The Consultative Committee for Space Data Systems

Research and Development for
Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE—RETURN CFDP PDU SERVICE

EXPERIMENTAL SPECIFICATION

CCSDS 922.27-O-1

ORANGE BOOK

May 2023

The Consultative Committee for Space Data Systems

Research and Development for Space Data System Standards

CROSS SUPPORT TRANSFER SERVICE—RETURN CFDP PDU SERVICE

EXPERIMENTAL SPECIFICATION

CCSDS 922.27-O-1

ORANGE BOOK

May 2023

AUTHORITY

Issue:	Orange Book, Issue 1
Date:	May 2023
Location:	Washington, DC, USA

This document has been approved for publication by the Consultative Committee for Space Data Systems (CCSDS). The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4).

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: secretariat@mailman.ccsds.org

FOREWORD

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Experimental Specification is therefore subject to CCSDS document management and change control procedures, which are defined in the *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the email address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Science Policy Office (BELSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- China Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Hellenic Space Agency (HSA)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Netherlands Space Office (NSO)/The Netherlands.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

PREFACE

This document is a CCSDS Experimental Specification. Its Experimental status indicates that it is part of a research or development effort based on prospective requirements, and as such it is not considered a Standards Track document. Experimental Specifications are intended to demonstrate technical feasibility in anticipation of a 'hard' requirement that has not yet emerged. Experimental work may be rapidly transferred onto the Standards Track should a hard requirement emerge in the future.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 922.27-O-1	Cross Support Transfer Service—Return CFDP PDU Service, Experimental Specification, Issue 1	May 2023	Original issue

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

Table Page

1 INTRODUCTION

1.1 SCOPE AND PURPOSE

1.1.1 GENERAL

This Experimental Specification defines the Return CCSDS File Delivery Protocol (CFDP) Protocol Data Unit (PDU) Cross Support Transfer Service (RCFDP-CSTS). The RCFDP-CSTS is an experimental CSTS that allows the reception of CFDP PDUs, in either full or reduced mode, from an RCFDP-CSTS Provider. The RCFDP-CSTS is constructed using procedures and one or more operations defined in the *Cross Support Transfer Service Specification Framework* (reference [1]).

1.1.2 SCOPE OF THE RCFDP-CSTS

The RCFDP-CSTS transfers CFDP PDUs in full or reduced operation mode. In full operation mode, all CFDP PDUs are transferred as defined in reference [3].

In the reduced operation mode, CFDP File Data PDUs are reduced before transfer. Reduced CFDP File Data PDUs are generated from original File Data PDU by replacing the original file data with information about the original length of the file data and, if the modular file checksum is used, a partial checksum.

Reduced CFDP File Data PDUs are defined within this standard.

1.1.3 SCOPE OF THIS EXPERIMENTAL SPECIFICATION

This Experimental Specification defines the RCFDP-CSTS in terms of:

a) the CSTS Framework procedures (reference [1]) that make up the service;

b) the extensions and refinements of the behavior of those CSTS procedures necessary to provide the transfer service;

c) the extensions and refinements of standard CSTS operations associated with each of the procedures;

d) the relationships among the procedures that make up the service;

e) the requirements on service production to enable the proper operation of the RCFDP-CSTS.

This Experimental Specification does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to produce or process CFDP PDUs;

d) the methods or technologies required for communication;

e) the management activities necessary to schedule, configure, and control the RCFDP-CSTS.

1.2 APPLICABILITY

The applicability and limits of applicability of Cross Support Transfer Services in general, as described in reference [1], pertain to the RCFDP-CSTS.

The RCFDP-CSTS is applicable in situations in which CFDP PDUs or reduced CFDP PDUs, as described in this Experimental Specification, need to be exchanged. Moreover, it is applicable for CFDP class 2 scenarios if the available terrestrial bandwidth does not match, the space link and file transactions may span over several ESLTs.

1.3 RATIONALE

The goal of this Experimental Specification is to create an interoperability standard for the exchange of CFDP PDUs between Cross Support Service Elements (CSSEs), as defined in reference [6], of various space agencies and Earth User Nodes (reference [6]).

The need for such a service arises in situations in which the data rates on the space link may exceed the data rate on the terrestrial link between the Earth-Space Link Terminals (ESLTs) and the Earth User Node, but the mission requires a fast closure of the protocol loop via the Earth User Node. In such cases, the RCFDP-CSTS shall be able to support

a) reconstruction of complete files at an ESLT with sending CFDP PDUs for uplink via the Earth User Node (e.g., via a central mission control system); and

b) centralized reconstruction of complete files with parts of the same file being received at different ESLTs and all CFDP PDUs for uplink being sent via the Earth User Node (e.g., via a central mission control system).

In particular, the RCFDP-CSTS shall support an architecture allowing distribution of an on-ground CFDP entity operating in Class 2, as specified by reference [3], over

- one or more ESLTs receiving CFDP PDUs;
- an Earth User Node hosting a mission control system closing the CFDP protocol loop to the spacecraft in real time;
- a centralized file reconstruction element reconstructing files by means of received CFDP PDUs in non-real time.

1.4 DOCUMENT STRUCTURE

1.4.1 DOCUMENT ORGANIZATION

Section 2 describes the RCFDP-CSTS in terms of the

- service summary of the RCFDP-CSTS;
- functional description of the RCFDP-CSTS;
- service management of the RCFDP-CSTS;
- operational scenarios that illustrate some applications of the service; and
- cross-support view of the RCFDP-CSTS.

Section 3 specifies the top-level composition of the RCFDP-CSTS. The service type identifier is declared, the procedures that make up the service are identified, and the CSTS state machine that applies to the RCFDP-CSTS is specified.

Section 4 defines the Return CFDP PDU Delivery procedure as a derivation of the Buffered Data Delivery procedure as specified in 4.5 of reference [1].

Section 5 specifies how the procedure configuration parameters are to be set for the RCFDP-CSTS.

Section 6 specifies the Return CFDP PDU–specific versions of the service-generic parameters and events that are defined in reference [1].

Section 7 defines the refinement of parameters and events defined in reference [1] as they apply to the RCFDP-CSTS.

Annex A documents the Implementation Conformance Statement (ICS) Proforma for the RCFDP-CSTS.

Annex B formally specifies the Object Identifiers (OIDs) for the RCFDP-CSTS service, the RCFDP-CSTS Provider Functional Resource Types, and the RCFDP Recording Buffer Functional Resource Type.

Annex C formally specifies the ASN.1 PDUs for the Return CFDP PDU Delivery procedure.

Annex D formally specifies the ASN.1 parameters, events, and directives for the Return CFDP PDU Delivery procedure.

Annex E defines the Return CFDP service production process.

Annex F addresses the security, Space Assigned Numbers Authority (SANA), and patent considerations associated with the RCFDP-CSTS.

Annex G lists the abbreviations and acronyms used in this Experimental Specification.

Annex H lists the informative references cited in this Experimental Specification.

1.4.2 CROSS SUPPORT TRANSFER SERVICES DOCUMENTATION

The basic organization of the CSTS documentation and the relationship to CSTS documentation are shown in figure 1-1.

The Cross Support Services documents related to Cross Support Transfer Services are:

a) *Cross Support Concept—Part 1: Space Link Extension Services* (reference [H1]), a report introducing the concepts of cross support and the SLE services. Many of the concepts for the SLE Transfer Services have been adopted for the CSTSes (refer to c) below);

b) *Cross Support Reference Model—Part 1: Space Link Extension Services* (reference [2]), a Recommended Standard that defines the framework and terminology for the specification of SLE services. Much of the framework and terminology of this reference model has been adopted or adapted for CSTSes in reference [1];

c) *Space Communication Cross Support Service Management* suite (references [H6], [H7], and [H8]), Recommended Standards that specify the Service Management Information Entities that are used to configure and schedule CSTSes;

d) The SLE Transfer Services suite, Recommended Standards that specify cross support transfer services used to transfer specific telecommand and telemetry protocol data units. The SLE Transfer Services are closely related to the CSTS suite in that they collectively define the set of operations that are the basis for reference [1]. However, for historical reasons (the SLE Transfer Services were already specified and implemented prior to development of reference [1]), the SLE Transfer Services are separated from CSTSes;

e) *Space Link Extension—Internet Protocol for Transfer Services* (reference [H2]), a Recommended Standard that defines a protocol for transfer of PDUs defined in the Cross Support Transfer Services. This Recommended Standard was originally developed to support SLE Transfer Services (hence the title), but it is also applicable to (and specified for) use by Cross Support Transfer Services.

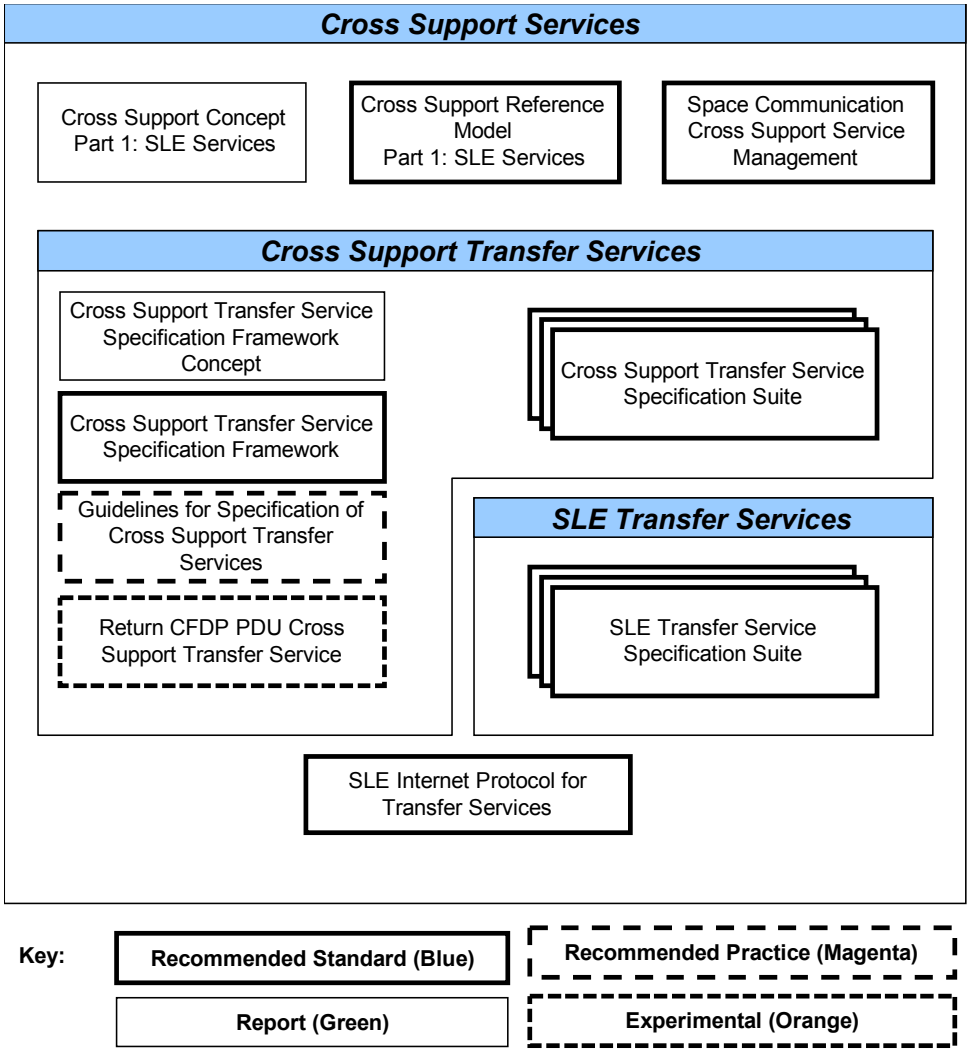

Figure 1-1: Cross Support Services Documentation

The documents specific to Cross Support Transfer Services are:

a) *Cross Support Transfer Services Specification Framework* (reference [1]), a Recommended Standard that specifies the Cross Support Transfer Services procedures and defines their basic building blocks;

b) *Guidelines for the Specification of Cross-Support Transfer Services*, a Recommended Practice that defines the guidelines for construction of a Cross Support Transfer Service based on reference [1];

c) *Cross Support Transfer Services Specification Framework Concepts* (reference [H3]), a report that provides tutorial material on the objectives and concepts of reference [1];

d) Cross Support Transfer Services suite, the set of specifications for actual CSTSes built from the procedures in reference [1] and in accordance with the CSTS Guidelines. The Cross Support Transfer Service Suite includes this Experimental Specification.

