

DIN EN 14776



ICS 49.140

**Raumfahrttechnik –
Bodensysteme und Bodenbetrieb –
Anwendung des Telemetrie- und Telekommando-Datenpakets;
Englische Fassung EN 14776:2004**

Space engineering –
Ground systems and operations –
Telemetry and telecommand packet utilization;
English version EN 14776:2004

Ingénierie spatiale –
Systèmes au sol et opérations –
Utilisation de la télémétrie et de la télécommande par paquets;
Version anglaise EN 14776:2004

Gesamtumfang 189 Seiten

Normenausschuss Luft- und Raumfahrt (NL) im DIN

Nationales Vorwort

Die *European Cooperation for Space Standardization (ECSS)*, eine Einrichtung zur Erstellung eines kohärenten, einheitlichen Normenwerkes zur Anwendung in allen europäischen Raumfahrtprojekten, hat am 1996-11-13 vom *Europäischen Komitee für Normung (CEN)* das Mandat (M/237) erhalten, für den CEN-Subsector T02 „Aerospace“ Europäische Normen (EN) auszuarbeiten.

Aufgrund der zwischen ISO/TC 20/SC 14 „Raumfahrt“ und ECSS vereinbarten A-Liaison werden die Normungsprogramme dieser Partner abgestimmt und die Abstimmungsregeln gemäß Abschnitt 5.2 der Wiener-Vereinbarung angewendet.

Die vorliegende Norm EN 14776:2004 wurde von CEN/CS T02 auf Basis der ECSS-E-70-41 erarbeitet.

English version

Space engineering - Ground systems and operations - Telemetry and telecommand packet utilization

Ingénierie spatiale - Systèmes au sol et opérations -
Utilisation de la télémétrie et de la télécommande par
paquets

Raumfahrttechnik - Bodensysteme und Bodenbetrieb -
Anwendung des Telemetrie- und Telekommando-
Datenpakets

This European Standard was approved by CEN on 30 April 2004.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: rue de Stassart, 36 B-1050 Brussels

Contents

	Page
Foreword.....	7
Introduction	8
1 Scope	9
2 Normative references	10
3 Terms, definitions and abbreviated terms	10
3.1 Terms and definitions.....	10
3.2 Abbreviated terms	13
4 PUS operations concepts	14
4.1 Introduction	14
4.2 Application processes.....	15
4.3 Packet design and routing	15
4.4 Telecommanding	17
4.5 Telemetry reporting	21
4.6 Software management.....	22
4.7 On-board operations scheduling	23
4.8 On-board monitoring	24
4.9 On-board operations procedures.....	25
4.10 Attaching actions to on-board events	25
4.11 On-board storage and retrieval	25
4.12 Telemetry generation and forwarding	26
4.13 Memory management.....	27
4.14 Diagnostic mode	27
4.15 Off-line testing.....	28
5 Service specification	28
5.1 Introduction	28
5.2 Conventions	29
5.3 Telecommand packet structure.....	31
5.4 Telemetry source packet structure	35
5.5 Standard services	38
6 Telecommand verification service	39
6.1 Scope	39
6.2 Service concept	39
6.3 Service requests and reports	40
6.4 Capability sets.....	42
7 Device command distribution service	42
7.1 Scope	42
7.2 Service concept	43
7.3 Service requests and reports	43
7.4 Capability sets.....	45
8 Housekeeping and diagnostic data reporting service	45
8.1 Scope	45
8.2 Service concept	46
8.3 Service requests and reports	47

8.4	Capability sets	53
9	Parameter statistics reporting service	56
9.1	Scope	56
9.2	Service concept	56
9.3	Service requests and reports	56
9.4	Capability sets	60
10	Event reporting service.....	60
10.1	Scope	60
10.2	Service concept	61
10.3	Service requests and reports	61
10.4	Capability sets	62
11	Memory management service	63
11.1	Scope	63
11.2	Service concept	63
11.3	Service requests and reports	64
11.4	Capability sets	69
12	Function management service.....	70
12.1	Scope	70
12.2	Service concept	70
12.3	Service requests and reports	70
12.4	Capability sets	71
13	Time management service.....	71
13.1	Scope	71
13.2	Service concept	72
13.3	Service requests and reports	72
13.4	Capability sets	73
14	On-board operations scheduling service	74
14.1	Scope	74
14.2	Service concept	74
14.3	Service requests and reports	78
14.4	Capability sets	91
15	On-board monitoring service	92
15.1	Scope	92
15.2	Service concept	93
15.3	Service requests and reports	97
15.4	Capability sets	108
16	Large data transfer service.....	108
16.1	Scope	108
16.2	Service concept	109
16.3	Service requests and reports	113
16.4	Capability sets	117
17	Packet forwarding control service.....	118
17.1	Scope	118
17.2	Service concept	118
17.3	Service requests and reports	119
17.4	Capability sets	124
18	On-board storage and retrieval service	125
18.1	Scope	125
18.2	Service concept	126
18.3	Service requests and reports	128
18.4	Capability sets	136
19	Test service	137
19.1	Scope	137
19.2	Service concept	138

19.3	Service requests and reports	138
19.4	Capability sets.....	138
20	On-board operations procedure service	138
20.1	Scope	138
20.2	Service concept	139
20.3	Service requests and reports	139
20.4	Capability sets.....	144
21	Event-action service	145
21.1	Scope	145
21.2	Service concept	145
21.3	Service requests and reports	146
21.4	Capability sets.....	148
22	Summary of service requests and reports.....	149
23	Parameter types and structure rules	153
23.1	Introduction	153
23.2	Conventions	154
23.3	Encoding formats of parameter types	154
23.4	Tailoring of packet structures for a mission.....	155
23.5	Simple parameter types	156
23.6	Structured field types.....	163
Annex A (normative)	Examples	170
Annex B (normative)	Mission constants	178
Annex C (normative)	Spacecraft time protocols	180
Annex D (normative)	Command pulse distribution.....	184
Bibliography		187

Figures

Figure 1 — Example of a telecommand execution profile	18
Figure 2 — Telecommand system layers	20
Figure 3 — The relation between execution result and interlock status	77
Figure 4 — Parameter check definitions and check status	95
Figure 5 — The splitting of a service data unit in parts to be downlinked (uplinked)	110
Figure 6 — An example of a storage and retrieval approach.....	127
Figure 7 — States and transitions for an on-board operations procedure	139
Figure 8 — Diagram conventions used by the structure rules set #1	164
Figure 9 — Diagram conventions used by the structure rules set #1 (continued).....	165
Figure A.1 — Standard packet check sequence generation	170
Figure A.2 — Encoder	172
Figure A.3 — Decoder	172

Tables

Table 1 — Telecommand acceptance and execution telemetry	18
Table 2 — Standard services specified within this document	38
Table 3 — Summary of telecommand verification service minimum capabilities	42
Table 4 — Summary of telecommand verification service additional capabilities	42
Table 5 — Summary of device command distribution service minimum capabilities	45
Table 6 — Summary of device command distribution service additional capabilities	45
Table 7 — Summary of housekeeping sub-service minimum capabilities	53
Table 8 — Summary of diagnostic sub-service minimum capabilities	54
Table 9 — Summary of report definitions control additional capabilities	54
Table 10 — Summary of report definitions reporting additional capabilities	54
Table 11 — Summary of sampling time offset reporting additional capabilities	55
Table 12 — Summary of filtered mode minimum capabilities	55
Table 13 — Summary of filtered mode additional capabilities	55
Table 14 — Summary of parameter statistics reporting service minimum capabilities	60
Table 15 — Summary of parameter statistics reporting service additional capabilities	60
Table 16 — Summary of event reporting service minimum capabilities	62
Table 17 — Summary of event reporting service additional capabilities	63
Table 18 — Summary of memory management service additional capabilities	69
Table 19 — Summary of function management service minimum capabilities	71
Table 20 — Summary of rate control sub-service minimum capabilities	73
Table 21 — Summary of time reporting sub-service minimum capabilities	73
Table 22 — Decision table for the release status of a telecommand	76
Table 23 — Summary of on-board operations scheduling service minimum capabilities	91
Table 24 — Summary of on-board operations scheduling service additional capabilities	92
Table 25 — Summary of on-board monitoring service minimum capabilities	108
Table 26 — Summary of on-board monitoring service additional capabilities	108
Table 27 — Summary of sending sub-service (downlink) minimum capabilities	117
Table 28 — Summary of receiving sub-service (uplink) minimum capabilities	117

Table 29 — Summary of sending sub-service (downlink) additional capabilities	118
Table 30 — Summary of receiving sub-service (uplink) additional capabilities.....	118
Table 31 — Decision table for the forwarding status of a packet	119
Table 32 — Summary of packet forwarding control service minimum capabilities	124
Table 33 — Summary of packet forwarding control service additional capabilities.....	125
Table 34 — Summary of packet selection sub-service minimum capabilities	136
Table 35 — Summary of storage and retrieval sub-service minimum capabilities.....	137
Table 36 — Summary of packet selection sub-service additional capabilities	137
Table 37 — Summary of storage and retrieval sub-service additional capabilities	137
Table 38 — Summary of test service minimum capabilities	138
Table 39 — Summary of on-board operations procedure service minimum capabilities	144
Table 40 — Summary of on-board operations procedure service additional capabilities	145
Table 41 — Summary of event-action service minimum capabilities	148
Table 42 — Summary of event-action service additional capabilities	149
Table 43 — Summary of requests and reports for PUS standard services.....	149

Foreword

This document (EN 14776:2004) has been prepared by CEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by February 2005, and conflicting national standards shall be withdrawn at the latest by February 2005.

It is based on a previous version¹⁾ originally prepared by the ECSS Working Group for Ground Systems and Operations, reviewed by the ECSS Engineering Panel and approved by the ECSS Steering Board. The European Cooperation for Space Standardization (ECSS) is a cooperative effort of the European Space Agency, National Space Agencies and European industry associations for the purpose of developing and maintaining common standards.

This European Standard is one of the series of space standards intended to be applied together for the management, engineering and product assurance in space projects and applications.

Requirements in this European Standard are defined in terms of what shall be accomplished, rather than in terms of how to organize and perform the necessary work. This allows existing organizational structures and methods to be applied where they are effective, and for the structures and methods to evolve as necessary without rewriting the standards.

The formulation of this European Standard takes into account the existing EN ISO 9000 family of documents.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

¹⁾ ECSS-E-70-41

Introduction

ECSS-E-50 and the CCSDS recommendations for packet telemetry and telecommand address the end-to-end transport of telemetry and telecommand data between user applications on the ground and application processes on-board the satellite, and the intermediate transfer of these data through the different elements of the ground and space segments.

This packet utilization standard (PUS) complements these standards by defining the application-level interface between ground and space, in order to satisfy the requirements of electrical integration and testing and flight operations.

1 Scope

This document addresses the utilization of telecommand packets and telemetry source packets for the purposes of remote monitoring and control of subsystems and payloads.

This document does not address mission-specific payload data packets, but the rules contained herein can be extended to suit the requirements of any mission.

This document does not address audio and video data as they are not contained within either telecommand or telemetry source packets.

This document is structured as follows. Firstly a set of operational concepts is identified, covering all the fundamental requirements for spacecraft monitoring and control during satellite integration, testing and flight operations. A set of services that satisfy these operational concepts is then defined. The detailed specification of these services includes the structure and contents of the associated service requests (telecommand packets) and service reports (telemetry source packets).

This document is applicable to any mission, no matter what its domain of application, orbit or ground station coverage characteristics. However, it is not the intention that the PUS is applicable to a given mission in its entirety. The operational concepts and corresponding services contained in this document cover a wide spectrum of operational scenarios and, for a given mission, only a subset of these operational concepts and services is appropriate.

