The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

CCSDS FILE DELIVERY PROTOCOL (CFDP)

Draft Recommended Standard

CCSDS 727.0-P-4.2

Pink Sheets
July 2019

The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

CCSDS FILE DELIVERY PROTOCOL (CFDP)

Draft Recommended Standard

CCSDS 727.0-P-4.2

Pink Sheets
July 2019

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 727.0-B-1	CCSDS File Delivery Protocol, Issue 1	January 2002	Original Issue (superseded)
CCSDS 727.0-B-2	CCSDS File Delivery Protocol, Issue 2	October 2002	Issue 2 (superseded)
CCSDS 727.0-B-3	CCSDS File Delivery Protocol, Issue 3	June 2005	Issue 3 (superseded)
CCSDS 727.0-B-4	CCSDS File Delivery Protocol (CFDP), Recommended Standard, Issue 4	January 2007	Current Issue: – adds corrections and clarifications recommended by the CFDP Interoperability Working Group.
CCSDS 727.0-P-4.1	CCSDS File Delivery Protocol (CFDP), Issue 4.1	November 2014	Draft update issue 4.1: – moves Store and Forward Overlay Operations from section 6 to a new normative annex (annex B); – removes Extended Procedures; – adds support for large files, multi-segment records, and per-segment metadata.
CCSDS 727.0-P-4.2	CCSDS File Delivery Protocol (CFDP), Issue 4.2	July 2019	Current draft update: – adds support for choice of checksum algorithm from a new SANA Checksum Types registry; – updates, clarifies some text based on interoperability testing.

4 PROTOCOL SPECIFICATION

4.1 CORE PROCEDURES

4.1.1 CRC PROCEDURES

4.1.1.1 CRC Procedures at the PDU Transmitting Entity

If the CRC option is active, the PDU transmitting entity shall set the CRC flag to 'true' and calculate and insert the CRC for each outgoing PDU.

4.1.1.2 CRC Procedures at the PDU Receiving Entity

If the CRC flag is set to 'true' in the incoming PDU, the PDU receiving entity shall calculate the CRC and discard the PDU if it fails the CRC validation procedure.

4.1.1.3 CRC Validation Procedure

4.1.1.3.1 The CRC computation algorithm shall be the standard CCSDS Telecommand CRC algorithm as specified in 4.2.1.3 of the CCSDS Telecommand Recommendation (reference [4]).

4.1.1.3.2 The CRC value shall be placed in the final octets of the PDU data field, and its length shall be included in the PDU data field length. The CRC algorithm shall be applied from the first octet of the PDU header to the last octet of the PDU data field prior to the CRC value.

4.1.2 CHECKSUM PROCEDURES

~~The checksum shall be 32 bits in length. It shall be either a CRC32 checksum or a modular checksum, depending on the applicable checksum type. For checksum computation at the sending entity, the applicable checksum type shall be the checksum type specified in the Management Information Base remote entity configuration information pertaining to the receiving entity. For checksum computation at the receiving entity, the applicable checksum type shall be the checksum type specified in the Management Information Base remote entity configuration information for reception from the sending entity.~~

4.1.2.1 The integrity of each file conveyed by CFDP shall be protected by means of a checksum that is computed over the transmitted file data and inserted into the EOF (No error) PDU by the sending entity, then re-computed and verified by the receiving entity.

4.1.2.2 The checksum shall be 32 bits in length. It shall be the result of executing the applicable checksum calculation algorithm, which shall be one of the first sixteen 16 algorithms enumerated in the SANA Checksum Types registry.

4.1.2.3 For checksum computation at the sending entity, the applicable checksum calculation algorithm shall be the algorithm identified by the checksum type specified in the Management Information Base remote entity configuration information pertaining to the receiving entity unless overridden for mission purposes (an implementation matter).

4.1.2.4 For checksum computation at the receiving entity, the applicable checksum calculation algorithm shall be the algorithm identified by the checksum type specified in the Metadata PDU.

4.1.2.5 When an entity is required to perform checksum calculation of a given type but is for any reason unable to do so, the value of the checksum shall be zero.