1.5 DEFINITIONS, NOMENCLATURE, AND CONVENTIONS

1.5.1 TERMS

1.5.1.1 Terms Defined in the CSTS Specification Framework

This Experimental Specification makes use of the following terms defined in reference [1]:

a) Association Control procedure;
b) blocking [operation];
c) Buffered Data Delivery procedure;
d) complete [data delivery mode];
e) Cross Support Complex;
f) Cross Support service production;
g) Cross Support Transfer Service;
h) Cross Support Transfer Service provision;
i) delivery mode;
j) discardable;
k) latency limit;
l) non-blocking [operation];
m) non-discardable;
n) procedure configuration parameter;
o) qualified parameter;
p) real-time [data delivery mode];
q) recording buffer;
r) return-buffer-size;
s) service instance provision period;
t) service management parameter;
u) service production data unit;
v) service-user-responding-timer;
w) start-generation-time;
x) stop-generation-time;
y) TransferDataInvocation.

1.5.1.2 Terms Defined in the Cross Support Reference Model

This Experimental Specification makes use of the following terms defined in reference [2]:

a) Complex Management, CM;

b) Forward CLTU, F-CLTU;

c) Mission Data Operations System, MDOS;

d) Return All Frames, RAF;

e) Service Package;

f) Space Link Session;

g) Utilization Management, UM.

1.5.1.3 Other Terms Defined in the CFDP Specification

This Experimental Specification makes use of the following terms defined in reference [3]:

a) CFDP Protocol Data Unit or CFDP PDU;

b) File Data PDU;

c) File Directive PDU;

d) Transaction;

e) Source;

f) Destination;

g) Sender or sending entity;

h) Receiver or receiving entity;

i) Filestore;

j) Offset.

1.5.1.4 Definitions from Abstract Syntax Notation One

This Experimental Specification makes use of the following terms defined in reference [4]:

a) Abstract Syntax Notation One, ASN.1;

b) Object Identifier, OID.

NOTE – OIDs belong to an ISO/IEC-standardized identifier mechanism for naming any object, concept, or 'thing' with a globally unambiguous persistent name. An OID represents a node in a hierarchically assigned name space defined by reference [4]. OIDs used in the CCSDS context are documented in dedicated registries (reference [5]).

1.5.1.5 Terms Defined in CCSDS Recommended Practice for Space Communication—Cross Support Architecture Requirements Document

This Experimental Specification makes use of the following terms defined in reference [6]:

a) Cross Support Service Element, CSSE;

b) Cross Support Service System, CSSS;

c) Earth-Space Link Terminal, ESLT;

d) Earth User Node.

1.5.1.6 Terms Defined in this Specification

CFDP Protocol State Machine: A subset of a CFDP entity as defined in reference [3]. It implements all procedures and state machines underlying the protocol specification as defined in [3]. This involves processing the received CFDP PDUs and sending appropriate CFDP PDUs (e.g., ACK, NAK). A CFDP Protocol State Machine is able to process reduced CFDP File Data PDUs as defined in this Experimental Specification.

CFDP File Reconstruction: Assembly of file parts transfered as PDUs by means of CFDP into a copy of the complete file transmitted by the Sender. CFDP File Reconstruction is accomplished by processing CFDP PDUs as specified by reference [3]. However, a CFDP File Reconstruction does not implement the CFDP state machines and does not send any CFDP PDUs, which is assumed to be done by a CFDP Protocol State Machine as defined above. In fact, the combination of a *CFDP Protocol State Machine* and a *CFDP File Reconstruction* constitute a *CFDP entity* as specified by reference [3].

Reduced CFDP File Data PDU: A PDU constructed from an original CFDP File Data PDU by preserving the header and replacing the original file data. Reduced CFDP File Data conform to the File Data PDU Format defined in reference [3]. The original file data is replaced by information about the original length of the file data segment and, if the modular file checksum is used, a partial checksum. The PDU Data field length in the header is adjusted in order to obtain a valid CFDP PDU. The processing of reduced CFDP File Data PDUs requires a CFDP Protocol State Machine, which can process these Reduced CFDP File Data PDUs.

Return CFDP PDU Cross Support Transfer Service, RCFDP-CSTS: The Return CFDP PDU Cross Support Transfer Service as defined by this book.

1.5.2 NOMENCLATURE

1.5.2.1 Normative Text

The following conventions apply for the normative specifications in this Experimental Specification:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.5.2.2 Informative Text

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

– Overview;

– Background;

– Rationale;

– Discussion.

1.5.3 CONVENTIONS

This Experimental Specification uses the conventions defined in reference [1].

1.6 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications. However, this issue of this Experimental Specification is valid only in conjunction with the specifications provided in the issue identified below in [1].

[1] *Cross Support Transfer Service—Specification Framework*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 921.1-B-2. Washington, D.C.: CCSDS, February 2021.

[2] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 910.4-B-2. Washington, D.C.: CCSDS, October 2005.

[3] *CCSDS File Delivery Protocol*. Issue 5. Recommendation for Space Data System Standards (Blue Book), CCSDS 727.0-B-5. Washington, D.C.: CCSDS, July 2020.

[4] *Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. 5th ed. International Standard, ISO/IEC 8825-1:2015. Geneva: ISO, 2015.

[5] "Functional Resources." Space Assigned Numbers Authority. https://sanaregistry.org/r/functional_resources/.

[6] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

2 OVERVIEW OF THE RETURN CFDP SERVICE

2.1 SERVICE SUMMARY

The Return CFDP PDU CSTS is an experimental CCSDS Cross Support Transfer Service that provides a service user with the means to receive CFDP PDUs formatted in accordance with reference [3]. The CFDP PDUs are produced during the utilization phase of the cross support Service Package.

2.2 FUNCTIONAL DESCRIPTION

2.2.1 SERVICE PRODUCTION

The production of CFDP PDUs for transfer via the RCFDP-CSTS is associated with a given Service Package that consists of:

a) the CFDP PDU Recording Buffer function that stores the CFDP PDUs for subsequent retrieval by RCFDP-CSTS instances;

b) the functions that can generate CFDP PDUs according to reference [3].

Generally speaking, the service production in the context of the Return CFDP PDU service needs to produce CFDP PDUs. This can be, for instance, a collocated CFDP entity. But service production also refers to when these CFDP PDUs are extracted from space packets and transfer frames, which have been received over the space link from an onboard CFDP entity.

2.2.2 SERVICE PROVISION

The RCFDP-CSTS provides CFDP PDUs by the means of its primary procedure, CFDP PDU Delivery Procedure, to service users.

The CFDP PDU Delivery Procedure has two operation modes:

– full;

– reduced.

In ‘full’ mode, the CFDP PDUs are forwarded as originally produced by a CFDP entity. In ‘reduced’ mode, the CFDP File Data PDUs are reduced in terms of file data, as defined in annex E. The ‘reduced’ mode primarily targets the distributed CFDP scenario as described in 2.4.3. The ‘full’ mode of the RCFDP-CSTS is required for both operational scenarios outlined in 2.4 and potentially others. (Refer to figure 2-2 for an illustration.)

In addition, forwarding can be restricted by the CFDP destination entity ID. The CFDP destination entity ID is part of the CFDP PDU header and can be used by the RCFDP service. Only CFDP PDUs matching a configured CFDP destination entity ID are forwarded to the RCFDP user. Filtering of this sort allows for the use of different operation modes for different

destinations. Examples include, but are not limited to, different treatment of housekeeping files sent to a control center and science files transferred to the payload data processing.

For delivery of CFDP PDUs, the RCFDP-CSTS service has two delivery modes:

– complete;

– real time.

In the complete data delivery mode, the RCFDP-CSTS is guaranteed to deliver all CFDP PDUs, but without a guaranteed limit on the latency of their delivery. This is expected to be the primary mode used for operations, assuming that the underlying communication network provides sufficient throughput in the given operational context and the selected operational mode. If sufficient throughput cannot be guaranteed, delayed delivery of CFDP PDUs might lead to (unnecessary) retransmission of CFDP PDUs (CFDP Class 2 only) and eventual CFDP faults with incomplete files, depending on the CFDP MIB configurations.

In the real-time data delivery mode, the CFDP PDUs are guaranteed to be delivered within a worst-case latency that is defined by the service user, respecting the minimum latency supported by the given service provider. In order to stay within this worst-case latency, a real-time RCFDP-CSTS instance can discard some CFDP PDUs. This can happen if backpressure in the connection between the service user and service provider makes delivery of all CFDP PDUs within the specified latency impossible (e.g., if the network that connects the service user and provider becomes congested, or the RCFDP user cannot operate fast enough). The consequences of such lost CFDP PDUs have to be evaluated in the context of the operational scenario. This may lead to retransmission of CFDP PDUs (CFDP Class 2 only) and eventual CFDP transaction faults with incomplete files depending on CFDP MIB configurations.

When the system operates in real-time delivery mode, the likelihood of CFDP retransmissions and faults resulting from CFDP PDUs lost due to backpressure has to be weight against the potential late delivery of CFDP PDUs in complete mode. Late delivery can occur if the underlying communication network provides a lower throughput than needed to transfer all CFDP PDUs in real time. Although the real-time mode is not expected to be the primary operational choice, it may be used to circumvent varying throughput characteristics of terrestrial communication networks.

2.3 SERVICE MANAGEMENT

Cross Support Service Management establishes the constraints on the Service Packages, to which a given spaceflight mission conforms (e.g., data rate and frequency ranges, types and numbers of Cross Support Transfer Service instances), and provides the mechanisms for instantiating conformant Service Packages (e.g., via scheduling).

With regard to the provision of RCFDP-CSTS instances, Cross Support Service Management:

a) schedules the Service Packages that specify the RCFDP-CSTS instances that transfer the CFDP PDUs to the users of those service instances; and

b) establishes the configuration of each RCFDP-CSTS instance during the execution of a Service Package.

In the context of a Service Package providing RCFDP-CSTS instances, the Service Package identifies the various space communication and RCFDP-CSTS functions that are to be performed by a Cross Support Complex during a specified period of time. Typically, the Service Package corresponds to the functions performed at a single ground station for one pass/contact/track.

The Service Package also defines, indirectly through reference to configuration profiles, the configuration parameters that specify the initial configurations of the space communication- and RCFDP-CSTS-related functions and the interrelationships among them (e.g., the frame length on each return link symbol stream and RCFDP-CSTS configuration; for instance, CFDP entities to forward CFDP PDUs). With respect to the Return RCFDP-CSTS Service, the Service Package identifies which resources perform the functions that produce the CFDP PDUs that are delivered by the Return CFDP PDU service.

The means by which service management schedules the Service Packages is outside the scope of this Recommended Standard. The CCSDS SCCS-SM suite (references [H7] and [H8]) defines a standard set of service management information entities used in the scheduling of Service Packages.