Choices are made early in the design phase of a new mission resulting in the need to tailor the PUS to suit the requirements of the particular mission. These choices include:

- a) The on-board system design and architecture, in terms of the number of on-board application processes, their on-board implementation (e.g. the allocation to on-board processors) and their roles (i.e. which functions or subsystems or payloads they support).
- b) Which PUS services are supported by each application process.

NOTE Tailoring is a process by which individual requirements or specifications, standards and related documents are evaluated and made applicable to a specific project by selection, and in some exceptional cases, modification of existing or addition of new requirements (ECSS-M-00-02A, clause 3 (Reference [11])).

Each mission should document the results of this design and selection process in a space-ground interface control document (SGICD).

Some missions implement a centralized architecture with a small number of application processes, whilst others have a highly-distributed architecture within which a correspondingly larger number of application processes are distributed across several on-board processors.

The specification of services in this document is adapted to the expectation that different missions require different levels of complexity and functionality from a given service. To this end, all services are optional and a given service can be implemented at one of several distinct levels, corresponding to the inclusion of one or more capability sets. The minimum capability set corresponds to the simplest possible level, which also remains sensible and coherent. At least this set is included in every implementation of a given service.

The PUS should be viewed as a “Menu” from which the applicable services and service-levels are selected for a given mission. This selection process is repeated for each on-board application process, since each application process is designed to provide a specific set of tailored services.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 13701:2001, *Space systems — Glossary of terms*.

EN 14737-1:2004, *Space engineering — Ground systems and operations — Part 1: Principles and requirements*.

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in EN 13701:2001, EN 14737-1:2004 and the following apply.

3.1.1

additional capability set

set of service capabilities, service requests and reports that a service may optionally provide

3.1.2

application process

entity, uniquely identified by an application process ID (APID), capable of generating telemetry source packets and receiving telecommand packets

NOTE 1 Application processes are the providers of the services defined within this document.

NOTE 2 An application process can be implemented in software, firmware or hardware.

NOTE 3 There are no restrictions on the mapping between application processes and the usual functional subdivision of a satellite into subsystems and payloads. In a simple satellite, there can be a centralized application process which provides a number of “dumb” platform subsystems with collection of housekeeping data, the distribution of device commands, on-board operations scheduling or on-board monitoring. In a more complex satellite, each subsystem and payload can be served by its own independent application process.

NOTE 4 A microprocessor can host one or more application processes. An application process (presumably a rather complex one) can be distributed across two or more microprocessors.

3.1.3

control loop

mechanism to maintain a parameter, or set of parameters, within defined limits

NOTE A control loop is comprised of a set of measurements and responses (commands) related according to a function, algorithm or set of rules. In the case of a satellite, if the characteristic dynamic behaviour of the parameter(s) being controlled is such that the measurement frequency is high and the control response time short, then these control loops are implemented on-board (e.g. attitude control).

3.1.4

device command

telecommand which is routed to and executed by on-board hardware, e.g. a relay-switching telecommand or a telecommand to load an on-board register

3.1.5

memory

on-board memory area, either main memory or mass memory

3.1.6**minimum capability set**

set of service capabilities, service requests and reports that are supported by all implementations of a service

3.1.7**parameter**

lowest level of elementary data item on-board

NOTE 1 A parameter has a unique interpretation.

NOTE 2 Traditionally, each distinct telemetry parameter is assigned a “parameter ID” for ground identification purposes. Within this document, the concept of “parameter number (#)” is introduced for the on-board “identification” of parameters. The parameter# is unique across a given spacecraft and there is a one-to-one correspondence between parameter ID and parameter#.

3.1.8**parameter validity**

Boolean value (true or false) that determines whether a parameter value is meaningful

NOTE A parameter is valid if the appropriate well-defined conditions prevail under which it is meaningful to interpret its telemetered value.

EXAMPLE The angular output of a gyro can only have a valid engineering meaning if the power to the gyro is “on”; at other times the output can be random (or should not be relied upon). Such a parameter is deemed conditionally valid, with its validity determined from the power status.

3.1.9**platform subsystem or payload**

combination of units within the satellite platform or the payload that fulfils a well-defined and self-contained set of on-board functions

3.1.10**service**

set of on-board functions offered to a service user that can be controlled and monitored (e.g. by a ground system) through a well-defined set of service requests and reports

NOTE 1 A service can interact with other on-board functions for the execution of its activities.

NOTE 2 See minimum capability set.

NOTE 3 See additional capability set.

3.1.11**service data unit**

packet data field part of a service request, response or indication, i.e. the complete packet excluding the packet header fields

3.1.12**service provider**

on-board application process which executes the service activities and which is the destination of the service requests and the source of the service reports

NOTE 1 Distinct instances of one and the same service can be provided by several on-board application processes.

NOTE 2 A given application process can provide several services.

3.1.13**service report**

data exchange between a service provider (the report initiator) and a service user to provide information either relating to the execution of an activity initiated by the user (e.g. notification of completion of execution) or

relating to a meaningful event which occurred during the internal execution of a continuous service activity (in the latter case, the report is a provider-initiated report)

NOTE 1 The initiation of a service report by an on-board service provider results in the sending of one or more telemetry source packets to a service user.

NOTE 2 When the only purpose of a service request is to obtain some service data, the service report used to pass the data is called a "service response". In all other cases, the service report is called a "service indication". A service indication is used to report on the progress or completion of the execution of a user-initiated activity and on events occurring during the execution of a continuous service activity.

NOTE 3 Clause 22 contains a summary of requests, responses and indications for the standard services defined in this document.

3.1.14

service request

data exchange between a service user (the request initiator) and a service provider to initiate the execution of a particular activity

NOTE 1 The invocation of an activity by a service user results in the transmission of one or more telecommand packets to the service provider, the reception of which initiates the corresponding activity.

NOTE 2 If several users can interact with a service, then additional information is passed to the recipient service provider to identify the user invoking the activity. This information is used by the service provider to route the related service report(s) back to the user who invoked the activity.

NOTE 3 All user-initiated activities are optionally confirmed.

3.1.15

service user

entity (e.g. a ground system) that initiates the service requests and receives the service reports

NOTE The specification of the activities performed by a service user (e.g. to process a service report) is beyond the scope of this document.

3.1.16

software function

software entity implementing a mission-specific on-board function that can be controlled by commands from the ground

3.1.17

sub-schedule

distinct subset of an on-board command schedule in the sense that it can be independently started and stopped

NOTE Commands within the same sub-schedule can be interlocked with each other.

3.1.18

supporting service

service used only in the context of another service, i.e. its service requests are invoked as a consequence of executing the user-initiated or continuous activities of the supported service

3.1.19

telecommand

data packet transported by a telecommunications system to give instruction or data to equipment located on-board a spacecraft

NOTE In line with common usage within the space domain, the terms command and telecommand are used interchangeably within this document.

3.1.20**telecommand function**

operationally self-contained control action which can comprise, or invoke, one or more lower-level control actions

3.1.21**vital telecommand**

telecommand which is not hazardous, but which, if not executed at the expected time, can cause a significant degradation to the mission

3.2 Abbreviated terms

The following abbreviated terms are defined and used within this document.

Abbreviation	Meaning
Ack	acknowledgement
AD	applicable document
ADT	aborted data transfer
AOCS	attitude and orbit control system
AOS	acquisition of signal
APID	application process ID
ASCII	American Standard Code for Information Interchange
CCSDS	Consultative Committee for Space Data Systems
CDS	CCSDS day segmented
CLCW	command link control word
CLTU	command link transfer unit
CPDU	command pulse distribution unit
CRC	cyclic redundancy code
CUC	CCSDS unsegmented code
EGSE	electrical ground support equipment
ESA	European Space Agency
GPS	global positioning system
ISO	International Organization for Standardization
LEO	low Earth orbit
LOS	loss of signal
LSB	least significant bit
MAP	multiplexed access point
MSB	most significant bit
OBT	on-board time
PAC	packet assembly controller
PC	parameter code
PCS	packet check sequence
PEC	packet error control

PFC	parameter format code
PSS	procedures, specifications, standards
PTC	parameter type code
PUS	packet utilization standard
RAM	random access memory
RF	radio frequency
RID	report identification
SAU	smallest addressable unit
SDU	service data unit
SGICD	space-ground interface control document
SID	structure identification
SSC	source sequence count
UTC	universal time coordinated
VC	virtual channel

4 PUS operations concepts

4.1 Introduction

This clause describes operations concepts that form the basis for the services specified in the remainder of this document. Not all of the operation concepts (nor all aspects of a given concept) are applicable to a given mission. Applicability of the concepts is decided on a case-by-case basis, dependent on the mission operational requirements.

The concepts are presented in the following sequence:

- a) Overall concepts and guidelines for application process and packet design.
- b) Concepts relating to routine operations scenarios, such as telecommanding and telecommand verification, telemetry reporting, on-board software management, on-board operations scheduling, on-board monitoring, on-board storage and retrieval and packet forwarding control.
- c) Concepts that relate to non-routine operations scenarios, such as software maintenance and troubleshooting. This includes memory management, diagnostic data reporting and off-line testing.

Some of these operations concepts are prescribed within higher-level CCSDS packet recommendations; others are developed from experience with past and present space missions.

Some operations concepts, or the corresponding services, introduce quantities for which values cannot be globally defined, but which shall be defined on a mission-by-mission basis. Examples of such quantities are time intervals and sizes of parameter sets. These are termed “mission constants” and are identified within this document in right-angled parentheses and typical values are indicated. These mission constants are also summarized in Annex B.

EXAMPLE <PKTS_NUM_STORED>.

4.2 Application processes

4.2.1 Background

Within the CCSDS packet telemetry and telecommand concept (CCSDS 102.0-B-5 to CCSDS 203.0-B-2), an on-board application process is defined as the source of telemetry source packets and the sink for telecommand packets. A given application process has a unique identifier, namely its application process ID (APID). These reference documents place no particular restrictions on the design of application processes or on the assignment of APIDs for a given mission, except that APID= 2047 (all “ones”) is reserved by CCSDS 102.0-B-5 for the idle packet.

This document imposes two additional rules:

- a) APID=0 shall be reserved for the standard spacecraft time packet;
- b) each command pulse distribution unit (CPDU) shall have a dedicated APID.

4.2.2 The assignment of APIDs for a mission

The process of allocation of APIDs for a given mission shall consider factors such as:

- a) the number of on-board data sources or sinks (e.g. subsystems, payloads);
- b) the virtual channelization scheme for the mission;
- c) the requirements for separation of data from a given source or to a given sink.

Different APIDs shall be assigned to different on-board data sources or sinks and also within the same source or sink where there are packet streams with different bandwidth characteristics and operational significance. This may apply to:

- real-time and deferred data;
- science packets and housekeeping packets from an instrument.

An on-board entity that is both a data source and sink should use the same APID for its telemetry source packets and telecommand packets. Once assigned by the mission control authority, the allocation of APIDs should not be changed for a mission.

4.3 Packet design and routing

4.3.1 Identification of both packet source and destination

For missions with several sources for a given telecommand packet or sinks for a given telemetry source packet (e.g. where CCSDS/PUS packets are used for on-board communication between application processes or where there are multiple ground commanding sources), both the source and the destination of the packet shall be explicitly indicated. For those missions, the “end-to-end” identification of the packet is used for the selective routing, filtering or other processing of packets, either on ground or on-board.