~~The CRC32 checksum shall be calculated by the method documented in Annex C of the Proximity-1 Space Link Protocol Coding and Synchronization Sublayer Blue Book (CCSDS 211.2-B-2).~~

~~NOTE The CRC32 checksum algorithm is an option for computing the mandatory File Checksum, not the optional PDU CRC. The two are not to be confused..~~

4.1.2.6 Checksum type zero (0) shall identify the modular checksum defined below.

4.1.2.7 The modular checksum shall be calculated by the following method (see annex F for an example):

a) it shall initially be set to all ‘zeroes’;

b) it shall be calculated by modulo 2^{32} addition of all 4-octet words, aligned from the start of the file;

c) each 4-octet word shall be constructed by copying some octet of file data, whose offset within the file is an integral multiple of 4 (such as 0, 4, 8, 12, etc.), into the first (high-order) octet of the word, and copying the next three octets of file data into the next three octets of the word;

d) the results of the addition shall be carried into each available octet of the checksum unless the addition overflows the checksum length, in which case carry shall be discarded.

NOTES

1 In order to include in a checksum the content of a file data PDU whose offset is not an integral multiple of 4, it is necessary to align the data properly before adding 4-octet blocks of it to the checksum. Data at offset Q may be aligned by inserting N octets of value ‘zero’ before the first octet of the data, where $N = Q \bmod 4$ (the remainder obtained upon dividing Q by 4).

2 In order to include in a checksum a sequence of M octets (the first of which is at a file offset that is an integral multiple of 4) where M is less than 4, it is necessary to pad the data to length 4 before adding it to the checksum. The

4.1.6.1.1.3 The Metadata PDU shall contain

~~a) an indication of whether or not the file contains records whose boundaries are to be respected when the file is segmented for transmission in file data PDUs;~~

a) the size of the file if known (i.e., for a bounded file);

b) the source and destination names (path names) of the file;

c) optional fault handler overrides, Messages to User, filestore requests, and/or flow label;

d) an indication of whether or not transaction closure is requested;

e) the checksum type number identifying the algorithm used to compute the checksum for this file.

NOTE – Assuring that all relevant metadata for the transaction are contained within a single Metadata PDU is an implementation responsibility.

4.1.6.1.1.4 For transactions that deliver more than just metadata, Copy File initiation also shall cause the sending CFDP entity to retrieve the file from the sending filestore and to transmit it in File Data PDUs.

4.1.6.1.1.5 Each segment of data shall be forwarded in a File Data PDU, which also shall contain the offset, in octets, from the beginning of the file at which the segment is located.

4.1.6.1.1.5.1 A File Data PDU may additionally contain information characterizing the manner in which the PDU's content aligns with the record structure of the file. This information, termed 'record continuation state', shall be included whenever the segmentation control service parameter has requested that record boundaries be respected; when observance of record boundaries has not been requested, record continuation state information may optionally be included but the record continuation state shall always be 00. When record continuation state is included in a File Data PDU, the Segment Metadata flag in the PDU header shall be set to 1, the optional segment administrative octet shall be present, and the first (high-order) 2 bits of the administrative octet shall indicate the record continuation state as defined in 5.3 below.

4.1.6.1.1.5.2 In addition, the File Data PDU may contain *N* octets of segment metadata, where *N* is in the range 0 to 63.

NOTE – The decision on whether or not to include segment metadata in a File Data PDU and, if so, the length and nature of that metadata is a local implementation matter.

4.1.6.1.1.5.3 Whenever segment metadata is included in a File Data PDU, the Segment Metadata flag in the PDU header shall be set to '1', and the optional segment administrative octet shall be present, immediately followed by the number of segment metadata octets indicated by the integer value expressed in the last (low-order) 6 bits of the administrative octet.

4.1.6.1.2.8 At the earliest time at which the transaction's Metadata, all file data (if any), and EOF (No error) PDU have all been received by the receiving entity:

a) a checksum shall be calculated for the delivered file by means of the checksum algorithm identified by the checksum type noted in the Metadata PDU;

b) the calculated and received file checksums shall be compared;

c) if the compared checksums are equal, *file delivery shall be deemed Complete*;

d) a File Checksum Failure fault shall be declared if the compared checksums are not equal.