2.4 OPERATIONAL SCENARIO

2.4.1 OVERVIEW

This subsection presents some operational scenarios making use of the RCFDP-CSTS. These scenarios are by no means prescriptive to the use of the RCFDP-CSTS. All presented scenarios imply the use of CFDP Class 2, which ensures retransmission of missing file-segments that are eventually lost in transfer.

Moreover, both presented operational scenarios are motivated by the potential limitation that the throughput of the communication network from the ESLT may have a lower throughput than the space link itself.

2.4.2 SINGLE ESLT CFDP SCENARIO

The single ESLT CFDP scenario, as shown in figure 2-1, illustrates a case of a spacecraft transferring onboard files to a particular ground station by means of a CFDP protocol entity running Class 2 reliable service in the ESLT. As a boundary condition, this scenario can only

transfer complete files to a single ESLT. Files cannot be partially downlinked to one ground station and partially to another ground station.

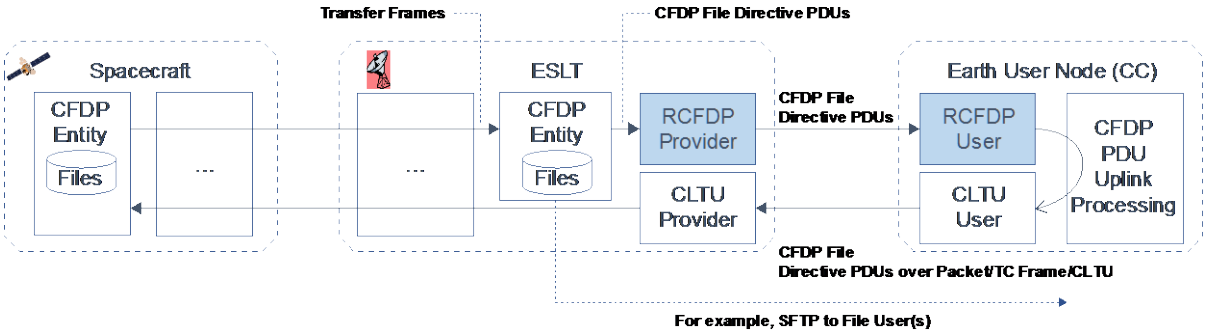

Figure 2-1: Single ESLT CFDP Scenario

However, this scenario can cope with an ESLT being connected to the Earth User Node at a lower bandwidth than the space link. The CFDP protocol, file processing, and Protocol State Machine are executed in real time between spacecraft and ESLT, but all generated CFDP File Directive PDUs (NAK, ACK, Finished) to be uplinked to the spacecraft are transferred by means of the RCFDP-CSTS from the ESLT to the Earth User Node. The Earth User Node controls delivery of the CFDP File Directives back to the Spacecraft via the normal uplink stream. After inserting CFDP PDUs into packets and TC frames and encoding them as a CLTU, it uses the CCSDS SLE CLTU service to uplink the CFDP PDUs to the spacecraft. Completed files are produced and stored in the ESLT and can be retrieved by standard terrestrial file transfer protocols from the ESLT to the desired destination.

The SLE CLTU protocol has been selected for demonstration purposes because it is a widely adopted service for transferring commands to an ESLT. However, the Forward Frame CSTS would also be a perfectly suitable alternative to CLTU in this scenario.

Finally, the files reconstructed on ground in the ESLT can be retrieved or pushed to users by, for example, the SFTP protocol or other available terrestrial file transfer protocols.

2.4.3 DISTRIBUTED CFDP SCENARIO

The distributed CFDP scenario takes the single ESLT CFDP scenario presented in 2.4.2 one step further. It provides the capability to transfer large files reliably, partially downlinked to different ground stations, to a central Earth User Node, where the onboard files are reconstructed by further processing of all of the CFDP PDUs.

Especially for low-Earth-orbiting spacecraft facing a combination of relatively short passes and fast downlink rates combined with less reliable space links (e.g., K or Ka Band), the distributed CFDP scenario as shown in figure 2-2 can be attractive. It is a key feature of this Experimental Specification that file transaction can span multiple passes over different ESLTs in the presence of terrestrial bandwidth constraints.

Conceptually, a CFDP entity as specified in reference [3] is distributed among the CFDP Protocol State Machine running in a 'Control' EUN that is able to handle reduced CFDP PDUs, and the CFDP File Reconstruction then accepts CFDP PDUs from all ESLTs and delivers the resulting complete file to the Users. Both of these entities are shown in grey in figure 2-2. It is important to stress that from a spacecraft perspective, the behavior is fully compliant with a complete CFDP entity as specified by reference [3].

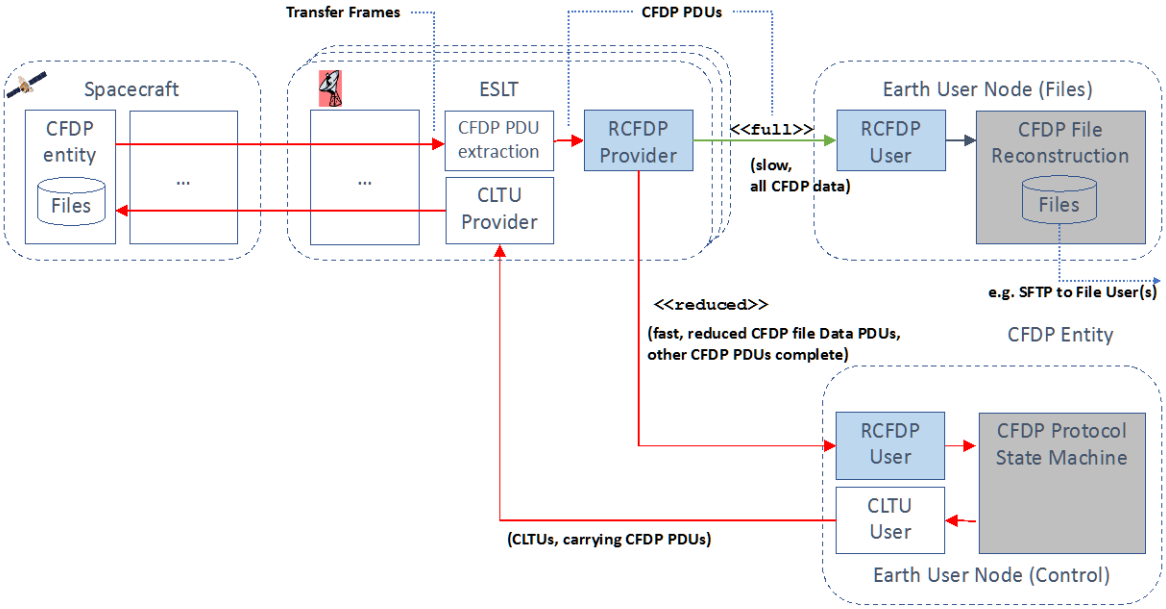

Figure 2-2: Distributed CFDP Scenario

While downlinking files, CFDP PDUs, usually included in packets and transfer frames, are sent from the spacecraft to the ESLT over the available communication channel, which may lose CFDP PDUs. In each ESLT, the signal from the spacecraft is received, demodulated, decoded, and processed. CFDP PDUs, extracted from packets and transfer frames, are then forwarded to the RCFDP-CSTS Provider, which provides the means to buffer the incoming CFDP PDUs and forward them. Forwarding of CFDP PDUs will happen according to the CFDP destination entity ID, which is also part of the RCFDP Service Instance Configuration. (See 4.6 for the configuration parameters of the CFDP PDU Delivery Procedure.)

Individual RCFDP-CSTS Service Instances, each operating either in 'full' or 'reduced' mode, implement the forwarding. To close the CFDP Class 2 protocol loop in real time, an RCFDP-CSTS Service Instance operating in 'reduced' mode forwards all CFDP File Directive PDUs as well as the reduced CFDP File Data PDUs to the Earth User Node in charge of controlling the spacecraft. More precisely, the reduced CFDP PDUs are forwarded to an instance of a CFDP Protocol State Machine, implementing the CFDP protocol behavior as defined in reference [3]. The CFDP Protocol State Machine must be able to process reduced CFDP File Data PDUs. The CFDP Protocol State Machine generates CFDP NAKs for missing file data, as well as CFDP ACK (EOF) and CFDP Finished PDUs according to the CFDP protocol as defined in reference [3], just as would be done for the original File Data PDU. The CFDP PDUs generated by the CFDP Protocol State Machine are

appropriately encapsulated in packets and TC frames, encoded as a CLTU and sent via the SLE CLTU service to the ESLT. The ESLT uplinks the CLTUs carrying CFDP PDUs to the spacecraft, where CFDP PDUs are finally received and processed by the onboard CFDP entity. Such a CFDP Class 2 real-time loop is indicated in figure 2-2 by the red arrows.

To reconstruct the files on the ground, the RCFDP-CSTS Service Instances in ESLTs operating in 'full' mode buffer and forward the CFDP PDUs to a central Earth User Node hosting a CFDP File Reconstruction instance. File reconstruction happens by reconstructing the original files from the received CFDP PDUs, which should for each file constitute the complete CFDP File Data PDUs necessary to reconstruct the file. Missing CFDP File Data PDUs have been requested to be eventually retransmitted by CFDP NAKs from the CFDP Protocol State Machine in real time. It must be noted that this transfer of CFDP PDUs to the CFDP File Reconstruction can happen at a lower data rate than the data rate of the space link. Moreover, the CFDP File Reconstruction can receive CFDP PDUs from several ground stations; this allows the downlink of larger files over several passes and ground stations. The capability of the RCFDP-CSTS to replay recorded CFDP PDUs may allow reprocessing in cases in which that is operationally required. Such a replay is of course limited by the available storage at ESLTs.

Finally, the files reconstructed on ground by the CFDP File Reconstruction can be retrieved or pushed to users by, for example, the SFTP protocol or other available terrestrial file transfer protocols.

3 RETURN CFDP PDU SERVICE COMPOSITION

3.1 DISCUSSION

The service-level OIDs for the CFDP Return PDU CSTS are specified in annex B.

The CFDP Return PDU CSTS specification is complete and therefore concrete; it can be implemented as defined herein without need for further extension or refinement.

3.2 PROCEDURES OF THE RETURN CFDP PDU CROSS SUPPORT TRANSFER SERVICE

3.2.1 The CFDP Return PDU CSTS shall be composed of the Association Control and CFDP PDU Delivery procedures.

3.2.2 The Association Control procedure shall conform to the Association Control procedure as defined in 4.3 of reference [1] without derivation.

3.2.3 There shall be one and only one instance of the Association Control procedure.

3.2.4 The CFDP PDU Delivery procedure shall be derived from the Buffered Data Delivery procedure as specified in 4.5 of reference [1]. The derivation is specified in section 4.

3.2.5 The CFDP PDU Delivery procedure shall be the primary procedure for the CFDP Return PDU Service.

3.2.6 There shall be one and only one instance of the CFDP PDU Delivery procedure.

3.2.7 The version number of the CFDP PDU Delivery procedure is 1.

NOTE – Table 3-1 summarizes the procedures that make up the Return CFDP PDU transfer service, where (a) the '[P]' designates the CFDP PDU Delivery Procedure as the primary procedure; (b) *Version* = '-' indicates that the version of the service procedure is the same as that of the procedure specified in reference [1] for the procedures that are directly adopted (Association Control and Information Query), and *Version* = '1' indicates the version of the refined and/or extended service procedures (CFDP PDU Delivery Procedure); (c) *No. of Instances* indicates the minimum and maximum number of allowed instances of each procedure type; (d) *Specification Approach* indicates which procedures are directly adopted or refined and extended; and (e) *Source* indicates the procedure specified in reference [1] from which the service procedure is adopted, refined and/or extended.