One solution is to assign APIDs to suit this purpose, i.e. to define APIDs that identify both the source and the destination of a packet. However, the limited size of the APID field (11 bits) seriously limits the number of possible source/destination combinations. The source (or destination) ID of the telecommand (or telemetry source) packet should be included in the CCSDS secondary header (referred to in this document as the data field header). This approach is also in line with the guidelines contained in the corresponding CCSDS concept documents (see CCSDS 100.0-G-1, CCSDS 200.0-G-6). If a mission uses both a source and destination ID, the packet source sequence count shall be separately maintained for each source-destination couplet, to avoid ambiguity.

4.3.2 Missing telemetry source packets

The source sequence count (SSC) for telemetry source packets shall be incremented separately for each on-board application process. Where a given APID generates a number of different types of telemetry source packet, and there is a discontinuity (for whatever reason) in the SSC, there is potential ambiguity concerning which packets are missing. Where it is important for a mission to identify lost telemetry source packets at a level of granularity lower than the APID, this document makes provision for including a secondary counter (termed a packet sub-counter) within the data field header.

4.3.3 Packet size

Taking into account factors such as uplink bandwidth, the duration of ground station passes and the time to recover from telecommand failure, consideration should be given to the definition of a maximum telecommand packet length (<TCPKT_MAX_LENGTH>) that is less than the maximum of 65K octets defined by CCSDS 203.0-B-2. For analogous reasons, a maximum telemetry source packet length may be defined (<TMPKT_MAX_LENGTH>) that is less than the maximum defined by CCSDS 102.0-B-5.

There may be circumstances where a set of data is transmitted (uplink or downlink) that exceeds the maximum packet size, for example, a large memory load or dump or a report of the contents of an on-board command schedule. This data can be sent as a group (TM) or sequence (TC) of packets, using the grouping (sequence) flags provided in the packet primary header. However, this mechanism does not provide security in the event of missing packets. This document therefore defines a capability to transfer large service data units in a more secure manner, on either the uplink or the downlink. This large data transfer capability utilizes packets of the maximum defined size (the last packet can be smaller, of course) with the possibility for acknowledgement of receipt (either intermediate or final). If packets are rejected or missing their re-transmission shall be initiated in order to complete the data transfer.

4.3.4 Packet routing

4.3.4.1 Telemetry source packets

The primary mechanism for telemetry bandwidth control is that of virtual channelization. This mechanism also provides the means to separate on ground data streams for distribution to different destinations. The allocation of data classes to virtual channels should not be changed dynamically during the course of a mission, however some missions may provide an on-board mechanism for the mapping of packets to VCs. In this case a report of the on-board mapping information shall be available.

Selection of telemetry source packets can be performed on the ground for priority routing, on the basis of operational significance, for example, where there is a bandwidth-limited communications link between a ground station and the control centre and forwarding of all the telemetry as it is received from the spacecraft in real time is not feasible. In addition to performing selection on the basis of VC, packets can be selected from within a given VC. This implies that the operational significance of a packet shall be indicated explicitly within the primary header or the data field header of the packet. This document defines fields for this purpose within the data field header (see Clause 5).

4.3.4.2 Telecommand packets

Commands of different operational priority should be assigned to different multiplexed access points (MAP), with the higher priority commands being routed to the priority MAP. With this mechanism, the transmission of a long telecommand packet can be "interrupted" by the transmission of another telecommand packet of higher priority that is issued later in time. This can apply when the packets are destined for different application processes and when they are destined for the same application process.

4.4 Telecommanding

4.4.1 The different levels of control

It is expected that the routine operations of a spacecraft are effected at a high-level, utilizing the specialized services and functions of an application process, as defined in this document, or by mission-specific services. However, the spacecraft design will include a number of elementary (indivisible) control actions, implemented either in hardware or software. Examples of such control actions include: switching a relay, loading data into a hardware register or starting or stopping a software task.

High-level commands to the spacecraft (service requests, application function commands) can result in the eventual triggering of one or more such elementary control actions, for example, a request to power up an experiment in a given mode can generate a particular sequence of “relay on” commands and software configuration commands. In some cases, there can be several hierarchical layers between the highest and lowest level of control.

Although not used during routine operations, the ground system shall have direct access to all on-board elementary control actions. This ensures that the ground system has a complete set of capabilities when performing contingency operations or troubleshooting, for example, to by-pass an on-board function that can be malfunctioning.

A telecommand function is an operationally self-contained control action. A telecommand function can comprise, or invoke, one or more lower-level control actions. The design of each telecommand packet should ensure that it contains only one telecommand function.

4.4.2 Device commands

One particular group of elementary control actions are routed to, and executed by, on-board hardware. These are termed device commands and comprise hard-wired commands as well as On-Off and register load²⁾ commands.

A number of different mechanisms exist for routing these device commands:

- a) They can be uplinked at telecommand segment level, by embedding one device command within a single telecommand segment. When the telecommand segment is received on-board, the device command is extracted and routed directly to the device. This mechanism can be used, for example, to by-pass a MAP or a packet assembly controller (PAC) or when an application process used to distribute device commands is not available (e.g. in the event of a software or processor anomaly).
- b) Most spacecraft include one or more command pulse distribution units (CPDUs), which are devices located within the decoder(s) that issue command pulses of specified duration. Each CPDU is addressed by means of a dedicated application process and is used for the distribution of vital (or “high-priority”) spacecraft commands with the minimum of on-board software intervention.
- c) Device commands can be uplinked within a telecommand packet to an application process that shall then extract the command and route it to the device.

Certain device commands can be grouped or sequenced in such a way that they constitute a higher-level command function. In order to ensure that either the complete function is executed, or nothing at all, the capability should exist to “pack” a number of device commands of the same type (On-Off or register load) within a single telecommand packet.

²⁾ In the days of PCM telecommanding, these were called memory load commands. This term is now reserved for the loading of on-board (RAM) memories.

4.4.3 Telecommand verification

The basic concept for telecommand verification is that the on-board system should provide feedback on the execution of a given telecommand at whichever stages are meaningful for that telecommand. An example of a telecommand execution profile is shown in Figure 1. The delay shown between transmission of the telecommand from the ground and acceptance by the “destination” application process can correspond to propagation delay (deep-space missions) or the on-board storage of the telecommand prior to release to the application process.

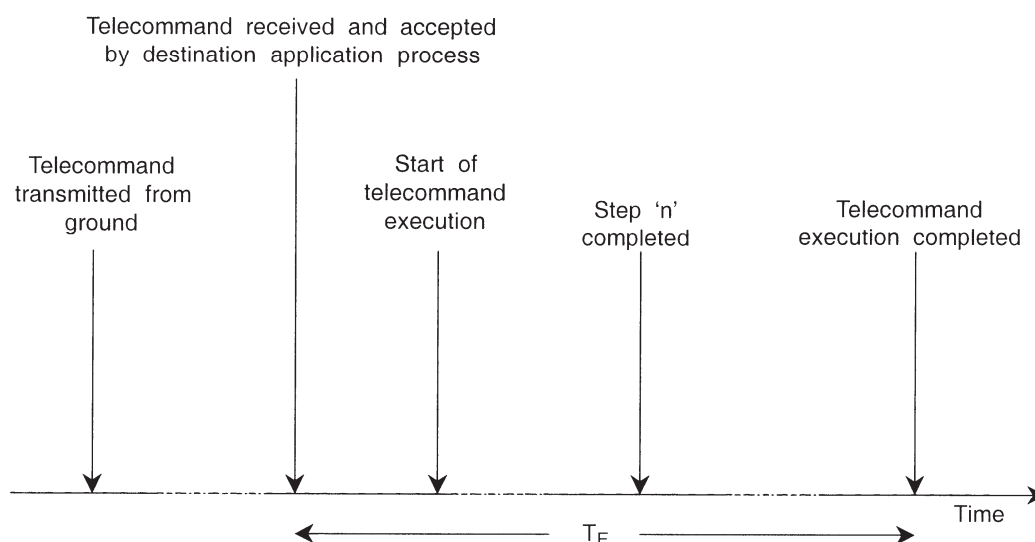


Figure 1 — Example of a telecommand execution profile

For an immediate telecommand, a separate notification of telecommand acceptance need not be given. For a short execution duration telecommand, only one execution verification report packet (combining start and completion status) can be sufficient. Requirements for the different stages of telecommand verification are summarized in Table 1 for telecommands with different execution characteristics:

Table 1 — Telecommand acceptance and execution telemetry

		Immediate command	Delayed execution command
Short duration execution $T_E < 10 \text{ s}$		$A + E_t$ or E_t	$A + E_t$
Long duration execution $T_E > 10 \text{ s}$		$A + E_s + E_p + E_t$ or $E_s + E_p + E_t$	$A + E_s + E_p + E_t$
Where A = Notification of telecommand acceptance E = Execution report: E_s = Execution start E_p = Execution progress (one or more reports) E_t = Execution complete			

Failure of telecommand execution at each verifiable stage should always be reported. However, the selection of which reports of execution success are generated for each individual instance of the telecommand depends on the prevailing operational scenario (e.g. whether or not there is ground coverage). For this purpose, this document has defined an "Ack" field within the data field header of a telecommand packet.

The existence of the capability for different stages of telecommand verification does not imply that every telecommand shall be verifiable at all stages. Those stages that can be verified shall be specified for each telecommand and the ground system shall be configured to ensure that only verification of those stages can be requested.

Figure 2 shows the different layers of the telecommand system; the layers below the dotted line are subject to the CCSDS recommendations (CCSDS 201.0-B-3 to CCSDS 203.0-B-2). This document addresses the application process layer, above the dotted line.

The packets notifying telecommand acceptance and telecommand execution verification and any other telemetry source packets used for end-to-end verification of telecommand execution represent progressively higher level protocols within the application process layer.

Telecommand verification reports shall unambiguously identify the command being verified and, in the case of failure, shall contain ancillary data to assist in the identification of the nature and reason for the failure.

This document defines a number of standard failure codes for the acceptance stage of telecommand verification. Failure codes for later stages are defined on a mission-specific basis.

The telecommand verification concept defined here refers to telecommands originating from ground. This includes commands pre-loaded into an on-board schedule for release to an application process at a later time (see 4.7). However, the concept does not apply to commands generated autonomously on-board; the notification to the ground system of the invocation (and any subsequent verification) of these commands shall be done using event reporting (see 4.5.2).