NOTE – The action taken upon such error need not necessarily entail discarding the delivered file. The default handler for File Checksum Failure faults may be Ignore, causing the discrepancy to be announced to the user in a **`Fault.indication`** but permitting the completion of the Copy File procedure at the receiving entity. This configuration setting might be especially appropriate for transactions conducted in unacknowledged mode.

4.1.6.1.2.9 Upon initial receipt of the EOF (No error) PDU, the file size indicated in the PDU shall be compared to the transaction reception progress and a File Size Error fault declared if the progress exceeds the file size.

4.1.6.1.2.10 The flow label may optionally be used to support prioritization and preemption schemes.

NOTE – The use of the flow label is implementation specific.

4.1.6.2 Optional Copy File Procedures at the Receiving Entity

Receipt of a File Data PDU may optionally cause the receiving CFDP, if it is the transaction's destination, to issue a **`File-Segment-Recv.indication`**. Initial receipt of the EOF PDU for a transaction may optionally cause the receiving CFDP, if it is the transaction's destination, to issue an **`EOF-Recv.indication.`**

4.1.6.3 Unacknowledged Mode Procedures

4.1.6.3.1 Transmission Paths

Unacknowledged mode procedures may be exercised over simplex transmission paths unless the transaction closure option is invoked, in which case duplex transmission paths are required.

4.1.6.6.2 Cancel Response Procedures at the Sending Entity

4.1.6.6.2.1 Receipt of a Finished (cancel) PDU shall cause the sending CFDP entity to issue a Notice of Completion (Canceled).

4.1.6.6.2.2 If an acknowledged mode is in effect, Positive Acknowledgement procedures shall be applied to the Finished (cancel) PDU with the Expected Response being an ACK (Finished) PDU.

4.1.6.7 Resume Procedures

4.1.6.7.1 General

Resume procedures apply upon receipt of a **`Resume.request`** primitive submitted by the CFDP user. However:

a) a **`Resume.request`** primitive shall be ignored if it pertains to a transaction that is not currently suspended;

b) if the transaction to which a **`Resume.request`** primitive pertains is currently not only suspended but also frozen (as defined in 4.1.12), then the transaction shall be considered no longer suspended but the only applicable procedure shall be the issuance of a **`Resumed.indication`**.

NOTE – Transaction resumption at the sending entity may result in unsatisfactory communication behavior if the transaction is not concurrently resumed at the receiving entity for the same transaction, and vice versa. To this end, it is recommended that a Remote Resume request message (described later) be transmitted to the peer entity whenever a transaction is locally resumed.

4.1.6.7.2 Resume Procedures at the Sending Entity

4.1.6.7.2.1 On receipt of a **`Resume.request`** primitive, the sending CFDP entity shall

a) resume transmission of Metadata PDU, file segments, and EOF PDU;

b) issue a **`Resumed.indication`**.

4.1.6.7.2.2 If operating in acknowledged mode,

a) any suspended transmission of Prompt PDUs shall be resumed;

b) the application of Positive Acknowledgment Procedures to PDUs previously issued by this entity shall be resumed.

4.1.6.7.3 Resume Procedures at the Receiving Entity

4.1.6.7.3.1 On receipt of a **`Resume.request`** primitive, the receiving CFDP entity shall

a) resume transmission of NAK PDUs;

b) resume any suspended transmission of Keep Alive PDUs;

c) issue a **`Resumed.indication`**.

4.1.6.7.3.2 The application of Positive Acknowledgment Procedures to PDUs previously issued by this entity shall be resumed.

4.1.6.8 Report Procedures

Upon receipt of a **`Report.request`** primitive during a Copy File procedure, the CFDP entity shall issue a **`Report.indication`** primitive providing information on the progress of the transaction.