Table 3-1: Procedures of the Return CFDP PDU CSTS

Procedure	Association Control	CFDP PDU Delivery procedure [P]
Version	-	1
No. of Instances	1..1	1..1
Specification Approach	adopted	Extended
Source	Subsection 4.3 of reference [1]: Association Control	Subsection 4.5 of reference [1]: Buffered Data Delivery

3.3 RETURN CFDP PDU SERVICE STATE MACHINE

The Return CFDP PDU Cross Support Transfer Service state machine conforms to the state machine for a CSTS with a stateful prime procedure instance, as defined in G3 of reference [1].

4 CFDP PDU DELIVERY PROCEDURE

4.1 DISCUSSION

4.1.1 PURPOSE

The CFDP PDU Delivery procedure supports transfer from the Provider to the User of CFDP PDUs as defined in reference [3], structured into data units using one of the real-time or complete delivery modes, as defined in 4.5 of reference [1]. The single data units are sequences of octet strings, each containing one or more, full or reduced, PDUs.

4.1.2 CONCEPT

The CFDP PDU Delivery Procedure (CFDPPD) is derived from the Buffered Data Delivery procedure specified in reference [1].

a) The contents of the `data` parameter of the TransferDataInvocations are extended specifying the format of the `extendedData` as a sequence of octet strings to carry CFDP PDUs as defined by reference [3].

b) The number of the PDUs contained in the transferred sequence depends on the operational needs defined through configuration parameters defining the maximum size and the maximum waiting time.

A CFDPPD instance can operate in full or reduced mode, as configured by Service Management.

NOTE – The parent Buffered Data Delivery procedure in reference [1] specifies that the START operation includes selection criteria for the selection of the TransferDataInvocations to be delivered. In the case of the CFDPPD, the selection criteria applied in the START operation are limited to the `start-generation-time` and `stop-generation-time` parameters specified in the parent Buffered Data Delivery procedure of reference [1]. Selection of the type of File Data PDU to be delivered is configured in the Service Package prior to the start of the RCFDP-CSTS service instance provision period and applies for the duration of the service instance provision period.

4.2 PROCEDURE TYPE IDENTIFIER

The procedure identifier `rtnCfdpPduBuffDel`, as specified in annex B, shall be used for this procedure.

4.3 EXTENSION

The CFDP PDU Delivery shall extend the Buffered Data Delivery procedure by modification of the behavior of the procedure and the definition of the format of a parameter of the TRANSFER-DATA operation.

4.4 BEHAVIOR

4.4.1 OVERVIEW

The overall activities of the CFDP PDU Delivery procedure are the same as those of the standard Buffered Data Delivery procedure as defined in 4.5 of reference [1]. The detailed behavior of the CFDP PDU Delivery procedure with respect to starting, transferring data and notifications, stopping, and aborting is the same as that of the standard Buffered Data Delivery procedure as defined in 4.5 of reference [1].

4.4.2 DERIVED BEHAVIOR—TRANSFERRING DATA AND NOTIFICATIONS

4.4.2.1 The `cfdp-pdu-operation-mode` configuration parameter shall specify one or more of the following operation modes:

- full;
- reduced.

4.4.2.2 The `cfdp-pdu-dest-entities` configuration parameter shall specify a list of CFDP destination entity IDs.

4.4.2.3 CFDP PDUs shall only be forwarded if the destination entity ID in the CFDP PDU Header matches an entry in the `cfdp-pdu-dest-entities` configuration parameter or if the list of `cfdp-pdu-dest-entities` is empty. These CFDP PDUs will be called ‘matching CFDP PDUs’ in the following.

4.4.2.4 In full operation mode, all matching CFDP PDUs shall be transferred unchanged.

4.4.2.4.1 Reduced operation mode should only be used with the modular checksum or the null checksum, according to reference [3].

4.4.2.4.2 Other checksum types may be used if they allow an incremental calculation such as the modular checksum.

4.4.2.5 If an incremental checksum is used, file segments in different File Data PDUs shall be pairwise disjunct or identical.

NOTE – This is required to allow an incremental calculation of the file checksum. If the segmentation of a file into file segments does not change even for retransmission of lost file data, this condition is fulfilled.

4.4.2.6 In reduced operation mode, matching CFDP File Directive PDUs shall be transferred unchanged.

4.4.2.7 In reduced operation mode, matching CFDP File Data PDUs shall be changed as follows:

4.4.2.7.1 If the modular checksum is used, the file data parameter shall be replaced completely by

a) the original file segment length (2 octets); followed by

b) the partial modular checksum calculated over the file data in the original File Data PDU (4 octets).

NOTE – The same approach may be used if other checksums are used which allow an incremental computation.

4.4.2.7.2 If the null checksum is used, the file data parameter shall be replaced completely by the original file segment length (2 octets).

4.4.2.7.3 The PDU Data Field Length in the Fixed PDU Header shall be adjusted to the length of the new PDU Data field.

NOTE – All other parameters in the File Data PDU will remain unchanged.

4.5 REQUIRED OPERATIONS

4.5.1 GENERAL

4.5.1.1 The CFDP PDU Delivery procedure shall use the STOP and NOTIFY operations of the Buffered Data Delivery procedure of reference [1] without extension or refinement.

4.5.1.2 CFDP PDU Delivery procedure shall use the START operation of the Buffered Data Delivery procedure of reference [1].

4.5.1.3 The CFDP PDU Delivery procedure shall refine the TRANSFER-DATA operation of the Buffered Data Delivery procedure of reference [1] as specified in 4.5.2.

4.5.1.4 The START and STOP operations of the CFDP PDU Delivery procedure shall be Blocking, as defined for the parent Buffered Data Delivery procedure in reference [1].

4.5.1.5 The TRANSFER-DATA and NOTIFY operations of the CFDP PDU Delivery procedure shall be Non-Blocking, as defined for the parent Buffered Data Delivery procedure in reference [1].

NOTE – Table 4-1 summarizes the operations of the CFDP PDU Delivery procedure.

Table 4-1: CFDP PDU Delivery Required Operations

Operations	Extended	Refined	Procedure Blocking/Non-Blocking
START	N	N	Blocking
STOP	N	N	Blocking
TRANSFER-DATA	Y	N	Non-Blocking
NOTIFY	N	N	Non-Blocking

4.5.2 TRANSFER-DATA (UNCONFIRMED)

The **`extendedData`** field of the **`data`** parameter choice shall be used and defined as a sequence of octet strings, each containing a full or reduced CFDP PDU, formatted as defined in reference [3].

4.6 CONFIGURATION PARAMETERS

Table 4-2 defines the configuration parameters of the CFDP PDU Delivery procedure that need to be configured in the context of this procedure. For each configuration parameter, the table provides the engineering unit (if applicable), provides a cross reference to the use of the parameter in the specification of the procedure, identifies whether the parameter may be read and/or dynamically modified, and also identifies the Parameter Identifier to be used in reporting the value of the parameter.

Table 4-2: CFDP PDU Delivery Configuration Parameters

Parameters	Cross-Reference	Readable	Dynamically modifiable	Configuration Parameter Identifier and Type
`return-buffer-size` (in number of TRANSFER-DATA and/or NOTIFY invocations the buffer will accommodate)	Subsection 4.5.3.2.7.3 of reference [1]	Yes	Yes	`pBDDreturnBufferSize` `PBDDreturnBufferSizeType` (subsection F3.16 of reference [1])
`delivery-latency-limit` (in seconds)	Subsection 4.5.3.2.7.2 of reference [1]	Yes	Yes	`pBDDdeliveryLatencyLimit` `PBDDdeliveryLatencyLimitType` (subsection F3.16 of reference [1])

Parameters	Cross-Reference	Readable	Dynamically modifiable	Configuration Parameter Identifier and Type
delivery-mode	Subsections 4.5.2.2.2.1 and 4.5.3.2.6 of reference [1]	Yes	No	pBDDdeliveryMode PBDDdeliveryModeType (subsection F3.16 of reference [1])
cfdp-pdu-operation-mode		Yes	No	pCFDPPDcfdpPduOperationMode PCFDPPDcfdpPduOperationModeType (annex D)
cfdp-pdu-dest-entities		Yes	No	pCFDPPDcfdpDestEntities PCFDPPDcfdpDestEntitiesType (annex D)
cfdp-pdu-aggregate-max-size		Yes	No	pCFDPPDcfdpPduAggregationMaxSize PCFDPPDcfdpPduAggregationMaxSizeType (annex D)
cfdp-pdu-aggregate-timeout		Yes	No	pCFDPPDcfdpPduAggregationTimeout PCFDPPDcfdpPduAggregationTimeoutType (annex D)

4.7 PROCEDURE STATE TABLE (PROVIDER SIDE)

The state table for the CFDP PDU Delivery procedure shall be the same as that for the Buffered Data Delivery procedure as specified in 4.5.6 of reference [1].

5 SETTING OF SERVICE MANAGEMENT AND CONFIGURATION PARAMETERS INHERITED FROM FRAMEWORK OPERATIONS AND PROCEDURES

5.1 OVERVIEW

The BIND operation defines the `responder-port-identifier` parameter (refer to 3.4.2.2.4.3 of reference [1]) to be a service management parameter of each CSTS. Subsection 5.2, below, specifies the classifier to be used for the `responder-port-identifier` parameter for the RCFDP-CSTS. The `parameterId` corresponding to this classifier is defined in the SANA Functional Resource Registry (reference [5]) subtree for the Return CFDP PDU CSTS Provider Functional Resource.

NOTE – As described in the specification of the `responder-port-identifier` parameter in reference [1], the contents of the parameter are not used by the procedures of the CSTS provider itself, but rather by the underlying communication service that delivers the incoming PDUs to a CSTS provider. The purpose of assigning a classifier and `parameterId` to this parameter is to allow its value to be reported or queried.

The procedures of reference [1] define configuration parameters for those procedures but defer to the derived procedures for specification of the method by which each of those configuration parameters is to be set. This section specifies the method by which each of the procedure configuration parameters defined in reference [1] is to be set for the RCFDP-CSTS.

For each of the procedure configuration parameters that are specified to be a service management parameter, the classifier for each parameter is also specified. The `parameterId` corresponding to each such classifier is defined in the SANA Functional Resource Registry (reference [5]) subtree for the Return CFDP PDU CSTS Provider Functional Resource.

5.2 `responder-port-identifier` SERVICE MANAGEMENT PARAMETER

The `responder-port-identifier` service management parameter (subsection 3.4.2.2.4.3 of reference [1]) shall have the classifier `tdResponderPortId`.

5.3 ASSOCIATION CONTROL PROCEDURE CONFIGURATION PARAMETERS

5.3.1 The service-user-responding-timer (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdServiceUserRespondingTimer`.

5.3.2 The `initiator-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdInitiatorId`.

5.3.3 The `responder-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdResponderId`.

5.3.4 The `service-instance-identifier` (refer to table 4-2 in 4.3.5 of reference [1]) shall be configured by a service management parameter with the classifier `tdServiceInstanceId`.

5.4 CFDP PDU DELIVERY PROCEDURE CONFIGURATION PARAMETERS

NOTE – Reference [1] defers the setting of the `delivery-mode`, `delivery-latency-limit`, and `return-buffer-size` configuration parameters to the service that uses the Buffered Data Delivery procedure or a procedure derived from it. For the RCFDP-CSTS, this is the CFDP PDU Delivery procedure.