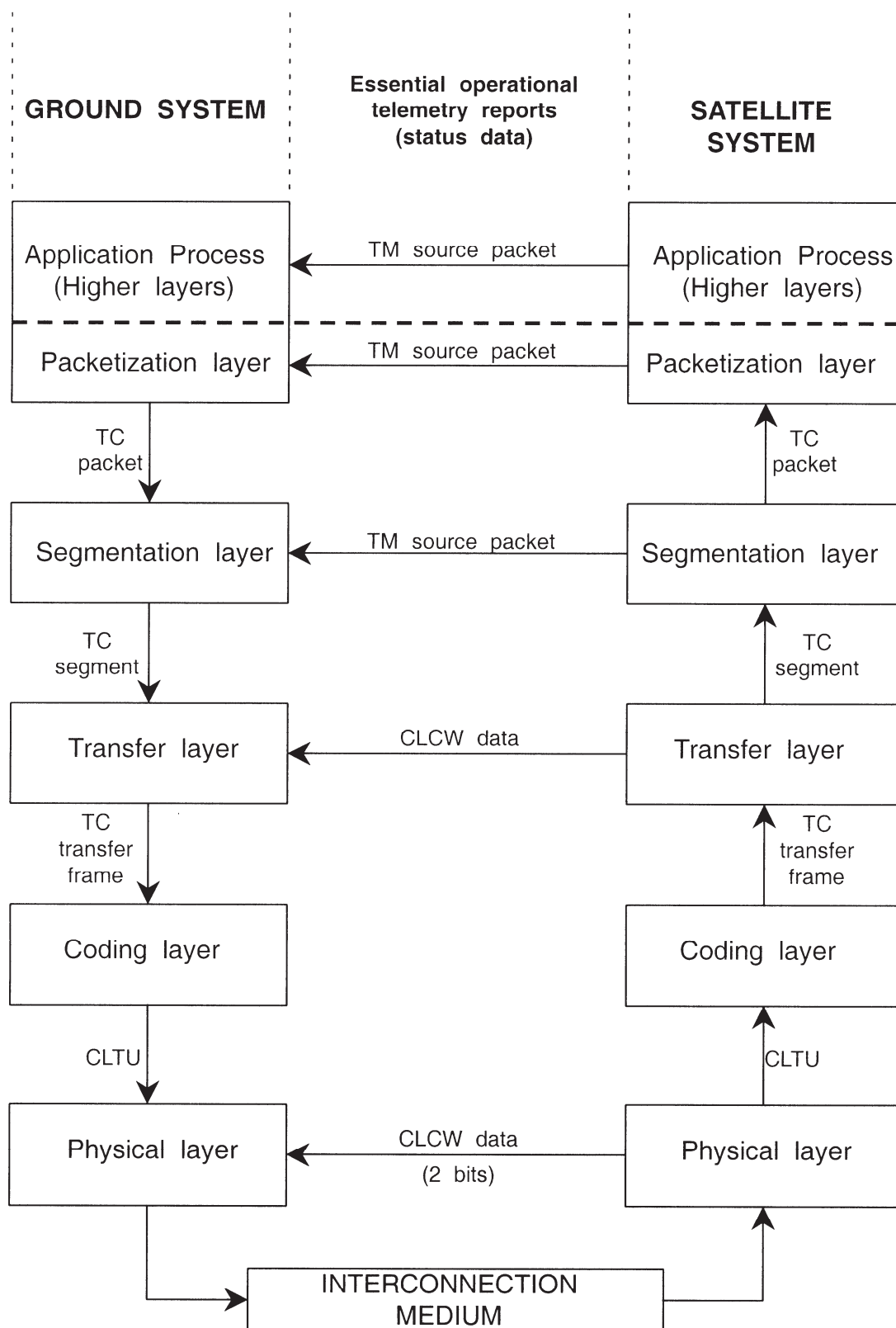


Figure 2 — Telecommand system layers

4.5 Telemetry reporting

4.5.1 Housekeeping data reporting

Every spacecraft has a number of predefined housekeeping packets, whose content, layout and generation frequency is specified by the spacecraft manufacturer. The criteria used in designing these packets shall ensure that:

- a) the spacecraft status is reported in a complete and unambiguous manner;
- b) the frequency of packet generation is compatible with any requirements on response time from control loops that are closed via the ground.

An appropriate reserved downlink bandwidth shall be provided for these packets (even if it is not always used).

To adapt to changing operational requirements, the capability shall be included to modify both the data content and generation frequency of housekeeping packets during the mission. Different data reporting schemes can be used due to factors such as:

- what was considered at the design stage, in terms of data content and packet frequency, can be less than optimal after some period in-orbit;
- sensor characteristics and equipment performance can evolve with time;
- there can be on-board failures and anomalies.

The normal mechanism for reporting of housekeeping data is to downlink housekeeping telemetry packets on a periodic basis. However, since the ground system is essentially only interested in being informed of any changes in status or performance, an additional mode of reporting is defined in which housekeeping packets are only generated when parameters within them have changed in value. In this filtered reporting mode, certain parameters can be filtered out from this change comparison process (e.g. “noisy” analogue parameters that continuously change value). In addition, the threshold for the change in value that triggers the generation of the packet can also be specified by ground. Since this filtered mode can result in long passages of time where no packet is generated, the concept shall support the production of a packet after a specified timeout period (the packet is generated periodically, but with a much longer period).

The capabilities to define new housekeeping packets and for generating filtered reports are optional. The minimum implementation of the housekeeping function consists of periodic reporting of predefined packets.

4.5.2 Event reporting

In addition to periodic reporting of predefined parameter sets, the spacecraft shall report whenever an event of operational significance is detected on-board, e.g. predefined milestones in the normal progress of activities initiated either from ground or autonomously on-board from the detection of anomalies by a failure detection and recovery function.

Because such event reports are infrequent (and hence consume low bandwidth) and are often unexpected, they should be designed to convey information about the event clearly and unambiguously. This can be achieved in two ways:

- a) flagging the severity of the report in the data field header, in order to facilitate ground system detection and routing;
- b) including an appropriate set of auxiliary data within the report itself so that the ground system can understand the context and cause of the on-board event.

4.5.3 Statistical data reporting

Another mechanism for reducing the quantity of engineering data reported to the ground system is to evaluate statistics (mean, min., max., standard deviation) on-board for a parameter set and to report only the results of this evaluation. This mechanism is appropriate for missions with limited ground coverage (e.g. low-earth orbiters), where the report can be used to provide a summary of the behaviour of parameters during the previous period of "no contact".

In the minimum implementation of this function, the set of parameters and the attributes being evaluated shall be predefined and the statistics shall be evaluated continuously. However, optional extensions include the capability for the ground system to modify the set of parameters or their attributes and the intervals of time over which the statistics are evaluated.

4.5.4 Time information

Annex C defines a mechanism for the spacecraft to report its time reference to the ground system, via the standard spacecraft time packet, and procedures to be used on ground to perform a correlation between on-board time (OBT) and ground time. The frequency of generation of the standard spacecraft time packet can be fixed for a mission, or it may be modified in-orbit (e.g. if the accuracy is a function of mission phase or experiment operation mode).

The knowledge of this correlation enables the ground system to reconstitute accurately the on-board time of other information reported in the telemetry, such as the time of occurrence of an event or the on-board sampling time of a parameter. The timing information may be either:

- a) explicitly reported, i.e. the event or the parameter sampling is time-stamped; or
- b) deduced from implicit knowledge of time offsets or on-board sampling sequences - in this case, the on-board time reference shall be reported within the data field header of the packet.

It is assumed that any on-board synchronization with a master clock is done autonomously; however, the possible mechanisms for on-board time distribution or synchronisation are beyond the scope of this document.

Missions using the global positioning system (GPS) for their on-board time reference have different requirements for the exchange of time information between the satellite and the ground segment. This concept is not covered by this document.

4.6 Software management

This document identifies a number of standard services that can be provided by one or more application processes. These services are standardized because they relate to on-board activities and capabilities that are generic in nature, both across many missions, as well as across different application processes for the same mission.

In addition, a given application process can support one or more software functions that are invoked from the ground. These functions relate to the specialized role assigned to the application process on-board the spacecraft, for example, responsibility for controlling the operation of a payload instrument or an AOCS subsystem. The nature of this control can vary quite considerably and is not prescribed or constrained in any way by this document. In principle, it covers all on-board nominal operations for subsystems and payloads including predefined mode change operations.

These functions can be implemented as mission-specific services, with their own request (TC) and report (TM) data structures and service models. However, they can also be implemented using a generic, high-level function service. In this concept, the function itself is specialized (e.g. configure experiment mode, set AOCS control mode), but the associated packet structure is standardized.

At whichever level the ground system interacts with on-board software (service-level, function-level), all communications between the ground system and on-board software shall be effected by means of telecommand and telemetry source packets specifically designed for the purpose. This communication should not be achieved by using general-purpose memory write (memory load) and memory read (memory dump) packets.

4.7 On-board operations scheduling

There are two scenarios where the capability for the on-board execution of operations that have been loaded in advance from the ground shall be implemented:

- a) Those missions that perform operations outside of ground contact because of limited ground station visibility (e.g. LEO spacecraft) or signal propagation delays (e.g. deep-space probes).
- b) Those missions whose operations concept is to minimize the dependency on the ground segment. Thus, a geostationary telecommunications or meteorological mission can perform all of its routine operations in this manner, even though the spacecraft is permanently in view of a ground station. This approach potentially increases the availability of operational services or mission products, since the continuous availability of the uplink is eliminated.

The simplest form of on-board operations scheduling consists of storing time-tagged commands that have been loaded from ground and releasing them to their destination application process(es) when their on-board time is reached. This is a purely open-loop function with no feedback being generated by the destination application process. This document includes the following (optional) scheduling functions:

- c) The command schedule may be partitioned into sub-schedules. These sub-schedules can be independently controlled (started, stopped) and can be used to group commands for different destinations (e.g. instruments) or commands constituting a related “sequence”. There is no size limit on a sub-schedule (except the size limit of the schedule as a whole) since it is only a logical partition of the command schedule.
- d) Commands can be interlocked, i.e. the release of a given command can be preconditioned on the successful (or unsuccessful) execution of another command that was released at an earlier time from the schedule. A given command can only be interlocked with a single earlier command. This implies a feedback from the destination application process (a sort of on-board telecommand verification loop). This capability provides simple preconditioning of command release (e.g. for safety). However, it also provides for alternative sequences of commands to be sent from the schedule depending on the success or failure of commands released at an earlier time (IF, THEN...ELSE capability). Interlocking is implemented at sub-schedule level, i.e. commands which are interlocked reside in the same sub-schedule.
- e) Because operations are often rescheduled in real life operations, commands can be time-shifted “en-bloc”, by supplying a delta-time from the ground system. This avoids the complicated procedure (often used in the past for time-tagged queues) of having first to delete them from the command schedule and then to reload them with new execution times.
- f) A relative (rather than absolute) execution time can be attached to a command with an indication of an “event” to which this relative time relates. These events can be mission-specific in nature; however a number of standard events are defined, such as the starting of the (overall) command schedule or the sub-schedule in which the command resides or the execution time (as opposed to the release time) of the command with which this command is interlocked.

4.8 On-board monitoring

Some missions use an on-board parameter monitoring capability that is functionally similar to the traditional monitoring performed on the ground. The primary purpose of this function is to reduce downlinking all the low-level housekeeping data to the ground system, which in turn implies that the ground system is continuously available to monitor this data.

The on-board monitoring function can be provided by one or more application processes and the maximum number of parameters which can be simultaneously monitored per application process is `<MONLIST_MAX_CHECKS>`. The following types of check can be applied (depending on the parameter and its type):

- limit-check, i.e. a check that the parameter value lies within a specified range;
- expected-value-check, i.e. a check against a specified value;
- delta-check, i.e. a check of the change in value of a parameter against a pair of thresholds.

A parameter can be conditionally valid, in which case its validity shall be determined from the value (“TRUE” or “FALSE”) of an associated Boolean “validity parameter”. A conditionally valid parameter shall only be checked if it is valid. Several pairs of limits or expected values or delta checks can be specified per parameter, to a maximum of `<MONLIST_MAX_CHECKS>`, each with an associated “check selection parameter”. The parameter shall be independently checked against each check where the check selection parameter is currently “TRUE”.

Parameters and their monitoring attributes can be added or deleted from the monitoring list and control over the monitoring process can be effected both globally (for the application process) and at the level of individual parameters. Thus, the monitoring of a subset of parameters can be temporarily disabled at the same time the remainder of the parameters in the list are enabled.

Check transition reports are generated to notify the ground system of all check transitions. A check transition occurs whenever:

- a parameter fails one of its enabled checks;
- a parameter passes a check having previously been in a failed state (e.g. it is back within limits);
- a checked parameter becomes valid;
- a check becomes enabled.

Optionally, an event report can be generated if a monitoring violation occurs, for example, if a parameter value exceeds the high limit specified for it. This event report can then be used to trigger the initiation of an on-board recovery action (see 4.10).

