol, Protocol Specification, Sep 1981 *[now known as IPv4]*
RFC 1883	Internet Protocol, Version 6 (IPv6) Specification, Dec 1995
ISO/IEC 7498-1	Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model, 2nd edition, Geneva 1994
TEC-ETC/2008.66/PMcM report	The transponder dual lock issue: Background and recommended remedial actions, June 2008