5.4.1 The `delivery-mode` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `cfdpPduDeliveryMode`.

5.4.2 The `cfdp-pdu-dest-entities` parameter allows definition of a filter based on CFDP Destination IDs. Only if the CFDP destination ID of a CFDP PDU matches an item in `cfdp-destination-ids` will it be sent. For an empty `cfdp-pdu-dest-entities`, all CFDP PDUs are sent.

5.4.3 The `return-buffer-size` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `rtnCfdpPduBufferSize`.

5.4.4 The `delivery-latency-limit` (refer to table 4-16 in 4.5.5 of reference [1]) shall be configured by a service management parameter with the classifier `cfdpPduDeliveryLatencyLimit`.

5.4.5 The `cfdp-pdu-aggregate-max-size` parameter shall be configured to define the maximum size of an aggregate (sequence of octets) of CFDP PDUs.

5.4.6 The `cfdp-pdu-aggregate-timeout` parameter shall be configured to define the maximum time the provider will wait to reach the maximum size, before sending an aggregate of lower size.

5.4.7 The `cfdp-pdu-operation-mode` parameter configures how the CFDP File Data PDUs are delivered. In full mode, they are delivered with the file data; in reduced mode, the file data is eliminated from the CFDP File Data PDU as defined in annex E.

6 CFDP RETURN PDU SERVICE-SPECIFIC VERSIONS OF SERVICE-GENERIC PARAMETERS AND EVENTS

6.1 OVERVIEW

Annex B of reference [1] specifies the following service-generic parameters and events for use by any CSTS:

a) A production status that can be monitored. The OID to be used for the parameter that contains the production status for every CSTS is specified in F3.17 of reference [1] with the classifier `svcProductionStatusVersion1`.

b) A production status change event that is to be emitted when the production status changes, as specified in 3.11.2.2.3.2 a) of reference [1]. The OID to be used for the production status change event for every CSTS is specified in F3.17 of reference [1] with the classifier `svcProductionStatusChangeVersion1`.

c) A production configuration change event that is to be emitted when any Functional Resource in the production experiences a configuration change, as specified in 3.11.2.2.3.2 b) of reference [1]. The OID to be used for the production configuration change event for every CSTS is specified in F3.17 of reference [1] with the classifier `svcProductionConfigurationChangeVersion1`.

Each CSTS is to provide its own label for the production status parameter, production status change event, and production configuration change event.

The Return CFDP PDU service supports the production status parameter as well as the production status change and production configuration change events.

6.2 `cfdpPduSvcProductionStatus` PARAMETER

The `cfdpPduSvcProductionStatus` parameter shall contain the production status, with the Published Identifier `svcProductionStatusVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F3.17 of reference [1].

6.3 `cfdpPduSvcProductionStatusChange` EVENT

The `cfdpPduSvcProductionStatusChange` event shall report production status changes, with the Published Identifier `svcProductionStatusChangeVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F3.17 of reference [1].

6.4 `cfdpPduSvcProductionConfigurationChange` EVENT

The `cfdpPduSvcProductionConfigurationChange` event shall report production configuration changes, with the Published Identifier `svcProductionConfigurationChangeVersion1` as specified in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F3.17 of reference [1].

7 REFINEMENT OF DEFINITIONS OF CSTS SPECIFICATION FRAMEWORK PARAMETERS, EVENTS, DIRECTIVES, AND DIAGNOSTIC VALUES USED BY THE CFDP RETURN PDU SERVICE

7.1 OVERVIEW

Except where explicitly refined in this section, the definitions of the parameters, events, directives, and diagnostic values of the operations of the procedures defined in reference [1] that are used by the Return CFDP PDU service are the same as their definitions in reference [1].

7.2 `cfdpPduSvcProductionStatus` PARAMETER DEFINITION REFINEMENT

NOTE – This refined definition applies to the `cfdpPduSvcProductionStatus` parameter, which has the Published Identifier `svcProductionStatusVersion1` in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F3.17 of reference [1].

7.2.1 PRODUCTION CONFIGURED

7.2.1.1 For an RCFDP-CSTS operating in real-time mode, the definition of the 'production configured' value of the `cfdpPduSvcProductionStatus` parameter shall be refined to mean that configuration of the resource performing the configuration function of the involved Functional Resources has been completed.

7.2.1.2 For an RCFDP-CSTS operating in complete mode, the definition of the 'production configured' value of the `cfdpPduSvcProductionStatus` parameter shall be refined to mean that configuration of the resource performing the CFDP Recording Buffer function has been completed.

7.2.2 PRODUCTION INTERRUPTED

7.2.2.1 For an RCFDP-CSTS operating in real-time mode and complete mode, the definition of the 'production interrupted' value of the `cfdpPduSvcProductionStatus` parameter shall be refined to mean that the resource performing the CFDP PDU production has been stopped because of a condition that may be temporary.

7.2.3 PRODUCTION HALTED

7.2.3.1 For an RCFDP-CSTS operating in real-time mode and complete mode, the definition of the 'production halted' value of the `cfdpPduSvcProductionStatus` parameter shall be refined to mean that the resource performing the CFDP PDU production has been stopped by management action.

7.2.4 PRODUCTION OPERATIONAL

7.2.4.1 For an RCFDP-CSTS operating in real-time mode and complete mode, the definition of the 'production operational' value of the `cfdpPduSvcProductionStatus` parameter shall be refined to mean that the resource performing the CFDP PDU production has changed to 'operational'.

7.3 `cfdpPduSvcProductionStatusChange` EVENT DEFINITION REFINEMENT

NOTE – This refined definition applies to the `cfdpPduSvcProductionStatusChange` event, which has the Published Identifier `svcProductionStatusChangeVersion1` defined in the CCSDS-CSTS-GENERIC-SERVICE-OBJECT-IDENTIFIERS module in F3.17 of reference [1].

7.3.1 For an RCFDP-CSTS operating in real-time mode, the definition of the `event-value` of the `cfdpPduSvcProductionStatusChange` event shall be refined to mean that the change refers to the resource performing the CFDP PDU Generation.

7.3.2 For an RCFDP-CSTS operating in complete mode, the definition of the `event-value` of the `cfdpPduSvcProductionStatusChange` event shall be refined to mean that the change refers to the resource performing the CFDP PDU Recording Buffer function.

ANNEX A

IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the ICS Requirements List (RL) for an implementation of the *Cross Support Transfer Service—Return CFDP PDU Service*, CCSDS 922.27-O-1. CCSDS 922.27-O-1 specifies the requirements on the provider of the Return CFDP PDU Cross Support Transfer Service.

The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation shall satisfy the mandatory conformance requirements reference in the RL.

The RL support column in this annex is blank. An implementation's completed RL is called the PICS. The PICS states which capabilities and options have been implemented. The following can use the PICS:

a) the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);

d) a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

The RL consists of information in tabular form. The status of features is indicated using the abbreviations and conventions described below.

Item Column

The item column contains a prefix identifying the element the given table is referring to and sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means: 'is this feature supported by the implementation?'

Status Column

The status column uses the following notations:

a) M mandatory;

b) O optional;

c) O.<n> optional, but support of at least one of the group of options labeled by the same numeral <n> is required;

d) C<n> conditional as defined in corresponding expression below the table;

e) X prohibited;

f) N/A not applicable.

Support Column Symbols

The support column is to be used by the implementer to state whether a feature is supported by entering Y, N, or N/A, indicating:

a) Y Yes, supported by the implementation;

b) N No, not supported by the implementation;

c) N/A Not applicable.

The support column should also be used, when appropriate, to enter values supported for a given capability.

Allowed Values Column

All PDU parameter types are specified in annex F of reference [1] or in this Experimental Specification using ASN.1. The ASN.1 data type specifications constrain, among others, the permissible value range, and therefore such constraints are not repeated in the Allowed Values column in the tables contained in this ICS annex. However, if a parameter is constrained for all instances of the given PDU to a subset of the range or set specified for that parameter type, then the subset is identified in the tables that contain PDU parameters.

Allowed Values Column Symbols

If the allowed values are too large to fit in the Allowed Values cell, the Allowed Values column uses the notation 'AV<n>' to indicate that the allowed values are specified below the table.

Supported Values Column

The Supported Values column is to be used by the implementer to state whether the specified range or set of values for the parameter is supported by entering Y or SV<n>, indicating:

a) Y — Yes, the range/set defined in the Experimental Specification is fully supported by the implementation;
b) SV<n> — The range/set defined in the Experimental Specification is not fully supported by the implementation. The supported subset is documented below the table.

A1.3 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Experimental Specification by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called a PICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 PICS PROFORMA FOR THE RETURN CFDP PDU CSTS PROTOCOL (CCSDS 922.27-O-1)

A2.1 GENERAL INFORMATION

The PICS for an RCFDP-CSTS implementation shall encompass the filled-in tables A-1 to A-4.

Table A-1: Identification of PICS

Date of Statement (DD/MM/YYYY)	
PICS serial number	
System Conformance statement cross-reference	

Table A-2: Identification of Implementation Under Test

Implementation name	
Implementation version	
Special Configuration	
Other Information	

Table A-3: Identification of Supplier

Supplier	
Contact Point for Queries	
Implementation Name(s) and Versions	
Other information necessary for full identification, for example, name(s) and version(s) for machines and/or operating systems; System Name(s)	

Table A-4: Identification of Specification

CCSDS 922.27-O-1	
Have any exceptions been required? NOTE – A YES answer means that the implementation does not conform to the Experimental Specification. Non-supported mandatory capabilities are to be identified in the PICS, with an explanation of why the implementation is nonconforming.	Yes [ ] No [ ]

A2.2 REQUIREMENTS LIST

This subsection provides the Requirement Lists for the elements specified in this Experimental Specification.

Table A-5: Required Procedures

Procedures				
Item	**Description**	**Reference**	**Status**	**Support**
proc-1	Association Control	Subsection 4.3 of reference [1]	M	
proc-2	Buffered Data Delivery	Subsection 4.5 of reference [1]	M	
proc-3	CFDP PDU Delivery	Section 4 of this Experimental Specification	M	

The Buffered Data Delivery procedure is mandatory in the sense that the CFDP PDU Delivery procedure (which is mandatory) is derived from the Buffered Data Delivery procedure. In this RCFDP-CSTS ICS, all requirements for the Buffered Data Delivery procedure are covered by the requirements for the CFDP PDU Delivery procedure.

Table A-6: Required PDUs

Item	PDU	Reference	Service-Provider-System		Service-User-System	
			Status	Support	Status	Support
pdu-1	BindInvocation	Subsection F3.5 of reference [1]	M		M	
pdu-2	BindReturn	Subsection F3.5 of reference [1]	M		M	
pdu-3	PeerAbortInvocation	Subsection F3.5 of reference [1]	M		M	
pdu-4	UnbindInvocation	Subsection F3.5 of reference [1]	M		M	
pdu-5	UnbindReturn	Subsection F3.5 of reference [1]	M		M	
pdu-6	StartInvocation	Subsection F3.4 of reference [1]	M		M	
pdu-7	StartReturn	Subsection F3.4 of reference [1]	M		M	
pdu-8	StopInvocation	Subsection F3.4 of reference [1]	M		M	
pdu-9	StopReturn	Subsection F3.4 of reference [1]	M		M	
pdu-10	TransferDataInvocation	Subsection F3.4 of reference [1]	M		M	
pdu-11	ReturnBuffer	Subsection F3.7 of reference [1]	M		M	

Table A-7: BIND Invocation Parameters

Parameters of the BindInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
bindInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
bindInv-2	invokeId	Subsection F3.3 of reference [1]	M			
bindInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV1	
bindInv-4	initiatorIdentifier	Subsection F3.5 of reference [1]	M			
bindInv-5	responderPortIdentifier	Subsection F3.5 of reference [1]	M			
bindInv-6	serviceType	Subsection F3.5 of reference [1]	M			
bindInv-7	versionNumber	Subsection F3.5 of reference [1]	M			
bindInv-8	serviceInstanceIdentifier	Subsection F3.5 of reference [1]	M			
bindInv-9	bindInvocationExtension	Subsection F3.5 of reference [1]	M		‘notUsed’	

AV1 For the BIND invocation, the procedureRole element of the parameter bindInv-3 must be set to ‘associationControl’.