In the simplest on-board implementation of this function, the set of parameters to be monitored is predefined and the monitoring function can merely be enabled or disabled. A given parameter shall be checked only against a single check and no parameter validity checking is performed. Successive extensions to the function shall provide:

- the concept of multiple checks;
- the concept of parameter validity;
- the possibility to add or delete or modify parameters and their monitoring attributes in the monitoring list;
- reporting of the current monitoring list and out-of-limits list;
- the generation of an event report to trigger an on-board recovery action.

4.9 On-board operations procedures

Many operational activities shall be performed on-board using standard services, mission-specific services or other software functions. However, a mission may also have a set of on-board operations procedures that can be loaded from the ground system, which then has control over their subsequent execution. These procedures are similar in concept to the standard procedures that are executed in a stepwise manner on the ground. In principle, these procedures can also be invoked autonomously on-board, e.g. as the result of the on-board detection of a specified event.

The operations language used for the specification of these procedures is beyond the scope of this document. However, a procedure shall have a structure comprised of logical sub-elements or steps. In addition to starting and stopping a procedure, it shall be controllable at a lower level by suspending and resuming execution at the level of its component steps.

In the same manner as for operations procedures that are executed in the ground system, run-time parameters can be provided to an on-board operations procedure, either at the time that it is started or during execution.

4.10 Attaching actions to on-board events

As an extension to the on-board capability for detecting events and reporting them asynchronously to the ground system (see 4.5.2), an action can be defined that is executed autonomously on-board whenever a given event is detected.

The class of events that can result in such an action are those events that also give rise to an event report. The associated action may be a telecommand of any type, i.e. a standard telecommand packet (as defined in this PUS) or any mission-specific telecommand.

A telecommand can initiate one of several possible types of action, including:

- directly reconfiguring on-board hardware;
- starting an on-board operations procedure (see 4.9);
- starting a command sub-schedule (see 4.7), which can contain a predefined command sequence.

4.11 On-board storage and retrieval

A capability to store packets on-board (e.g. in a mass memory) and to dump them, on request, to the ground should be considered under the following circumstances:

- a) Ground station coverage is intermittent. In this case, the on-board storage capacity shall be sized to store all packets generated on-board for spacecraft monitoring and control purposes, for a duration at least

equal to the longest non-coverage period plus a mission-dependent margin `<PKT_STORAGE_TIME>`. These packets shall then be retrieved during subsequent ground station passes according to a selection strategy based upon the operational significance of the stored packets.

EXAMPLE 1 Access to anomaly report packets is needed on the ground with shorter delays than routine status reporting packets.

For these types of mission, there can also be user-generated requirements for the on-board storage and retrieval of payload data.

- b) To recover lost packets. For missions with continuous ground coverage, the loss of one-shot packets (i.e. event-driven or request-driven packets) can be remedied by the short-term on-board storage of the last `<PKTS_NUM_STORED>`, typically the last 100. Telecommand verification packets are request-driven in this context. The ground system can detect that a packet has been lost by checking the source sequence count or the packet sub-counter (see earlier) for a given application process.

The storage function (service) can be provided by a different application process from that which generates the packets.

EXAMPLE 2 There may be a centralized storage and retrieval function that serves all on-board application processes.

For this reason, the function is split into two distinct parts: one part that performs the selection of packets to be stored and a second part that performs the control of the storage and retrieval functions. The latter part shall be located with the application process(es) that manages the packet store(s).

Depending on the on-board design, individual packet stores can be either circular, where the oldest packets are overwritten when the packet store is full (this is the only option for the lost packet recovery store), or bounded, where storage terminates when the packet store is full.

The storage of packets shall not be interrupted if the ground system requests a retrieval or deletion from an on-board packet store.

Retrieval from a packet store over a specified (storage) time window, or for a specified range of packets, should be provided.

In the simplest implementation of the storage and retrieval function, there is no selection possibility for the packets to be stored, but the storage can be enabled and disabled. The extended capabilities of this service include:

- 1) adding and deleting packets to or from storage selection;
- 2) reporting of packet storage selection;
- 3) time-stamping of stored packets;
- 4) deletion of packets from a packet store.

4.12 Telemetry generation and forwarding

The generation, on-board routing and transmission to the ground of telemetry source packets utilizes various resources, such as on-board processing capacity, on-board communications networks and downlink bandwidth, which are not unlimited. The generation and forwarding of telemetry source packets shall therefore be designed such that these resources are only used when actually needed, i.e. appropriate controllability of the generation and forwarding functions shall be provided.

Where applicable, generation control shall be effected at the level of the individual originating service, i.e. it is not a centralised function.

However, forwarding control shall be effected at a higher level, either:

- a) at the level of the originating application process (in this design option, a packet may be generated by a given service but not forwarded by the application process); or
- b) by a centralised application process which receives all telemetry source packets to be transmitted to the ground system and which controls the routing on the downlink.

4.13 Memory management

Direct management by the ground system of on-board memory areas is not a routine operational activity, but an “off-line” activity in support of troubleshooting operations or for the purposes of undertaking maintenance of on-board software. For such purposes, read access should be available to any on-board memory and the ground system should have the means to load any on-board RAM area.

To provide flexibility when loading (or dumping) memory, the capability should exist to load (or dump) either a contiguous area of memory or to load (or dump) several dispersed areas at the same time; also known as a “scatter-load” (or “scatter-dump”). Optionally, on-board protection shall be provided to ensure that corrupted data cannot be loaded into memory. This shall be achieved by calculating a checksum over the data area(s) to be loaded and comparing this with a checksum that is provided by the ground system in the load packet.

Where implemented, the on-board end-to-end verification of a memory load shall consist of confirming that the data have been correctly loaded into their destination memory, by reading them back from the memory and comparing them with the original load data.

To check the content of an on-board memory (e.g. because a corruption is suspected), the capability shall be provided to request the calculation of a checksum on-board (across a specified memory area) and to send the result to the ground via telemetry. Comparing the checksum to a known reference value avoids the need to dump the suspected area to ground to check its content against a ground-based memory image.

Within the CCSDS recommendations there is a file delivery protocol (the CFDP, see CCSDS 727.0-R-5) designed specifically for use with on-board memories that are organized as “Filestores”. The CFDP provides the capability for the delivery of files from ground-to-space and vice versa, as well as capabilities for the management of files and directories within the filestore. Missions that use file management may also define their own mission-specific additions to the services defined within this document.

NOTE File delivery concerns the transport of data files between space and ground, regardless of their content, whilst this document defines the syntax and semantics of the data to be transported, regardless of the transport mechanism. As such, the two aspects are complementary.

4.14 Diagnostic mode

For the purposes of anomaly investigation (troubleshooting), a telemetry-sampling setup quite different from that used in normal operations can be defined. A selected set of parameters pertaining to the problem shall be sampled on-board at a higher rate than normal and then telemetered to the ground system. This scenario is termed “Diagnostic mode” and is synonymous with the terms “Dwell mode” and “Oversampling mode”, which have been used in the past. When operating in diagnostic mode, the ground system shall carefully select the extent to which these diagnostic packets shall occupy the downlink bandwidth taking into account the other prevailing operational requirements.

Functionally, diagnostic packets shall have the same characteristics as housekeeping packets, for example, new packets can be defined and their generation can be controlled. However, their usage is quite distinct and they shall therefore be identified separately for routing (and possible storage) and processing purposes. Hence, they have different “operational significance” information within the packet data field header.

All on-board telemetry parameters shall be accessible for sampling and reporting via diagnostic packets. In order to ensure that high-rate sampling can be achieved a minimum sampling interval shall be definable for all on-board parameters (<DIAG_MIN_INTERV>, typically 10 ms to 30 ms). In practice, the sampling interval selected for different subsets of parameters can vary quite considerably. The minimum sampling interval can be used for parameters which can vary rapidly (e.g. AOCs sensor data), whilst for slowly varying parameters (e.g. temperatures), a longer sampling interval (e.g. one second or more) can be used.

4.15 Off-line testing

For troubleshooting or pre-operational validation purposes, an on-board unit or process can be operated in an off-line manner, i.e. not as a part of any active control loop, but nevertheless providing all its normal outputs to the telemetry for analysis on the ground. For these purposes, each application process shall provide a set of specialized (i.e. mission-specific) end-to-end test functions that can be exercised under ground control.

As a minimum, a standard “are you alive” function shall be provided for testing the end-to-end connection between the ground system and an application process.

5 Service specification

5.1 Introduction

Clause 4 identified a number of operations concepts that imply distinct sets of capabilities to be implemented on-board the satellite along with corresponding monitoring and control facilities on the ground.

These sets of capabilities constitute “services”, the monitoring and control of which shall be achieved via a well-defined set of interactions between the provider of the service (i.e. an on-board application process) and the user(s) of the service (e.g. a ground system).

This document contains the specification of those standard services which can be provided by on-board application processes in fulfilment of the operations concepts for satellite monitoring and control.

A service specification defines:

- a) what can be requested of the service;
- b) what can be issued by the service to notify the user of the success or failure of its requests and to report events detected during the execution of the service activities;
- c) what internal activities the service shall perform (to process a request or to detect or process events relevant to the service).

The specification of the information to be maintained by a service and of the activities to be performed by a service are expressed as functional requirements. How these functional requirements are implemented is not enforced by this document (however, the actual implementation of the service state information and activities shall be functionally equivalent).

The following criteria are used for identifying and defining these standard services:

— Commonality

A service should correspond to a group of functionalities applicable to many missions.

— Coherence

The capabilities to be provided by a service should be closely related and their scope should be unambiguously specified. A service should cover all the activities for managing inter-related state information and all activities which use that state information. A service should have minimum and well-defined interactions with other services or software functions.

— Implementation independence

A service should neither assume nor exclude a particular satellite on-board architecture (hardware or software).

This clause contains:

- Conventions which are used within the later service specifications.
- Items which are common between services. This includes the overall structure of telecommand and telemetry source packets, which is defined elsewhere (CCSDS 102.0-B-5, CCSDS 203.0-B-2) and those standard elements of packets which are defined within this document.
- A summary of the standard services defined within this document.

5.2 Conventions

In the following clauses, the specification of a service consists of:

- a) An introduction containing the rationale for, and the scope of, the service.
- b) The service concept describing the continuous activities and the state information to be maintained by the service, where these are not explicitly identified in the corresponding operations concept.

The activity and information concepts correspond to a service implementation which provides the full set of capabilities. However, a given implementation may use an alternative but compatible activity and information design, and only implement those aspects applicable to the minimum capabilities set and the subset of additional capabilities which it supports.

Some services may be constituted of distinct parts which may be implemented in isolation from other parts (and even, in some cases, may be implemented within a different application process). The concept of sub-service is introduced for these distinct service parts.

- c) The definition of the structure and content of the service requests and reports of the user initiated and continuous activities. Each distinct service request and report is uniquely identified by a couplet of numbers. The first pertains to the service to which it belongs (the Service Type) and the second is a unique number within the service (the Service Subtype).

An activity specification contains the following information:

- 1) For a user-initiated activity, the parameters of the service request and the structure of the corresponding service data unit to be placed in the telecommand packet used to transport the request.
- 2) The actions to be executed by the service provider to perform the activity. This includes an indication of error or failure cases wherever identified during the specification of an activity. However, this document does not identify all possible such error or failure cases. Nor does it preclude the identification of implementation-specific errors when a service is designed and implemented for a particular mission.

Many errors relate to checks performed on data contained in service requests. These errors should be reported as telecommand verification failures where the appropriate level of telecommand verification is supported. Otherwise, they should be reported by means of an event report.