NOTES

1 Since there is no interaction with any other protocol entity, there is no protocol associated with this procedure.

~~2 The **`Report.indication`** primitive may also be issued on an asynchronous basis without the necessity for a **`Report.request`** primitive.~~

4.1.7 POSITIVE ACKNOWLEDGEMENT PROCEDURES

4.1.7.1 Positive Acknowledgement Procedures at PDU Sending End

If Positive Acknowledgement procedures apply to a PDU,

a) upon issuing the PDU the sending CFDP entity shall start a timer and retain the PDU for retransmission as necessary;

b) if the Expected Response is not received before expiry of the timer, the sending CFDP entity shall reissue the original PDU;

c) the sending CFDP entity shall keep a tally of the number of transmission retries;

d) if a preset limit is exceeded, the sending CFDP entity shall declare a Positive ACK Limit Reached fault;

e) receipt of the Expected Response shall cause the sending CFDP to release the retained PDU.

4.1.11.2.3.3 If receiving in unacknowledged mode, and if the transaction's Metadata PDU has been received and the Closure Requested flag is set to '1' in that PDU, then the receiving CFDP entity shall issue a Finished (cancel) PDU indicating the reason for transaction termination: Cancel.request received or the condition code of the fault whose declaration triggered the Notice of Cancellation.

4.1.11.3 Notice of Suspension Procedures

4.1.11.3.1 General

4.1.11.3.1.1 At any time during a Copy File procedure, either the sending CFDP entity or the receiving CFDP entity may issue a Notice of Suspension.

NOTE

1 A Notice of Suspension may be issued in reaction to the declaration of a fault or to receipt of a **`Suspend.request`** primitive submitted by the CFDP user.

2 Transaction suspension at the sending entity may result in unsatisfactory communication behavior if the transaction is not concurrently suspended at the receiving entity for the same transaction, and vice versa. To this end, it is recommended that a Remote Suspend request message (described later) be transmitted to the peer entity whenever a transaction is locally suspended.

4.1.11.3.1.2 However, a **`Notice of Suspension`** shall be ignored if it pertains to a transaction that is already suspended or if it is issued by the receiving CFDP entity for a transaction sent in Unacknowledged mode.

4.1.11.3.1.3 The following lists of the effects of transaction suspension at the sending and receiving entities are exhaustive; no additional effects should be inferred. In particular, received EOF and Finished PDUs shall be acknowledged and processed in accordance with Positive Acknowledgment Procedures and the relevant Copy File procedures regardless of any transaction suspension that may be in effect.

NOTE – It follows from the above that transaction suspension has no effect whatsoever on the disposition of File Data PDUs that are received at the receiving entity. No received PDUs are discarded, and in fact it is possible, and valid, for the entire file to be reassembled and delivered to the user application while the transaction is suspended.

4.1.11.3.2 Notice of Suspension Procedures at the Sending Entity

4.1.11.3.2.1 On Notice of Suspension of the Copy File procedure, the sending CFDP entity shall

a) suspend transmission of Metadata PDU, file segments, and EOF PDU;

b) save the status of the transaction.

5.2.5 METADATA PDU

The contents of the Parameter field for a File Directive having a Code of Metadata PDU shall be as shown in table 5-9.

Table 5-9: Metadata PDU Contents

Parameter	Length (bits)	Values	Comments
Reserved for future use	1		Set to '0'.
Closure requested	1	'0' – Transaction closure not requested '1' – Transaction closure is requested	If transaction is in Acknowledged mode, set to '0' and ignored.
Reserved for future use	~~6~~2		Set to all 'zeroes'.
Checksum type	4	Checksum algorithm identifier as registered in the SANA Checksum Types Registry	Value zero indicates use of the legacy modular checksum.
File size	FSS	Length of File	In octets. Set to all 'zeroes' for a file of unbounded size.
Source file name	LV		When there is no associated file, e.g., messages used for Proxy operations, the LV Length field indicates zero length and the LV value field is omitted.
Destination file name	LV		When there is no associated file, e.g., messages used for Proxy operations, the LV Length field indicates zero length and the LV value field is omitted.

6.2 PROXY OPERATION

NOTE – The term 'proxy operation' refers to the use of CFDP services by some CFDP user, referred to as the 'originator' of the operation, to initiate the transmission of a file by some remote CFDP entity's user, referred to as the 'respondent', to some other entity's user, referred to as the 'beneficiary'. The beneficiary of a proxy operation might be either the originator itself (in which case the proxy operation functions as a 'Get') or some third CFDP entity's user. Note that, while the core CFDP file transmission service may be invoked by the Proxy mechanism, the Store and Forward Overlay file transmission service (see Appendix A) may not.