The parameters bindInv-1, bindInv-2, and bindInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.5 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of reference [1].

Table A-8: BIND Return Parameters

Parameters of the BindReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
bindRet-1	performerCredentials	Subsection F3.3 of reference [1]	M			
bindRet-2	invokeId	Subsection F3.3 of reference [1]	M			
bindRet-3	result	Subsection F3.3 of reference [1]	M			
bindRet-4	positive	Subsection F3.3 of reference [1]	C1		'notUsed'	
bindRet-5	diagnostics	Subsection F3.3 of reference [1]	C2		AV2	
bindRet-6	negExtension	Subsection F3.3 of reference [1]	C2		'notUsed'	
bindRet-7	responderIdentifier	Subsection F3.5 of reference [1]	M			

C1 IF bindRet-3 = 'positive', THEN M, ELSE X

C2 IF bindRet-3 = 'negative', THEN M, ELSE X

AV2 For the negative BIND return, the parameter bindRet-5 is extended by the type AssocBindDiagnosticExt defined in F3.5 of reference [1]. Therefore the parameter bindRet-5 may have (a) any value defined for the `Diagnostic` type in F3.3 of reference [1] except 'diagnosticExtension'; or (b) any value defined by 'diagnosticExtension': 'acBindDiagExt': 'AssocBindDiagnosticExt' defined in F3.5 of reference [1] except 'assocBindDiagnosticExtExtension'.

All parameters of the BIND return PDU except bindRet-7 are contained the complex parameter of the type StandardReturnHeader specified in F3.3 of reference [1]. Specific extensions are, however, specified in F3.5 of that document.

Table A-9: PEER-ABORT Invocation Parameters

Parameters of the PeerAbortInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
peerAbortInv-1	diagnostic	Subsection F3.5 of reference [1]	M		40..126	

Table A-10: UNBIND Invocation Parameters

Parameters of the UnbindInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
unbindInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
unbindInv-2	invokeId	Subsection F3.3 of reference [1]	M			
unbindInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV3	
unbindInv-4	unbindInvocationExtension	Subsection F3.5 of reference [1]	M		'notUsed'	

AV3 For the UNBIND invocation, the procedureRole element of the parameter unbindInv-3 must be set to 'associationControl'.

The parameters unbindInv-1, unbindInv-2, and unbindInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.5 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of that document.

Table A-11: UNBIND Return Parameters

Parameters of the UnbindReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
unbindRet-1	performerCredentials	Subsection F3.3 of reference [1]	M			
unbindRet-2	invokeId	Subsection F3.3 of reference [1]	M			
unbindRet-3	result	Subsection F3.3 of reference [1]	M		AV4	

AV4 The value of the parameter unbindRet-3 shall always be set to 'positive': 'notUsed'; that is, the result is always positive and not extended.

All parameters of the UNBIND return PDU are contained the complex parameter of the type StandardReturnHeader specified in F3.3 of reference [1].

Table A-12: START Invocation Parameters

Parameters of the StartInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
startInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
startInv-2	invokeId	Subsection F3.3 of reference [1]	M			
startInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV5	
startInv-4	startInvocationExtension	Subsection F3.4 of reference [1]	M		AV6	
startInv-5	startGenerationTime	Subsection F3.7 of reference [1]	M			
startInv-6	stopGenerationTime	Subsection F3.7 of reference [1]	M			
startInv-7	buffDataDelStartInvocExt Extension	Subsection F3.7 of reference [1]	M		‘notUsed’	

AV5 The value of the procedureRole element of the parameter startInv-3 must be set to ‘prime procedure’.

AV6 The parameter startInv-4 shall be set to the value ‘bddStartInvocExt’: ‘BuffDataDelStartInvocExt’.

The parameters startInv-1, startInv-2, and startInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.4 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of that document.

Table A-13: START Return Parameters

Parameters of the StartReturn PDU						
					Values	
Item	Parameter	Reference	Status	Support	Allowed	Supported
startRet-1	performerCredentials	Subsection F3.3 of reference [1]	M			
startRet-2	invokeId	Subsection F3.3 of reference [1]	M			
startRet-3	result	Subsection F3.3 of reference [1]	M			
startRet-4	positive	Subsection F3.3 of reference [1]	C3		AV7	
startRet-5	diagnostics	Subsection F3.3 of reference [1]	C4		AV8	
startRet-6	negExtension	Subsection F3.3 of reference [1]	C4		'notUsed'	
startRet-7	buffTrkDataDelStartPos ReturnExtExtension	Annex C	C3		'notUsed'	

C3 IF startRet-3 = 'positive', THEN M, ELSE X

C4 IF startRet-3 = 'negative', THEN M, ELSE X

AV7 The parameter startRet-4 has the value 'buffTrkDataDelStartPosReturnExt': 'BuffTrkDataDelStartPosReturnExt', as defined in annex C.

AV8 For the START return PDU, the parameter startRet-5 is extended by the types StartDiagnosticExt defined in F3.4 and BuffDataDelStartDiagnosticExt defined in F3.7 of reference [1]. Therefore the parameter startRet-5 may have (a) any standard value defined for the `Diagnostic` type in F3.3 of that document except 'diagnosticExtension'; (b) any value defined by the extension 'diagnosticExtension': 'startDiagnosticExt': 'StartDiagnosticExt' in F3.4 of that document except 'startDiagnosticExtExtension'; or (c) any value defined by the extension 'diagnosticExtension': 'startDiagnosticExt': 'StartDiagnosticExt': 'startDiagnosticExtExtension': bddStartDiagExt': 'BuffDataDelStartDiagnosticExt' in F3.7 of that document except 'buffDataDelStartDiagnosticExtExtension'.

Table A-14: STOP Invocation Parameters

Parameters of the StopInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
stopInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
stopInv-2	invokeId	Subsection F3.3 of reference [1]	M			
stopInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV9	
stopInv-4	stopInvocationExtension	Subsection F3.4 of reference [1]	M		'notUsed'	

AV9 The value of the procedureRole element of the parameter stopInv-3 must be set to 'prime procedure'.

The parameters stopInv-1, stopInv-2, and stopInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.4 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of that document.

Table A-15: STOP Return Parameters

Parameters of the StopReturn PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
stopRet-1	performerCredentials	Subsection F3.3 of reference [1]	M			
stopRet-2	invokeId	Subsection F3.3 of reference [1]	M			
stopRet-3	result	Subsection F3.3 of reference [1]	M			
stopRet-4	positive	Subsection F3.3 of reference [1]	C5		‘notUsed’	
stopRet-5	diagnostics	Subsection F3.3 of reference [1]	C6		AV10	
stopRet-6	negExtension	Subsection F3.3 of reference [1]	C6		‘notUsed’	

C5 IF stopRet-3 = ‘positive’, THEN M, ELSE X

C6 IF stopRet-3 = ‘negative’, THEN M, ELSE X

AV10 The parameter stopRet-5 may have any standard value defined for the `Diagnostic` type in F3.3 of reference [1] except ‘diagnosticExtension’.

All parameters of the STOP return PDU are contained in the complex parameter of the type StandardReturnHeader specified in F3.3 of reference [1].

Table A-16: TRANSFER-DATA Invocation Parameters

Parameters of the TransferDataInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
transferDataInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
transferDataInv-2	invokeId	Subsection F3.3 of reference [1]	M			
transferDataInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV11	
transferDataInv-4	generationTime	Subsection F3.4 of reference [1]	M			
transferDataInv-5	sequenceCounter	Subsection F3.4 of reference [1]	M			
transferDataInv-6	data	Subsection F3.4 of reference [1]	M		AV12	
transferDataInv-7	transferDataInvocation Extension	Subsection F3.4 of reference [1]	M		'notUsed'	

AV11 The value of the procedureRole element of the parameter transferDataInv-3 must be set to 'prime procedure'.

AV12 The value of transferDataInv-6 is extended through the definition of the extendedData to be a sequence of octet strings, each formatted as a CFDP PDU as defined by reference [3].

The parameters transferDataInv-1, transferDataInv-2, and transferDataInv-3 are contained in the complex parameter standardInvocationHeader shown in F3.4 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of that document.

Table A-17: NOTIFY Invocation Parameters

Parameters of the NotifyInvocation PDU						
Item	Parameter	Reference	Status	Support	Values	
					Allowed	Supported
notifyInv-1	invokerCredentials	Subsection F3.3 of reference [1]	M			
notifyInv-2	invokeId	Subsection F3.3 of reference [1]	M			
notifyInv-3	procedureInstanceId	Subsection F3.3 of reference [1]	M		AV13	
notifyInv-4	eventTime	Subsection F3.4 of reference [1]	M			
notifyInv-5	eventName	Subsection F3.4 of reference [1]	M			
notifyInv-6	eventValue	Subsection F3.4 of reference [1]	M		AV14	
notifyInv-7	notifyInvocationExtension	Subsection F3.4 of reference [1]	M		'notUsed'	

AV13 The value of the procedureRole element of the parameter notifyInv-3 must be set to 'prime procedure'.

AV14 The value of the notifyInv-6 parameter can be any value that can be expressed using the type SequenceOfQualifiedValues defined in F3.3 of reference [1] or 'empty'. The value of 'eventValue' must not be set to 'eventValueExtension'.

The parameters notifyInv-1, notifyInv-2, and notifyInv-3 are contained in the complex parameter standardInvocationHeader in the NotifyInvocation type shown in F3.4 of reference [1]. This parameter is of the type StandardInvocationHeader specified in F3.3 of that document.