- 3) For each report (response or indication) that can be issued during the execution of the activity, the parameters of the service report and the structure of the corresponding service data unit to be placed in the telemetry source packet(s) used to send the report.
- 4) The definition of the minimum and additional capability sets of the service, if any.

The service data units contained within a service request or report are preceded by their Name and their Service Type (x) and Service Subtype (y) in the format (x,y). The structure of the service data unit is presented in a graphical tabular format:

Parameter1	Parameter2	Parameter3
Unsigned Integer	Boolean	Enumerated

The first row of the table gives the name of the parameters of the service data unit. The second row indicates the type of the parameters. The parameter types are defined in clause 23.

Where the presence of a field, or group of fields is optional, this is indicated by the text “Optional” below the corresponding fields in the packet structure diagram. A field, or group of fields is optional if either:

- its presence is determined at the level of the mission, application process or service instance; or
- its presence is determined by the value(s) of one or more preceding fields in the service data unit.

The level at which a given field is optional is explained within the corresponding definition for that field.

The omission of an optional field may imply that the value to be used is known by both the service provider and the service user, for example, the service provider may use a fixed value or a “current value” which can be set by the service user through other means. The service provider may even use the value(s) of other preceding field(s) in the service request or report to access a fixed or modifiable look-up table in which the values are contained.

Where a field, or group of fields, constitutes an entry in a fixed-length or a variable-length array, this is indicated below the table by: “← Repeated N times →”, where N is the number of repetitions within the array. In the case of a variable-length array, N is given explicitly at the start of the array; in the case of a fixed-length array, N is known implicitly for the mission. Where the term “array” is used within this document, this should not be interpreted to imply any particular implementation connotation. Instead, one of the array structure types formally defined in clause 23 is intended.

Following each service data unit, a description is given of each of the contained fields. However, to avoid repetition, this description is not repeated if the same field appears (with the same interpretation) in subsequent service data units.

Within the specification of a service data unit, the following abbreviations are used:

Boolean parameter	“Boolean”
Enumerated parameter	“Enumerated”
Integer parameter	
unsigned integer	“Unsigned Integer”
signed integer	“Signed Integer”
Real parameter	“Real”
Bit-string parameter	
fixed length bit string	“Fixed BitString”
variable length bit string	“Variable BitString”
Octet-string parameter	
fixed length octet string	“Fixed OctetString”
variable length octet string	“Variable OctetString”
Character string parameter	
fixed length character string	“Fixed CharString”
variable length character string	“Variable CharString”
Time parameter	
absolute time parameter	“Absolute Time”
relative time parameter	“Relative Time”
Parameter with a deduced type	“Deduced”
Telecommand packet	“Any TC”
Telemetry packet	“Any TM”
A data structure following the structure rule set #1 defined in clause 23.	“Any”

5.3 Telecommand packet structure

5.3.1 Overview

All telecommand packets shall conform to the structure defined in CCSDS 203.0-B-2. These telecommand packet fields are represented, in this document as follows:

Packet Header (48 Bits)							Packet Data Field (Variable)			
Packet ID				Packet Sequence Control		Packet Length	Data Field Header (Optional) (see Note 1)	Application Data	Spare	Packet Error Control (see Note 2)
Version Number (=0)	Type (=1)	Data Field Header Flag	Application Process ID	Sequence Flags	Sequence Count					
3	1	1	11	2	14					
16				16		16	Variable	Variable	Variable	16

Standardization of the shaded part of the telecommand packet structure is a primary purpose of this document.

NOTE 1 Although the data field header field is indicated as optional in CCSDS 203.0-B-2, this document specifies it to be mandatory for all telecommand packets except for the CPDU telecommands described in Annex D and Clause 7 of this document, which have no data field header. However, for consistency with other telecommand packets, future missions may choose to include a data field header for CPDU telecommands, if this is feasible for their CPDU implementation.

NOTE 2 The packet error control field is not explicitly identified in CCSDS 203.0-B-2. However, this document specifies it to be mandatory for all telecommand packets.

5.3.2 Packet header

Packet ID (16 bits)

Version Number:

By changing the version number, future variations of the telecommand packet structure can be introduced. At present, however, only one version number shall be used: (value = 0).

Type:

This bit distinguishes between telecommand packets and telemetry source packets. For telecommand packets, the type = 1.

Data Field Header Flag:

This indicates the presence or absence of a data field header. With the exception of CPDU telecommands, all telecommand packets shall have a data field header, for which this bit shall be set to 1.

Application Process ID (APID):

The APID corresponds uniquely to an on-board application process which is the destination for this packet. The choice of APID values is mission-specific.

Packet sequence control (16 bits)

Sequence Flags:

The sequence flags are intended for use if a series of packets are sent in a particular sequence. The interpretation of the sequence flags shall be:

- 01 means first packet of a sequence of packets;
- 00 means continuation packet;
- 10 means last packet of a sequence of packets;
- 11 means “stand-alone” packet.

All standard telecommand packets defined within this document are stand-alone packets.

Sequence Count:

This field is provided to identify a particular telecommand packet so that it can be traced within the end-to-end telecommand system. A separate sequence count shall be maintained by each telecommand source for each APID.

When an acknowledgement of a packet is generated (see “Ack” field in the data field header), the packet sequence control field shall be included in the telemetry acknowledge packet as the identifier of the telecommand packet being acknowledged.

Packet length (16 bits)

The packet length field specifies the number of octets contained within the packet data field. The number shall be an unsigned integer “C” where:

$$C = (\text{Number of octets in packet data field}) - 1$$

NOTE The actual length of the entire telecommand packet, including the packet header, is 6 octets longer. The largest telecommand packet length is 65 542 octets.

5.3.3 Packet data field**Data Field Header (variable length)**

The data field header is the subject of definition within this document; its content depends on the nature of the service requests defined in the remainder of this document; however all data field headers shall have the same basic structure, as follows:

CCSDS Secondary Header Flag	TC Packet PUS Version Number	Ack	Service Type	Service Subtype	Source ID	Spare
Boolean (1 bit)	Enumerated (3 bits)	Enumerated (4 bits)	Enumerated (8 bits)	Enumerated (8 bits)	Enumerated (n bits)	Fixed BitString (n bits)

← Optional →
← Optional →

CCSDS Secondary Header Flag:

According to CCSDS 203.0-B-2, this bit shall be set to zero to indicate that the PUS data field header is a “non-CCSDS defined secondary header”.

TC Packet PUS Version Number:

By changing the PUS version number, future variations of the telecommand packet can be introduced. Users of this document shall use version 1 (value = 1).³⁾

Ack:

This field shall be used to indicate which acknowledgements, in the form of telecommand verification packets, shall be sent to the ground to notify acceptance and to verify execution of this telecommand packet. This shall relate only to acknowledgement of successful acceptance and execution, since failure reports are generated by default; all zeros means no success acknowledgements shall be sent.

In addition to informing the on-board application process of the various telecommand verification packets expected, the ground system also knows what stages of verification to expect for this telecommand packet and can report on “packets uplinked but not yet verified”.

³⁾ Version 0 was used by the ESA PUS.

The bit settings corresponding to these stages are as follows:

- 1 (bit 3 of the Ack field set): acknowledge acceptance of the packet by the application process
- 1- (bit 2 of the Ack field set): acknowledge start of execution
- 1-- (bit 1 of the Ack field set): acknowledge progress of execution
- 1--- (bit 0 of the Ack field set): acknowledge completion of execution (whether successful or failed)

A binary value of "1111" in the Ack field indicates that acknowledgement of at least 4 different steps in the telecommand execution is expected in the telemetry.

NOTE Acknowledgement packets are routed only to the source of the original telecommand. This means, for instance, that the ground system only receives acknowledgements for those telecommands that originate from the ground system and not for any telecommands that can be generated autonomously on-board.

Service Type:

This indicates the service to which this packet relates.

Service types 0 to 127 shall be reserved for this document, service types 128 to 255 are mission-specific.

Service Subtype:

Together with the service type, the subtype uniquely identifies the nature of the service request constituted by this telecommand packet.

Within standard services, subtypes 0 to 127 shall be reserved for this document, subtypes 128 to 255 are mission-specific. Within mission-specific services, all subtypes (0 to 255) are available for mission-specific use.

The definition of service type and subtype is unique across all application processes.

Source ID:

This field indicates the source of the telecommand packet.

EXAMPLE Different control centres on the ground or a given on-board source.

The length of this field is known implicitly for the mission.

This field shall be systematically omitted if the mission has only one telecommand source or has no requirement to distinguish between sources.

Spare:

Spare bits shall be introduced, in order to pad the data field header to an integral number of words (octets or longer, if, for example, the on-board processors are based on a 16-bit, 24-bit, 32-bit or 64-bit architecture).

All spare bits used for padding purposes shall be set to zero.

Application Data (variable length)

The telecommand application data constitutes the data element of the service requests from the service user (e.g. the ground system).

Spare (variable length)

In order that the overall packet size is an integral number of words, spare bits may be used for padding purposes at the end of the application data.

Packet Error Control (PEC)(16 bits)

The packet error control field transports an error detection code that is used by the receiving application process to verify the integrity of the complete telecommand packet. The type of the PEC (either a checksum or CRC) is fixed for the complete mission for each application process and is defined by the mission constant <TC_CHECKSUM_TYPE>.

5.4 Telemetry source packet structure

5.4.1 Overview

All telemetry source packets shall conform to the structure defined within CCSDS 102.0–B-5. These telemetry source packet fields are represented in this document as follows:

Packet Header (48 Bits)							Packet Data Field (Variable)			
Packet ID				Packet Sequence Control		Packet Length	Data Field Header (Optional) (see Note 1)	Source Data	Spare (Optional)	Packet Error Control (Optional)
Version Number (=0)	Type (=0)	Data Field Header Flag	Application Process ID	Grouping Flags	Source Sequence Count					
3	1	1	11	2	14					
16				16		16	Variable	Variable	Variable	(see Note 2)

Standardization of the shaded part of the telemetry packet structure is a primary purpose of this document.

NOTE 1 Although the data field header is indicated as optional in CCSDS 102.0–B-5, this document specifies its inclusion for all telemetry source packets except for the “spacecraft time source packet” described in Annex C and Clause 13 of this document, which has no data field header. However, for consistency with other telemetry packets, future missions may choose to include a data field header for the spacecraft time source packet, if this is feasible for their on-board implementation.

NOTE 2 This (optional) field is not defined in CCSDS 102.0–B-5, but is introduced in this document. When present, it shall be 16 bits in length.

5.4.2 Source packet header

Packet ID (16 bits)

Version Number:

By changing the version number, future variations of the telemetry source packet structure can be introduced. At present, however, only one version number shall be used: (value = 0).

Type:

This bit distinguishes between telecommand packets and telemetry source packets. For telemetry source packets, the type = 0.

Data Field Header Flag:

This indicates the presence or absence of a data field header. With the exception of the spacecraft time source packet, all telemetry source packets shall have a data field header, for which this bit shall be set to 1.

Application Process ID (APID):

The APID corresponds uniquely to an on-board application which is the source of this packet. The choice of Application Process ID values is mission-specific.

APID = 0 is reserved for the time packet.

APID = “1111111111” (eleven ones) is reserved for idle packets.

Packet Sequence Control (16 bits)**Grouping Flags:**

The grouping flags shall be used when a number of telemetry source packets originating from the same application process are sent in a group. The interpretation of the grouping flags shall be:

- 01 means first packet of a group of packets;
- 00 means continuation packet;
- 10 means last packet of a group of packets;
- 11 means “stand-alone” packet.

All standard telemetry source packets defined within this document are stand-alone packets.