6.2.1 GENERAL

To enable interoperability, the following mandatory behavior shall be observed by CFDP users that are in compliance with the CFDP proxy operations specification.

6.2.2 PROXY OPERATIONS MESSAGE TYPES

The message type field for each Reserved CFDP Message used in Proxy operations shall contain one of the values specified in table 6-3.

Table 6-3: Proxy Operations Message Types

Message Type (hexadecimal)	Interpretation
00	Proxy Put Request
01	Proxy Message to User
02	Proxy Filestore Request
03	Proxy Fault Handler Override
04	Proxy Transmission Mode
05	Proxy Flow Label
06	Proxy Segmentation Control
07	Proxy Put Response
08	Proxy Filestore Response
09	Proxy Put Cancel
~~0B~~0A	Proxy Closure Request

8.3 REMOTE ENTITY CONFIGURATION INFORMATION

For each item of remote entity configuration information, a single value shall apply for each remote entity with which the local CFDP entity may be in direct communication.

Table 8-2: Remote Entity Configuration Information

Item	Comment
Remote entity ID	
Protocol version number	CFDP protocol version implemented at this entity.
UT address	UT Address to use when transmitting to this entity.
Positive ACK timer interval	Expressed as a time interval, or N/A.
NAK timer interval	Expressed as a time interval, or N/A.
Keep Alive interval	Expressed as a time interval, or N/A.
Immediate NAK mode enabled	True or false.
Default transmission mode	Acknowledged or unacknowledged.
Transaction closure requested	True or false
Check limit	For use in determining imputed end of transaction.
~~Type~~ Default type of checksum to calculate for all file transmission to this remote entity	~~0 = modular checksum, 1 = CRC32~~As defined in the SANA Checksum Types registry.
~~Type of checksum to calculate for all file reception from this remote entity~~	~~0 = modular checksum, 1 = CRC32~~
Disposition of incomplete received file on transaction cancellation	Discard or retain.
CRCs required on transmission	True or false.
Maximum file segment length	In octets.
Keep Alive discrepancy limit	Expressed as a number of octets, or N/A.
Positive ACK timer expiration limit	Number of expirations.
NAK timer expiration limit	Number of expirations.
Transaction inactivity limit	A time limit.
Start of transmission opportunity	A signal produced by the operating environment.
End of transmission opportunity	A signal produced by the operating environment.
Start of reception opportunity	A signal produced by the operating environment.
End of reception opportunity	A signal produced by the operating environment.

A4.2.4 Management Information Base Remote Entity Configuration Parameters

Item	Protocol Feature	Reference	Status	Parameter Value
RMIB-1R-01	Remote entity ID	8.3	M	
RMIB-1R-02	Protocol version number	8.3	M	
RMIB-1R-03	UT address	8.3	M	
RMIB-1R-04	Default transmission mode	8.3	M	
RMIB-1R-05	Transaction closure requested	8.3	M	
RMIB-1R-06	Check limit	8.3	M	
~~RMIB-1R-07~~	~~Type of checksum to calculate for all file reception from this remote entity~~	~~8.3~~	~~M~~	
RMIB-1R-08	Disposition of incomplete received file on transaction cancellation	8.3	M	
RMIB-1R-09	CRCs required on transmission	8.3	M	
RMIB-1R-10	Transaction inactivity limit	8.3	M	
RMIB-1R-11	Start of transmission opportunity	8.3	M	
RMIB-1R-12	End of transmission opportunity	8.3	M	
RMIB-1R-13	Start of reception opportunity	8.3	M	
RMIB-1R-14	End of reception opportunity	8.3	M	