ANNEX B

SERVICE OBJECT IDENTIFIERS MODULE

(NORMATIVE)

This annex defines the Object Identifiers for the CSTS Return CFDP service, as requested by annex D5 in reference [1].

```
CCSDS-CSTS-RTN-CFDP-PDU-OBJECT-IDENTIFIERS
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) rtnCfdpPduService(100)
   rtnCfdpPduServiceModules(4)   object-identifiers(1) version(1)
}

DEFINITIONS

IMPLICIT TAGS
::= BEGIN

EXPORTS  rtnCfdpPduBuffDelExtendedOpsParam
,        rtnCfdpPduDerivedServices
,        rtnCfdpPduExtendedServiceParameters
,        rtnCfdpPduServiceProcedures
;

IMPORTS  services
,        crossSupportFunctionalities
   FROM CCSDS-CSTS-OBJECT-IDENTIFIERS

         BufferSize
   FROM CCSDS-CSTS-COMMON-TYPES
;

-- The CCSDS-CSTS-OBJECT-IDENTIFIERS and CCSDS-CSTS-COMMON-TYPES modules
-- are defined in F3.1 and F3.3 of reference [1].

-- ******************************************************
-- Root Object Identifiers of the Service

rtnCfdpPduService OBJECT IDENTIFIER ::= {services 100}
rtnCfdpPduDerivedServices  OBJECT IDENTIFIER ::=
                                              {rtnCfdpPduService 1}
rtnCfdpPduExtendedServiceParameters OBJECT IDENTIFIER ::=
{rtnCfdpPduService 2}
rtnCfdpPduServiceProcedures   OBJECT IDENTIFIER ::=
                                                {rtnCfdpPduService 3}
rtnCfdpPduServiceModules   OBJECT IDENTIFIER ::= {rtnCfdpPduService 4}
```

```
-- ********************************************************
-- Procedure Type Identifier:
rtnCfdpPduBuffDel    OBJECT IDENTIFIER ::=
                                  {rtnCfdpPduServiceProcedures 1}

rtnCfdpPduBuffDelExtendedOpsParam   OBJECT IDENTIFIER ::=
                                    {rtnCfdpPduBuffDel     1}
rtnCfdpPduBuffDelExtendedProcParam  OBJECT IDENTIFIER ::=
                                    {rtnCfdpPduBuffDel     2}
rtnCfdpPduBuffDelEventsId           OBJECT IDENTIFIER ::=
                                    {rtnCfdpPduBuffDel     3}
rtnCfdpPduBuffDelDirectivesId       OBJECT IDENTIFIER ::=
                                    {rtnCfdpPduBuffDel     4}

-- ********************************************************
-- Root Object Identifiers of the RCFDP-CSTS Provider Functional
-- Resource Type

-- The Object Identifiers of the RCFDP-CSTS Provider FR type are specified and
-- registered in the SANA Functional Resources registry (reference [5])
-- under the crossSupportFunctionalities subtree. The root of the RCFDP-
CSTS
-- Provider FR type OID has the classifier cfdpPduCstsProvider.
-- All parameters of the RCFDP-CSTS Provider FR shall be registered under the
-- node of the cfdpPduCstsProvider subtree that has the OID
{cfdpPduCstsProvider 1}.
-- All events of the RCFDP-CSTS Provider FR shall be registered under the
-- node of the cfdpPduCstsProvider subtree that has the OID
{cfdpPduCstsProvider 2}.
-- All directives of the RCFDP-CSTS Provider FR shall be registered under the
-- node of the cfdpPduCstsProvider subtree that has the OID
{cfdpPduCstsProvider 3}.


END
```

ANNEX C

PROCEDURE – CFDP RETURN DELIVERY PDUS

(NORMATIVE)

C1 CFDP RETURN CFDP-RETURN-DELIVERY-PDUS MODULE

This annex defines the data types used for the data parameter of the Transfer Data operation, which is used by the CFDP PDU Delivery Procedure defined in section 4. More precisely, the data parameter of the Transfer Data operation as defined by reference [1] is *extended* to deliver CFDP PDUs.

```
CCSDS-CSTS-RTN-CFDP-PDU-BUFFERED-DELIVERY-PDUS
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) rtnCfdpPduService(100)
   rtnCfdpPduServiceModules(4) extensions(2) rtnCfdpPduBuffDelPdus(1)
   version(1)
}

DEFINITIONS
IMPLICIT TAGS
::= BEGIN

EXPORTS  CfdpDeliveryPdu
;

IMPORTS  BufferedDataDeliveryPdu
   FROM CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS
-- CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS is defined in F3.7 of reference
-- [1]).
         Extended
   FROM CCSDS-CSTS-COMMON-TYPES
         rtnCfdpPduBuffDelExtendedOpsParam
   FROM CCSDS-CSTS-RTN-CFDP-PDU-OBJECT-IDENTIFIERS
;

-- ============================================================
-- The CFDP PDU Delivery procedure is derived from
-- the Buffered Data Delivery procedure. It reuses the PDU defined in the
-- Buffered Data Delivery procedure: BufferedDataDeliveryPdu type defined
-- in the CCSDS-CSTS-BUFFERED-DATA-DELIVERY-PDUS module in F3.7 of
-- reference [1].
```

```
CfdpDeliveryPdu  ::=  BufferedDataDeliveryPdu

CfdpDeliveryPduData ::= SEQUENCE OF OCTET STRING

cfdpDeliveryPduDataId OBJECT IDENTIFIER ::=
                  {rtnCfdpPduBuffDelExtendedOpsParam  1}


-- START negative return extension parameters
-- No extension parameters are added to the START negative return of the
-- BufferedDataDelivery procedure. Therefore 'StartReturn':
-- 'StandardReturnHeader': 'result': 'negative': 'negExtension' (refer to
-- the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F3.4 of reference [1])
-- shall be set to 'notUsed'.

-- START negative return extension diagnostics
-- No extension diagnostics are added to the START negative return of the
-- BufferedDataDelivery procedure. Therefore the negative StartReturn is
-- returned using one of the common diagnostics of 'StandardReturnHeader':
-- 'result': 'negative': 'diagnostic': 'Diagnostic' (refer to 3.3.2.7 of
-- reference [1]) or one of the additional diagnostics defined by
-- 'StartReturn': 'StandardReturnHeader': 'result': 'negative':
-- 'diagnostic': 'Diagnostic': 'diagnosticExtension':
-- 'startDiagnosticExt': 'StartDiagnosticExt' or any of the additional
-- values defined by 'StartReturn': 'StandardReturnHeader': 'result':
-- 'negative': 'diagnostic': 'Diagnostic': 'diagnosticExtension':
-- 'startDiagnosticExt': 'StartDiagnosticExt':
-- 'startDiagnosticExtExtension': 'bddStartDiagExt':
-- 'BuffDataDelStartDiagnosticExt' except
-- buffDataDelStartDiagnosticExtExtension.

-- *****
-- STOP Invocation extension parameters
-- No extension parameters are added to the STOP Invocation of the
-- BufferedDataDelivery procedure. Therefore 'StopInvocation':
-- 'stopInvocationExtension'(refer to the CCSDS-CSTS-COMMON-OPERATIONS-PDUS
-- module in F3.4 of reference [1]) shall be set to 'notUsed'.

-- STOP positive return extension parameters
-- No extension parameters are added to the STOP positive return of the
-- BufferedDataDelivery procedure. Therefore 'StopReturn':
-- 'StandardReturnHeader': 'result': 'positive' (refer to the
-- CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F3.4 of reference [1]) shall
-- be set to 'notUsed'.

-- STOP negative return extension parameters
-- No extension parameters are added to the STOP negative return of the
-- BufferedDataDelivery procedure. Therefore 'StopReturn':
-- 'StandardReturnHeader': 'result': 'negative': 'negExtension' (refer to
-- the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F3.4 of reference [1])
-- shall be set to 'notUsed'.

-- STOP negative return extension diagnostics
-- The negative StopReturn is returned using one of the common diagnostics
-- of 'StandardReturnHeader': 'result': 'negative': 'diagnostic':
-- 'Diagnostic' (refer to 3.3.2.7 of reference [1]). No additional
-- diagnostics are specified, that is, 'StopReturn': 'StandardReturnHeader':
-- 'result': 'negative': 'diagnostic': 'Diagnostic' must not be set to
-- 'diagnosticExtension'.
```

```
-- *****
-- TRANSFER-DATA Invocation extension parameters
-- No extension parameters are added to the TRANSFER-DATA Invocation of the
-- BufferedDataDelivery procedure. Therefore 'TransferDataInvocation':
-- transferDataInvocationExtension (refer to the
-- CCSDS-CSTS-COMMON- OPERATIONS-PDUS module in F3.4 of reference [1])
-- shall be set to 'notUsed'.

-- TRANSFER-DATA Invocation data parameter resolution
-- The data parameter of the TRANSFER-DATA Invocation is resolved as an
-- octet string. Therefore 'TransferDataInvocation': data':
-- 'AbstractChoice' (refer to the CCSDS-CSTS-COMMON-OPERATIONS-PDUS module
-- in F3.4 of reference [1]) shall be set to 'opaqueString'.

-- *****
-- NOTIFY Invocation extension
-- No extension parameters are added to the NOTIFY Invocation of the
-- BufferedDataDelivery. Therefore 'NotifyInvocation':
-- 'notifyInvocationExtension' (refer to the
-- CCSDS-CSTS-COMMON-OPERATIONS-PDUS module in F3.4 of reference [1]) shall
-- be set to 'notUsed'.

END
```

C2 TRANSFER SYNTAX

N/A.

ANNEX D

CFDP RETURN PDU SERVICE PROCEDURE PARAMETERS, EVENTS, AND DIRECTIVES

(NORMATIVE)

```
CCSDS-CSTS-RTN-CFDP-PDU-SERVICE-PROCEDURE-PARAMETERS-EVENTS-DIRECTIVES
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) csts(1) services(2) serviceIdentifiers(2)
   rtnCfdpPduService(100) rtnCfdpPduServiceModules(4)
   procedureParamEventDirective(3) version(1)
}

DEFINITIONS
IMPLICIT TAGS
::=   BEGIN

IMPORTS  rtnCfdpPduBuffDelExtendedProcParam
FROM CCSDS-CSTS-RTN-CFDP-PDU-OBJECT-IDENTIFIERS
;

-- ============================================================

-- CFDP PDU Delivery

pCFDPPDcfdpPduOperationMode   OBJECT IDENTIFIER ::=
                           {rtnCfdpPduBuffDelExtendedProcParam    1}

PCFDPPDcfdpPduOperationModeType ::= ENUMERATED
{
   fullCfdpData      (0)
,  reducedCfdpData   (1)
}

pCFDPPDcfdpDestEntities OBJECT IDENTIFIER ::=
                           {rtnCfdpPduBuffDelExtendedProcParam    2}

PCFDPPDcfdpDestEntitiesType ::= SEQUENCE OF INTEGER

-- CFDP PDU Aggregation
pCFDPPDcfdpPduAggregationMaxSize OBJECT IDENTIFIER ::=
                         {rtnCfdpPduBuffDelExtendedProcParam 3}

PCFDPPDcfdpPduAggregationMaxSizeType ::= INTEGER

pCFDPPDcfdpPduAggregationTimeout OBJECT IDENTIFIER ::=
                         {rtnCfdpPduBuffDelExtendedProcParam 4}

PCFDPPDcfdpPduAggregationTimeoutType ::= INTEGER

END
```

ANNEX E

CFDP PDU PRODUCTION

(NORMATIVE)

E1 SPECIFICATION OF THE CFDP PDU GENERATION FUNCTION

E1.1 CFDP PDU GENERATION FUNCTION BEHAVIOR

E1.1.1 The CFDP PDU Generation function is an abstraction of a Functional Resource that is able to provide CFDP PDUs as defined in reference [3]. In terms of Functional Resources, a Functional Resource with classifier `CfdpEntity` is the Functional Resource producing CFDP PDUs. However, the way in which the CFDP PDUs are provided to an instance of the RCFDP-CSTS represented by a Functional Resource instance with the classifier `RcfdpProvider` is dependent on the scenario. In one possible scenario, CFDP PDUs are produced by a local instance of `CfdpEntity` on the ground, passed to the `CfdpRecordingBuffer`, and finally passed to the `RcfdpProvider`. In another scenario, the `CfdpRecordingBuffer` and, finally, the `RcfdpProvider` are provided with CFDP PDUs received on ground, which have been produced by a `CfdpEntity` on board the spacecraft.

E1.1.2 If the CFDP PDU Delivery Procedure parameter `cfdp-pdu-operation-mode` is set to `reducedCfdpData`, all CFDP File Data PDUs are reduced in terms of file data; that is, they do not provide any file data in the CDP File Data PDUs, but only:

a) the CFDP PDU Header with an updated PDU Data field length field according to the new size of the PDU Data field;

b) the Checksum as calculated over the real file data.

E1.1.3 CFDP File Directive PDU are also in reducedCfdpData mode as they are produced by the CFDP PDU Generation function.

E1.1.4 For the value `fullCfdpData` of parameter `cfdp-pdu-operation-mode`, the CFDP PDU CFDP PDU Delivery Procedure delivers CFDP PDUs as described in reference [3]. In other words, the CFDP File Data PDUs are not reduced.