Source Sequence Count:

A separate source sequence count shall be maintained by each APID and shall be incremented by 1 whenever it releases a packet. If the application process can send distinct packets to distinct destinations using the optional Destination ID field shown below, then a separate source sequence count is maintained for each destination. Therefore the counter corresponds to the order of release of packets by the source and enables the destination (e.g. the ground system) to detect missing packets. This counter should never re-initialize; however, under no circumstances shall it “short-cycle” (i.e. have a discontinuity other than to a value zero). The counter wraps around from $2^{14}-1$ to zero.

Packet Length (16 bits)

The packet length field specifies the number of octets contained within the packet data field. The number shall be an unsigned integer “C” where:

$$C = (\text{Number of octets in packet data field}) - 1$$

NOTE The actual length of the entire telemetry source packet, including the source packet header, is 6 octets longer.

5.4.3 Packet data field**Data Field Header (variable length)**

The data field header shall be the subject of definition within this document; its content depends on the nature of the service reports defined in the remainder of this document; however all data field headers have the same basic structure, as follows:

Spare	TM Source Packet PUS Version Number	Spare	Service Type	Service Subtype	Packet Sub- counter	Destination ID	Time	Spare
Fixed BitString (1 bit)	Enumerated (3 bits)	Fixed BitString (4 bits)	Enumerated (8 bits)	Enumerated (8 bits)	Unsigned Integer (8 bits)	Enumerated	Absolute Time	Fixed BitString (n bits)
◀ Optional ▶ ◀ Optional ▶ ◀ Optional ▶ ◀ Optional ▶								

Spare:

To maintain symmetry with the telecommand packet data field header, this bit shall be reserved and set to zero.

NOTE In the previous version of CCSDS 102.0–B-5, this was a “CCSDS secondary header flag”.

TM Source Packet PUS Version Number:

By changing the PUS version number, future variations of the telemetry source packet can be introduced. It is expected to introduce a new version in the future for telemetry source packets that contain more than one service data unit.

Users of this document shall use version 1 (value = 1). ⁴⁾

Spare:

Spare bits shall be introduced in order to make up an integral octet. These spare bits shall be set to zero.

Service Type:

This indicates the service to which this telemetry source packet relates.

Service types 0 to 127 shall be reserved for this document, service types 128 to 255 are mission-specific.

Service Subtype:

Together with the service type, the subtype uniquely identifies the nature of the service report constituted by this telemetry source packet.

Within standard services, subtypes 0 to 127 shall be reserved for this document, subtypes 128 to 255 are mission-specific. Within mission-specific services, all subtypes (0 to 255) are available for mission-specific use.

The definition of service type and subtype shall be unique across all application processes and the combination of these fields can be used on the ground to determine the processing priority level.

Packet Sub-counter:

This is a counter that increments each time the application process releases a new packet of this type and subtype. If the application process can send distinct packets to distinct destinations using the optional Destination ID field shown below, then a separate source sequence count shall be maintained for each destination. Whilst the source sequence count enables the destination (e.g. the ground) to detect a transmission gap, this packet sub-counter gives more information on the nature of the particular packets that are missing. This field shall be systematically omitted if this information is not applicable for this application process.

Destination ID:

This field identifies the destination of the telemetry source packet. The length of this field is known implicitly for the mission. This field shall be systematically omitted if the mission has only a single destination (i.e. the ground system) for telemetry source packets or has no requirement to distinguish between destinations.

Time:

The on-board reference time of the packet, expressed in either the CUC or CDS format (as defined in CCSDS 301.0-B-2). The choice between CUC and CDS formats and the resolution of the time field can vary from mission to mission and even between different application processes within the same satellite. For some missions or application processes, this time field may even not be present. The presence or absence of the field and its encoding are explicitly defined by the mission constants <APPL_TIME_CODE> of which there are as many as there are on-board application processes. The representations of the time field are those defined in clause 23 of this document.

If the CDS format is used, the standard CCSDS epoch of 1958 January 1 shall be applicable, see CCSDS 301.0-B-2.

The packet on-board time corresponds to any well-defined time before the sampling of any data within the packet.

⁴⁾ Version 0 was used by the ESA PUS.

Spare:

Spare bits shall be introduced in order to pad the data field header to an integral number of words. These spare bits shall be set to zero.

Source Data (variable length)

The telemetry source data constitutes the data element of the service reports to the service user (e.g. the ground system).

Spare (variable length)

In order that the overall packet size is an integral number of words, spare bits may be used for padding purposes at the end of the source data.

Packet Error Control (optional, 16 bits)

The packet error control field transports an error detection code that can be used by the ground system to verify the integrity of the complete telemetry source packet. The presence of the PEC and its type (either a checksum or CRC) shall be fixed for the complete mission for each application process and defined by the mission constant <TM_CHECKSUM_TYPE>.

5.5 Standard services

The standard services listed in Table 2 are specified within the following clauses. The services can have a minimum capability set and one or more additional capability set(s); this is indicated in the corresponding service clause.

For each service, the minimum capability set corresponds to the minimum coherent implementation of that service.

Table 2 — Standard services specified within this document

Service Type	Service Name
1	Telecommand verification service
2	Device command distribution service
3	Housekeeping & diagnostic data reporting service
4	Parameter statistics reporting service
5	Event reporting service
6	Memory management service
7	Not used
8	Function management service
9	Time management service
10	Not used
11	On-board operations scheduling service
12	On-board monitoring service
13	Large data transfer service
14	Packet forwarding control service
15	On-board storage and retrieval service
16	Not used
17	Test service
18	On-board operations procedure service
19	Event-action service

6 Telecommand verification service

6.1 Scope

The telecommand verification service provides the capability for explicit verification of each distinct stage of execution of a telecommand packet, from on-board acceptance through to completion of execution. In this sense, it is a supporting service for the telecommand packets (service requests) belonging to all other standard and mission-specific services.

Although this service provides the possibility for explicit telecommand verification, there is no implication that all telecommands for a given mission shall necessarily be verifiable at each distinct stage. For many telecommands, the application process can have little or no knowledge of the command execution profile or effects. The extent to which the telecommand verification process is supported on-board for a given telecommand is specified on a mission-specific basis.

This service satisfies the concepts described in 4.4.3 of this document.

Any number of on-board application processes may provide a single instance of the telecommand verification service.

Telemetry source packets and telecommand packets belonging to the telecommand verification service shall be denoted by Service Type = 1.

6.2 Service concept

The following stages of telecommand processing are identified:

- a) Acceptance of the telecommand by the destination application process. At this stage of telecommand processing, all checks to be applied to the telecommand packet prior to the start of execution shall be performed. This shall include verification that the telecommand has not been corrupted (by checksumming the telecommand) and that the application process supports the service or sub-service, requested. This element of the telecommand verification service should therefore be supported for all telecommands.
- b) Start execution of the telecommand. This stage may follow immediately after reception, but for a complex command it may also be delayed pending some level of pre-execution validation. The checks performed at this stage shall include feasibility and validity checks, such as confirmation that the telecommand parameters are correctly encoded, are within their range of values and are valid for the current state of the service.
- c) Progress of execution. The various steps which reflect the progress of execution of the telecommand shall be telecommand-specific in nature.
- d) Completion of execution of the telecommand.

The telecommand verification service shall generate a report if a telecommand fails at any of its identified stages of execution. It shall also generate a report of successful completion of the same stages, if this has been requested in the acknowledgement flags in the telecommand packet header. These reports shall provide auxiliary data for the ground system to fully understand the report (e.g. to identify the nature and cause of a telecommand failure).

Since the distinction between telemetry reports of telecommand verification success and failure is extremely important operationally, different subtypes are defined for these distinct outcomes. This shall facilitate the selection of telecommand verification packets on this basis, for example, for the purposes of priority routing or storage either on-board or on ground.

6.3 Service requests and reports

6.3.1 General

Currently no user-initiated service requests are defined.

6.3.2 Telecommand acceptance

The reports of acceptance of a telecommand packet shall be as follows:

Telecommand Acceptance Report - Success (1,1)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control
2 octets	2 octets

Telecommand Packet ID:

This is a copy of the corresponding fields from the packet header of the telecommand to which this verification packet relates.

Packet Sequence Control:

This is a copy of the corresponding fields from the packet header of the telecommand to which this verification packet relates.

Telecommand Acceptance Report - Failure (1,2)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control	Code	Parameters
2 octets	2 octets	Enumerated	Any
			

Code:

The code indicates the reason for the failure of the telecommand at this verification stage.

At the acceptance stage, the following standard code values are defined:

0 = illegal APID (PAC error);

1 = incomplete or invalid length packet;

2 = incorrect checksum;

3 = illegal packet type;

4 = illegal packet subtype;

5 = illegal or inconsistent application data.

Other values of the code can be application process specific or command-specific (i.e. dependent on combinations of the type, subtype and individual command function). The code is an identifier for the auxiliary information provided with this report, i.e. the parameters field.

Parameters:

The parameters field provides complementary information relating to the particular value of the code field. For full interpretation of failure of a command, knowledge of the nature of the command shall be

available. The parameters field shall conform to the structure rules set #1 and each parameter shall be one of the data types defined in this document (see clause 23).

6.3.3 Telecommand execution started

The reports of start of execution of a telecommand packet shall be as follows:

Telecommand Execution Started Report - Success (1,3)

Telemetry source packet, source data: Same as for subtype 1.

Telecommand Execution Started Report - Failure (1,4)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control	Code	Parameters
2 octets	2 octets	Enumerated	Any

6.3.4 Telecommand execution progress

The reports of progress of execution of a telecommand packet shall be as follows:

Telecommand Execution Progress Report - Success (1,5)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control	Step Number
2 octets	2 octets	Enumerated

Step Number:

This indicates the intermediate step number of the telecommand execution profile whose execution has been completed. The values it can take are telecommand-specific.

Telecommand Execution Progress Report - Failure (1,6)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control	Step Number	Code	Parameters
2 octets	2 octets	Enumerated	Enumerated	Any

6.3.5 Telecommand execution complete

The reports of completion of execution of a telecommand packet shall be as follows:

Telecommand Execution Completed Report - Success (1,7)

Telemetry source packet, source data: Same as for subtype 1.

Telecommand Execution Completed Report - Failure (1,8)

Telemetry source packet, source data:

Telecommand Packet ID	Packet Sequence Control	Code	Parameters
2 octets	2 octets	Enumerated	Any

6.4 Capability sets**6.4.1 Minimum capability set**

The minimum capability set shall consist of the service reports given in Table 3.

Table 3 — Summary of telecommand verification service minimum capabilities

Subtype	Service request, report or capability
1	Telecommand Acceptance Report - Success
2	Telecommand Acceptance Report - Failure

6.4.2 Additional capability sets

The telecommand verification service can implement the additional capabilities given in Table 4. These options can be effected at the level of individual telecommands.

Table 4 — Summary of telecommand verification service additional capabilities

Subtype	Service request, report or capability	Requires subtype
3	Telecommand Execution Started Report - Success	4
4	Telecommand Execution Started Report - Failure	3
5	Telecommand Execution Progress Report - Success	6
6	Telecommand Execution Progress Report - Failure	5
7	Telecommand Execution Completed Report - Success	8
8	Telecommand Execution Completed Report - Failure	7

7 Device command distribution service**7.1 Scope**

The device command distribution service provides the capability for the distribution of:

- a) On-Off and register load commands;
- b) command pulse distribution unit (CPDU) commands for the re-configuration of vital spacecraft functions.

This service satisfies the concepts described in 4.4.2 of this document.

Any number of on-board application processes may provide a single instance of the device command distribution service.

Telemetry source packets and telecommand packets belonging to the device command distribution service shall be denoted by Service Type = 2.