A4.3 SERVICE CLASS 2 – SENDER

A4.3.1 CFDP Protocol Data Units

Type ID	Protocol Feature	Reference	Status	Support
CFDP-2S-01	General	5.1, tables 5-1, 5-2, 5-3 5.2.1, tables 5-4, 5-5	M	
CFDP-2S-02	End-of-file	5.2.2, table 5-6	M	
CFDP-2S-03	Finished	5.2.3, table 5-7 5.4, tables 5-17 and 5-18	M	
CFDP-2S-04	Ack	5.2.4, table 5-8	M	
CFDP-2S-05	Metadata	5.2.5, table 5-9 5.4, tables 5-15, 5-16, 5-19	M	
CFDP-2S-06	NAK	5.2.6, tables 5-10 and 5-11	M	
CFDP-2S-07	Prompt	5.2.7, table 5-12	M	
CFDP-2S-08	Keep-alive	5.2.8, table 5-13	M	
CFDP-2S-09	File data	5.3, table 5-14	M	

A4.4.4 Management Information Base Remote Entity Configuration Parameters

Item	Protocol Feature	Reference	Status	Parameter Value
RMIB-2R-01	Remote entity ID	8.3	M	
RMIB-2R-02	Protocol version number	8.3	M	
RMIB-2R-03	UT address	8.3	M	
RMIB-2R-04	Positive ACK timer interval	8.3	M	
RMIB-2R-05	NAK timer interval	8.3	M	
RMIB-2R-06	Keep Alive interval	8.3	M	
RMIB-2R-07	Immediate NAK mode enabled	8.3	M	
~~RMIB-2R-08~~	~~Type of checksum to calculate for all file reception from this remote entity~~	~~8.3~~	~~M~~	
RMIB-2R-09	CRCs required on transmission	8.3	M	
RMIB-2R-10	Positive ACK timer expiration limit	8.3	M	
RMIB-2R-11	NAK timer expiration limit	8.3	M	
RMIB-2R-12	Transaction inactivity limit	8.3	M	
RMIB-2R-13	Start of transmission opportunity	8.3	M	
RMIB-2R-14	End of transmission opportunity	8.3	M	
RMIB-2R-15	Start of reception opportunity	8.3	M	
RMIB-2R-16	End of reception opportunity	8.3	M	

A4.5 STORE AND FORWARD OVERLAY – SENDER

A4.5.1 SFO Messages

Type ID	Protocol Feature	Reference	Status	Support
SFOS-01	General	B2.1, table B-1	M	
SFOS-02	Request message	B2.4.2, table B-2	M	
SFOS-03	Message to User message	B2.4.3, table B-3	M	
SFOS-04	Filestore request message	B2.4.4, table B-4	M	
SFOS-05	Fault Handler override message	B2.4.5, table B-5	M	
SFOS-06	Flow label message	B2.4.6, table B-6	M	
SFOS-07	Report message	B2.4.6, table B-7	M	
SFOS-08	Filestore response message	B2.4.6, table B-8	M	

A4.5.2 SFO Procedures

Item	Protocol Feature	Reference	Status	Support
SFOS-09	Routing procedures	B2.3	M	
SFOS-10	SFO transmission initiation procedures	B2.4	M	

ANNEX C

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

C1 SECURITY

CFDP was designed prior to the emergence of awareness within CCSDS that communication protocols need to be secured against attack or compromise. Consequently CFDP includes no security mechanisms of any kind.

CFDP is of course subject to the same confidentiality, data integrity, authentication, and availability concerns as any other protocol. Given the absence of security mechanisms built into CFDP itself, the UT-layer protocol stack over which CFDP functions MUST provide whatever services are required in order to ensure the secure operation of the protocol as deployed. The details of these services are necessarily opaque to CFDP and are beyond the scope of this Recommendation.

C2 SANA

In support of this Recommendation, SANA has created a registry named “CCSDS FILE DELIVERY PROTOCOL (CFDP) ENTITY IDENTIFIER”. This registry lists CFDP entity identifier numbers as discussed in 3.2.4 and 3.2.5 above. It is unmanaged.

SANA is further requested to create a registry of CFDP protocol version numbers.

SANA is further requested to create a registry of checksum types (that is, CFDP checksum calculation algorithm identifiers as discussed in 4.1.2 above).

C3 PATENTS

The CCSDS File Delivery Protocol and its normative references are not protected by any known patents.