If `cfdp-pdu-dest-entities` contains one or more CFDP entity IDs, only CFDP PDUs matching one of the configured destination entities are forwarded. Matching is based on the entity ID of each CFDP PDU.

E2 SPECIFICATION OF THE CFDP PDU RECORDING BUFFER FUNCTION

E2.1 CFDP PDU RECORDING BUFFER FUNCTION BEHAVIOR

E2.1.1 The functionality of the CFDP PDU Recording Buffer shall be as that specified for Recording Buffer in 4.5.7 of reference [1].

E2.1.2 The Service Production Data Units recorded by the CFDP PDU Recording Buffer function shall be CFDP PDUs generated by the CFDP PDU Generation function (E1).

E2.1.3 The CFDP PDU Recording Buffer function shall make its current status available through the `cfdpPduRecordingBufferStatus` parameter, with values 'configured', 'operational', interrupted', and 'halted'.

E2.1.4 The CFDP PDU Recording Buffer function shall generate a `cfdpPduRecordingBufferStatusChange` event for each change of status, with the `event-value` equal to the value of the `cfdpPduRecordingBufferStatus` parameter after the status change.

E2.1.5 The CFDP PDU Recording Buffer shall make the size of the recording buffer available through the `cfdpPduRecordingBufferSize` parameter, which conforms to the specification of the queryable `recording-buffer-size` parameter defined in 4.5.7.10 of reference [1].

E2.1.6 In accordance with 4.5.7.5 b) of reference [1], if the recording buffer overflows, the CFDP PDU Recording Buffer FR type shall emit a `cfdpPduFrRecordingBufferOverflow` event which conforms to the specification of the 'fr recording buffer overflow' event. The `event-value` for the `cfdpPduFrRecordingBufferOverflow` event shall be empty.

E2.2 CFDP PDU RECORDING BUFFER FUNCTION SERVICE MANAGEMENT INFORMATION

NOTE – Reference [1] states in 4.5.7.5 that the 'time span over which data is retained in the recording buffer, the policy for deleting data from the recording buffer, and the conditions under which the recording buffer begins to accept data following an overflow condition are outside the scope of" reference [1]. They are also outside the scope of this Experimental Specification.

E2.3 CFDP PDU RECORDING BUFFER FUNCTIONAL RESOURCE TYPE REGISTRATION DESCRIPTION

E2.3.1 The OID for the Functional Resource Type that represents the CFDP PDU Recording Buffer function is specified in the SANA Functional Resource Registry (reference [5]), using the FR classifier `cfdpPduRecordingBuffer`.

E2.3.2 The cfdpPduRecordingBufferStatus and cfdpPduRecordingBufferSize parameters are registered under the cfdpPduRecordingBufferParametersId ({cfdpPduRecordingBuffer 1}) node of the SANA FR registry.

E2.3.3 The cfdpPduRecordingBufferStatusChange and cfdpPduFrRecordingBufferOverflow events are registered under the cfdpPduGeneventsId ({cfdpPduGen 2}) node of the SANA FR registry.

E2.3.4 Any other monitored parameters, notifiable events or directives that may be defined for the CFDP PDU Recording Buffer FR type will be registered under the cfdpPduRecordingBufferParametersId, cfdpPduRecordingBufferEventsId, or cfdpPduRecordingBufferDirectivesId ({cfdpPduRecordingBuffer 3}) nodes, respectively, of the SANA FR registry.

ANNEX F

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

F1 SECURITY CONSIDERATIONS

F1.1 INTRODUCTION

This annex subsection describes security aspects of the Return CFDP PDU service.

Reference [1] explicitly provides authentication and access control for CSTSes. As one in the suite of CSTSes, the Return CFDP PDU service inherits the authentication and access control capabilities defined in reference [1]. The Return CFDP PDU service provides no service-specific security capabilities. As specified in reference [1], additional security capabilities, if required, are levied on the underlying communications services that support the RCFDP-CSTS. Specification of the various underlying communications technologies and, in particular, their associated security provisions, are outside the scope of this Experimental Specification.

F1.2 SECURITY CONCERNS WITH RESPECT TO THE RETURN CFDP PDU SERVICE

The Statements of Security Concerns subsection (refer to H1 of reference [1]) identifies the support for capabilities that respond to security concerns in the areas of data privacy (also known as confidentiality), data integrity, authentication, access control, availability of resources, and auditing.

F1.3 POTENTIAL THREATS AND ATTACK SCENARIOS

As a member of the suite of CSTSes, the R-CFDP PDU service depends on unspecified mechanisms operating in the underlying communications service, or on privacy-ensuring capabilities in the service-specific application processes that interoperate through the procedures defined in reference [1], to ensure data privacy (confidentiality). If no such mechanisms are actually implemented, or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could read the data contained in the RCFDP-CSTS protocol data units as they traverse the WAN between service user and service provider.

Reference [1] constrains the ability of a third party to seize control of an active CSTS instance, but it does not specify mechanisms that would prevent an attacker from intercepting the PDUs.

If the CSTS authentication capability is not used, and if authentication is not ensured by the underlying communications service, attackers could somehow obtain valid `initiator-`

`identifier` values and use them to initiate RCFDP-CSTS instances by which they could gain access to the CFDP PDUs transferred via the service.

The RCFDP-CSTS depends on unspecified mechanisms operating in the underlying communications service to ensure that the supporting network has sufficient resources to provide sufficient support to legitimate service users. If no such mechanisms are actually implemented or the mechanisms selected are inadequate or inappropriate to the network environment in which the mission is operating, an attacker could prevent legitimate service users from using the RCFDP-CSTS.

If the service provider of the RCFDP-CSTS provides no security auditing capabilities, or if a service user chooses not to employ auditing capabilities that do exist, then attackers may delay or escape detection while stealing data exchanged via the service.

F1.4 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The consequences of not applying security to the RCFDP-CSTS are possible degradation and loss of ability to use the service, or the interception of CFDP PDUs that could lead to corruption of files transferred by means of the CFDP PDUs returned by this service.

F2 SANA CONSIDERATIONS

The RCFDP-CSTS relies on the SANA Functional Resource Registry (reference [5]) to provide the identification and definition of Functional Resource parameters and events.

As described in this Experimental Specification, the RCFDP-CSTS reports parameters and events that are named in the context of Functional Resources. Functional Resource Types are registered under the

```
{  iso(1) identified-organization(3) standards-producing-organization(112)
   ccsds(4) css(4) crossSupportResources(2)
}
```

node of the OID registration tree.

There are two subnodes under the `crossSupportResources` node: `crossSupportFunctionalities` and `agencyFunctionalities`, used to register CCSDS-standard Functional Resource Types and agency-unique Functional Resource Types, respectively. Agency-unique Functional Resource Types are not relevant in the RCFDP-CSTS context. Under each Functional Resource Type OID, the parameters, events, and directives are registered under dedicated subnodes.

The RCFDP-CSTS relies on the SANA Functional Resource registry to register the following:

– the RCFDP-CSTS Provider Functional Resource Type, registered under the `crossSupportFunctionalities` with the classifier `cfdpPduCstsProvider`;

– the parametersId node of the `cfdpPduCstsProvider` subtree (`{cfdpPduCstsProvider 1}`);

– the eventsId node of the `cfdpPduCstsProvider` subtree (`{cfdpPduCstsProvider 2}`);

– the directivesId node of the `cfdpPduCstsProvider` subtree (`{cfdpPduCstsProvider 3}`).

Maintenance of the SANA registry of the Functional Resource Types, parameters, events, and directives under the `crossSupportFunctionalities` subnode is under the purview of the CCSDS Cross Support Services Area in accordance with the process and procedures identified in the CSTS Specification Framework (reference [1]).

The positions in the OID tree where the OIDs of the Functional Resources associated with the RCFDP-CSTS are registered are specified in E2.3 and E2.3. The OID values are defined in annex B.

F3 PATENT CONSIDERATIONS

There are no patents that are known to apply to the technology used in the Return CFDP PDU service.

ANNEX G

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

ACK	acknowledgement
ASN.1	Abstract Syntax Notation One
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
CFDPPD	CFDP PDU Delivery (procedure)
CLTU	Communication Link Transmission Unit
CM	Complex Management
CSSE	cross support service element
CSSS	cross support service system
CSTS	Cross Support Transfer Service
DOR	Differential One-way Ranging
EOF	end of file
ESLT	Earth-space link terminal
F-CLTU	Forward CLTU
ICS	Implementation Conformance Statement
ISO	International Organization for Standardization
MIB	Management Information Base
MDOS	Mission Data Operation System
NAK	negative acknowledgement
OID	Object Identifier
PDU	Protocol Data Unit
RAF	Return All Frames
RCFDP-CSTS	Return CFDP PDU Cross Support Transfer Service

RF	Radio Frequency
RL	Requirements List
SANA	Space Assigned Numbers Authority
SCCS-SM	Space Communication Cross Support Service Management
SM	Service Management
SLE	Space Link Extension
UM	Utilization Management

ANNEX H

INFORMATIVE REFERENCES

(INFORMATIVE)

[H1] *Cross Support Concept—Part 1: Space Link Extension Services*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 910.3-G-3. Washington, D.C.: CCSDS, March 2006.

[H2] *Space Link Extension—Internet Protocol for Transfer Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 913.1-B-2. Washington, D.C.: CCSDS, September 2015.

[H3] *Cross Support Transfer Service Specification Framework Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 920.0-G-1. Washington, D.C.: CCSDS, May 2023.

[H4] *Space Link Extension—Return All Frames Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 911.1-B-4. Washington, D.C.: CCSDS, August 2016.

[H5] *Space Link Extension—Forward CLTU Service Specification*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 912.1-B-4. Washington, D.C.: CCSDS, August 2016.

[H6] *Extensible Space Communication Cross Support—Service Management—Concept*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 902.0-G-1. Washington, D.C.: CCSDS, September 2014.

[H7] *Cross Support Service Management—Service Management Utilization Request Formats*. Issue 1. Draft Recommendation for Space Data System Standards (Red Book), CCSDS 902.9-R-1. Washington, D.C.: CCSDS, September 2022.

[H8] *Cross Support Service Management—Simple Schedule Format Specification*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 902.1-B-1. Washington, D.C.: CCSDS, May 2018.

ANNEX I

RETURN CFDP PDU CSTS IMPLEMENTATION

(INFORMATIVE)

The European Space Agency has created a prototype and a full, operational implementation supporting all features of this CSTS. The prototype has been based on ESA's Cross-Support Transfer Service Framework implementation and focused on the CFDPPD, which is derived from the Buffered Data Delivery procedure specified in reference [1]; however, it is a full implementation of the service, including all procedures and operations adopted from reference [1]. Another test focus was the feasibility of running a CFDP Protocol State Machine with the input of reduced CFDP PDUs as they are specified within this CSTS. Clearly, the CFDP entity, or more precisely, the CFDP Protocol State Machine of the CFDP entity, must be able to process reduced CFDP File Data PDUs. As it turns out, it has been possible to integrate the processing of reduced CFDP File Data PDU into an existing CFDP implementation with low effort by using a dedicated file store implementation.

An implementation intended for operational use of the Return CFDP PDU CSTS has been created by ESA to realize a distributed CFDP scenario very similar to the one described in 2.4.3. ESA's Distributed CFDP implementation will support the next generation of ESA's Sentinel Missions. In that context the required performance to support CFDP transactions over space links operating at up to 7.5 Gbit/s has been considered and tested. Furthermore, ESA's operational CFDP implementation has been updated to include processing of reduced CFDP PDUs. The Return CFDP PDU CSTS features like the aggregation of potentially reduced CFDP PDUs has been proven to be very useful, which was one of the design goals of this CSTS.