7.2 Service concept

7.2.1 On-Off and register load commands

Following reception of a service request containing On-Off commands or register load commands:

- a) The commands shall be unpacked (if more than one command is present); an On-Off command consists of an On-Off address; a register load command consists of a register address and data to be loaded in the register.
- b) The command(s) shall be reformatted, to the structure applicable by an internal command bus or the destination device driver. The structure of the address field is mission-dependent and may be comprised of several subfields.
- c) The command(s) shall be distributed. If the packet contains more than one command, these shall be distributed in the same sequence in which they occur within the packet.

An example of an OBDH 24-bit command accommodated within this service data unit structure is given in A.3.

7.2.2 CPDU commands

A CPDU is a simple redundant unit, solely accessible from the ground, which can issue command pulses of specified duration on output lines driving vital spacecraft actuators. Each CPDU can have up to 256 output lines.

Each CPDU has a dedicated APID and can receive “CPDU telecommand packets” containing command pulse instructions. A CPDU telecommand packet has no data field header but has a packet error control field and shall be carried inside a single telecommand segment.

Following reception of a CPDU telecommand packet, it shall be checked as defined in Annex D and if no error is found, the CPDU shall process the command pulse instructions in the same sequence as they occur within the packet.

For a given implementation, there shall be a limit on the number of command pulse instructions in a CPDU telecommand packet (at least 12 and at most 120), identified by the mission constant <CPDU_MAX_INSTR>.

7.3 Service requests and reports

7.3.1 General

Currently no service reports are defined.

7.3.2 Distributing On-Off commands

The request for the distribution of On-Off command(s) by means of a telecommand packet shall be:

Distribute On-Off Commands (2,1)

Telecommand packet, application data:

N	Address
Unsigned Integer	Enumerated
← Optional →	← Repeated N times →

N:

The number of On-Off commands which follow ($N > 0$).

This field shall be systematically omitted if the service only supports the distribution of one On-Off command at a time (i.e. $N=1$).

Address:

This gives the hardware address to which the On-Off command shall be routed.

7.3.3 Distributing register load commands

The request for the distribution of register load command(s) by means of a telecommand packet shall be:

Distribute Register Load Commands (2,2)

Telecommand packet, application data:

N	Register Address	Register Data
Unsigned Integer	Enumerated	Any
← Optional →	← Repeated N times →	

N:

The number of register load commands which follow ($N > 0$).

This field shall be systematically omitted if the service only supports the distribution of one register load command at a time (i.e. $N=1$).

Register Address:

This gives the hardware address of the register.

Register Data:

The register data consist of a set of parameters whose structure is known implicitly from the foregoing register address and which conforms to the structure rule set #1 (see clause 23).

7.3.4 Distributing CPDU commands

The request for the generation of Command Pulses on the output lines of a CPDU shall be:

Distribute CPDU Commands (2,3)

CPDU telecommand packet, application data:

Output Line ID	Duration		Output Line ID	Duration
Enumerated (1 octet)	Unsigned Integer (1 octet)		Enumerated (1 octet)	Unsigned Integer (1 octet)
← 1st Command Pulse Instruction →			← Nth Command Pulse Instruction →	

Output Line ID:

This identifies the CPDU output line on which the command pulse is issued. The values it takes are mission specific.

Duration:

A value between 0 and 8 which determines the duration of the command pulse as follows:

Command pulse duration = <CPDU_DURATION_UNIT> * 2^{Duration}

where <CPDU_DURATION_UNIT> is defined for the CPDU (between 10 ms and 15 ms).

The number of command pulse instructions in the CPDU telecommand packet is variable.

On reception of this request, the CPDU shall perform the processing outlined in 7.2.2 and specified in Annex D.

7.4 Capability sets

7.4.1 Minimum capability set

The minimum capability set shall consist of either:

- Distribute CPDU Commands (2,3) (since this is supported by a dedicated application process); or
- one of the service requests given in Table 5.

Table 5 — Summary of device command distribution service minimum capabilities

Subtype	Service request, report or capability
1	Distribute On-Off Commands
2	Distribute Register Load Commands

7.4.2 Additional capability sets

A device command distribution service that does not implement subtype 3 “Distribute CPDU Commands” can also implement one of the additional service requests given in Table 6.

Table 6 — Summary of device command distribution service additional capabilities

Subtype	Service request, report or capability
1	Distribute On-Off Commands
2	Distribute Register Load Commands

8 Housekeeping and diagnostic data reporting service

8.1 Scope

This service, along with the parameter statistics reporting and event reporting services, provides for the reporting to the ground of all information of operational significance that is not explicitly provided within the reports of another service.

The service consists of two independent sub-services which cover, respectively, the requirements for:

- a) housekeeping data reporting (both periodic and filtered in nature);
- b) diagnostic data reporting.

This service satisfies the concepts described in 4.5.1 and 4.14 of this document.

Any number of on-board application processes may provide a single instance of the housekeeping and diagnostic data reporting service.

Telemetry source packets and telecommand packets relating to the housekeeping and diagnostic data reporting service shall be denoted by Service Type = 3.

8.2 Service concept

8.2.1 Housekeeping service concept

8.2.1.1 General

The housekeeping data reporting sub-service shall sample sets of housekeeping parameters in accordance with a set of reporting definitions stored on-board. There is a predefined set of such definitions on-board (designed by the satellite manufacturer) for the housekeeping monitoring of the mission. However, these definitions may be modified or deleted and new definitions may be added, modified or deleted by the ground system, at any time.

A structure identification (SID) is associated with each distinct reporting definition and associated housekeeping parameter report. The SID shall be used on the ground, together with the APID and knowledge of the nature of the packet (i.e. that it is a housekeeping packet, as opposed to a diagnostic packet) to identify the housekeeping parameters report and to interpret its content. The SID shall be unique to a given service implementation and packet nature (i.e. housekeeping or diagnostic), however different instances of the service within different application processes can use the same values of SID.

8.2.1.2 Data collection

Each reporting definition has an associated data collection interval, which is the time interval over which the housekeeping parameters are sampled.

Parameters within a reporting definition can be sampled once per collection interval or more than once. Parameters sampled more than once are sampled regularly in time at a rate which shall be determined on-board from the combination of the data collection interval and the number of samples of the parameter in the report.

8.2.1.3 Parameter report generation

Two modes of generating housekeeping parameter reports shall exist:

- a) Periodic mode, where a housekeeping parameter report is generated once for each collection interval.
- b) "Filtered" mode, where a housekeeping parameter report is only generated at the end of the collection interval when the value of one or more parameters in the report has changed by more than a given threshold since the previous collection interval. A subset of parameters to be filtered in this manner can be selected; this subset and the thresholds to be used are specified by the ground system. In the case of parameters sampled once per collection interval, the value from the current collection interval is compared with the value from the previous collection interval. In the case of parameters sampled more than once, the change of value of one or more samples of the parameter since the last collection interval constitutes an event for this purpose.

In addition, the data sampling, data collection and parameter-report-generation activities for a housekeeping parameter report may be temporarily disabled (e.g. to reduce the on-board processing load).

The requests for selecting the mode of housekeeping parameter report generation shall be part of this service, but the requests for disabling and enabling the forwarding of housekeeping parameter reports to the ground system shall be part of the generic packet forwarding control service (see Clause 17).

Filtered mode is expected to be useful primarily for status-type parameters, where a report is only transmitted to the ground system when a significant change of status has occurred. However, in principle, any parameter within the reporting definition can be used to trigger a filtered parameter report. Those parameters that are not used for this purpose shall be “filtered out” from the comparison process whilst all unfiltered parameters shall be used. To limit the circumstances under which the parameter report is generated, a threshold for the change shall be defined, on a parameter basis, which is expressed either as a percentage or an (absolute) delta change in parameter value.

Finally, for filtered mode, the concept of parameter-report time-out is introduced whereby if no change has been detected, and hence no report generated, for a pre-specified time, a one-shot report shall be generated. This measure ensures that extensive periods of time do not elapse without the occurrence of a given (enabled) housekeeping parameter report.

8.2.1.4 Parameter sampling times

Absolute (as well as relative) sampling times of parameters in housekeeping parameter reports shall be determinable to a given accuracy (<PARAM_ABS_SAMPL_TIME> and <PARAM_REL_SAMPL_TIME>) on the ground. Three alternative methods can be used to satisfy these requirements:

- a) The parameter sampling times (e.g. time offsets with respect to packet time) can be deduced from knowledge of the on-board parameter sampling mechanism.
- b) Telemetry parameters can be explicitly time-stamped in the telemetry. For example, an on-board time register can be provided, which is sampled and telemetered along with each sample of the parameter.

Alternatively, the application process responsible for sampling can measure the sampling time and insert this time in the telemetry. However, both of these solutions are costly in terms of packet overhead.

- c) The sampling time offsets (with respect to packet time) can be measured (or otherwise evaluated) on-board over a given collection interval and reported to the ground system. The sampling “pattern” and hence the time offsets are then assumed to hold true for all housekeeping parameter reports.

For case c), explicit reporting (on request) of the measured time offsets shall be supported.

8.2.2 Diagnostic service concept

The diagnostic data reporting sub-service shall be functionally identical to the housekeeping data reporting sub-service. Different service subtypes shall be used, however, primarily for the purposes of distinguishing the diagnostic parameter reports for routing and (ground) processing.

A means to disable the generation of certain diagnostic parameter reports (whose definitions can remain on-board for intermittent use, for example, when a particular anomaly occurs) shall be provided. Because of the nature of diagnostic mode, it is anticipated that the parameter reports contain a predominance of fixed-length arrays corresponding to parameters sampled at very high rates, many times per report.

8.3 Service requests and reports

8.3.1 Defining new housekeeping or diagnostic parameter reports

The requests which provide the definition of a new housekeeping or diagnostic parameter report shall be:

Define New Housekeeping Parameter Report (3,1)

Telecommand packet, application data:

Define New Diagnostic Parameter Report (3,2)

Telecommand packet, application data:

SID	Collection Interval	NPAR1	Parameter#
Enumerated	Unsigned Integer	Unsigned Integer	Enumerated
← Optional →		← Repeated NPAR1 times →	

NFA	NREP	NPAR2	Parameter#
Unsigned Integer	Unsigned Integer	Unsigned Integer	Enumerated
← Repeated NFA times →		← Repeated NPAR2 times →	

SID:

The structure identification which corresponds to this reporting definition.

Collection Interval:

The data collection interval for this housekeeping or diagnostic parameter report definition, expressed in units of <DIAG_MIN_INTERV>.

This field shall be systematically omitted if the service uses a default value for the collection interval.

NPAR1:

The number of parameters in the definition that are sampled once per collection interval.

Parameter#:

The parameter number to be sampled. A “parameter number” is used on-board, for optimization purposes. It has a unique correspondence with the “Parameter ID” which is used on the ground for identification purposes.

NFA:

The number of fixed-length arrays (see 23.6.2 in this definition).

NREP:

The number of values to be sampled for each parameter within this fixed-length array.

NPAR2:

The number of different parameters within this fixed-length array, each of which shall be sampled “NREP” times per collection interval.

When this request is received, the new housekeeping or diagnostic parameter report definition is recorded, a corresponding “Report Generation Flag” is created and this flag is set to “Disabled”. The generation mode for this parameter report shall also be set to “Periodic” by default.

8.3.2 Clearing housekeeping or diagnostic parameter report definitions

The requests to clear one or more housekeeping or diagnostic parameter report definitions shall be:

Clear Housekeeping Parameter Report Definitions (3,3)

Telecommand packet, application data:

Clear Diagnostic Parameter Report Definitions (3,4)

Telecommand packet, application data: