

**Draft Recommendation for
Space Data System Standards**

**UNIFIED SPACE
DATA LINK
PROTOCOL**

DRAFT RECOMMENDED STANDARD

CCSDS 732.1-R-1

RED BOOK
August 2016

**Draft Recommendation for
Space Data System Standards**

**UNIFIED SPACE
DATA LINK
PROTOCOL**

DRAFT RECOMMENDED STANDARD

CCSDS 732.1-R-1

RED BOOK
August 2016

AUTHORITY

Issue:	Red Book, Issue
Date:	August 2016
Location:	Not Applicable

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF AUTHORITY:)

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. 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), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

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

STATEMENT OF INTENT

(WHEN THIS RECOMMENDED STANDARD IS FINALIZED, IT WILL CONTAIN THE FOLLOWING STATEMENT OF INTENT:)

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

- o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.
- o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:
 - The **standard** itself.
 - The anticipated date of initial operational capability.
 - The anticipated duration of operational service.
- o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

This document is a technical Recommended Standard for use in developing flight and ground systems for space missions and has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The Unified Space Data Link Protocol described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard 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 e-mail 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 Federal Science Policy Office (BFSP0)/Belgium.
- Central Research Institute of Machine Building (TsNIMash)/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.
- Chinese 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.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- 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.
- 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 draft CCSDS Recommended Standard. Its 'Red Book' status indicates that the CCSDS believes the document to be technically mature and has released it for formal review by appropriate technical organizations. As such, its technical contents are not stable, and several iterations of it may occur in response to comments received during the review process.

Implementers are cautioned **not** to fabricate any final equipment in accordance with this document's technical content.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 732.1-R-1	Unified Space Data Link Protocol, Draft Recommended Standard, Issue	August 2016	Current draft

CONTENTS

<u>Section</u>	<u>Page</u>
1 INTRODUCTION.....	1-1
1.1 PURPOSE.....	1-1
1.2 SCOPE.....	1-1
1.3 APPLICABILITY.....	1-1
1.4 RATIONALE.....	1-2
1.5 DOCUMENT STRUCTURE	1-2
1.6 CONVENTIONS AND DEFINITIONS.....	1-3
1.7 REFERENCES	1-7
2 OVERVIEW	2-1
2.1 CONCEPT OF UNIFIED SPACE DATA LINK PROTOCOL	2-1
2.2 OVERVIEW OF SERVICES	2-6
2.3 OVERVIEW OF FUNCTIONS.....	2-14
2.4 SERVICES ASSUMED FROM LOWER LAYERS	2-17
3 SERVICE DEFINITION.....	3-1
3.1 OVERVIEW	3-1
3.2 SOURCE DATA.....	3-1
3.3 MAP PACKET SERVICE.....	3-4
3.4 MAP ACCESS SERVICE.....	3-6
3.5 MAP OCTET STREAM SERVICE	3-14
3.6 MASTER CHANNEL OPERATIONAL CONTROL FIELD (MC_OCF) SERVICE	3-18
3.7 VIRTUAL CHANNEL FRAME (VCF) SERVICE.....	3-22
3.8 MASTER CHANNEL FRAME (MCF) SERVICE.....	3-25
3.9 INSERT SERVICE.....	3-28
3.10 COPS MANAGEMENT SERVICE.....	3-33
4 PROTOCOL SPECIFICATION WITHOUT SDLS OPTION	4-1
4.1 PROTOCOL DATA UNIT.....	4-1
4.2 PROTOCOL PROCEDURES AT THE SENDING END.....	4-18
4.3 PROTOCOL PROCEDURES AT THE RECEIVING END.....	4-32
5 MANAGED PARAMETERS WITHOUT SDLS OPTION.....	5-1
5.1 OVERVIEW	5-1
5.2 MANAGED PARAMETERS FOR A PHYSICAL CHANNEL	5-2

CONTENTS (continued)

<u>Section</u>	<u>Page</u>
5.3 MANAGED PARAMETERS FOR A MASTER CHANNEL.....	5-3
5.4 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL	5-4
5.5 MANAGED PARAMETERS FOR A MAP CHANNEL	5-6
5.6 MANAGED PARAMETERS FOR PACKET TRANSFER.....	5-6
6 PROTOCOL SPECIFICATION WITH SDLS OPTION.....	6-1
6.1 OVERVIEW	6-1
6.2 USE OF SDLS PROTOCOL	6-1
6.3 USLP TRANSFER FRAME WITH SDLS	6-1
6.4 SENDING END PROTOCOL PROCEDURES WITH SDLS.....	6-4
6.5 RECEIVING END PROTOCOL PROCEDURES WITH SDLS.....	6-7
6.6 MANAGED PARAMETERS WITH SDLS	6-9
ANNEX A IMPLEMENTATION CONFORMANCE STATEMENT (ICS)	
PROFORMA (NORMATIVE).....	A-1
ANNEX B FRAME ERROR CONTROL FIELD CODING PROCEDURES	
(NORMATIVE)	B-1
ANNEX C SECURITY, SANA, AND PATENT CONSIDERATIONS	
(INFORMATIVE)	C-1
ANNEX D INFORMATIVE REFERENCES (INFORMATIVE)	D-1
ANNEX E PROXIMITY-1 VARIABLE-LENGTH SUPERVISORY	
PROTOCOL DATA FIELD FORMATS (INFORMATIVE)	E-1
ANNEX F PROTOCOL DATA UNIT (CONTROL WORDS/DIRECTIVES)	
FORMATS (INFORMATIVE)	F-1
ANNEX G ABBREVIATIONS AND ACRONYMS (INFORMATIVE).....	G-1

Figure

1-1 Bit Numbering Convention.....	1-6
2-1 Relationship with OSI Layers.....	2-1
2-2 Relationships between Channels	2-5
2-3 Asynchronous Service Model.....	2-9
2-4 Synchronous Service Model	2-10
2-5 Internal Organization of Protocol Entity (Sending End)	2-15
2-6 Internal Organization of Protocol Entity (Receiving End).....	2-16
2-7 Unified Space Data Link Protocol Channel Tree	2-17
4-1 USLP Transfer Frame Structural Components.....	4-1
4-2 Transfer Frame Primary Header	4-2
4-3 Transfer Frame Data Field.....	4-9

CONTENTS (continued)

<u>Figure</u>	<u>Page</u>
4-4 Transfer Frame Data Field Header	4-10
4-5 Internal Organization of Protocol Entity (Sending End)	4-18
4-6 Abstract Model of Packet Processing Function for Fixed TFDFs.....	4-20
4-7 Abstract Model of MAP Packet Processing Function for a Variable-Length TFDF .	4-21
4-8 Abstract Model of MAP Generation Function for Fixed-Length TFDF	4-23
4-9 Abstract Model of MAP Generation Function for Variable-Length TFDF	4-24
4-10 Abstract Model of the MAP Octet Stream Processing Function	4-25
4-11 Abstract Model of MAP Multiplexing Function	4-26
4-12 Abstract Model of Virtual Channel Generation Function	4-27
4-13 Abstract Model of Virtual Channel Multiplexing Function	4-29
4-14 Abstract Model of Master Channel Multiplexing Function	4-30
4-15 Abstract Model of All Frames Generation Function	4-31
4-16 Internal Organization of Protocol Entity (Receiving End).....	4-32
4-17 Abstract Model of MAP Packet Extraction Function for Fixed-Length TFDFs	4-34
4-18 Abstract Model of MAP Packet Extraction Function for Variable-Length TFDFs ...	4-35
4-19 Abstract Model of MAP Reception Function for Fixed-Length TFDFs	4-36
4-20 Abstract Model of MAP Reception Function for Variable-Length TFDFs	4-37
4-21 Abstract Model of MAP Octet Stream Extraction Function	4-38
4-22 Abstract Model of MAP Demultiplexing Function	4-39
4-23 Abstract Model of Virtual Channel Reception Function.....	4-40
4-24 Abstract Model of Virtual Channel Demultiplexing Function.....	4-41
4-25 Abstract Model of Master Channel Demultiplexing Function	4-42
4-26 Abstract Model of All Frames Reception Function.....	4-43
6-1 Frame without SDLS Compared to Frame with SDLS	6-2
B-1 Logic Diagram of the CRC-16 Encoder	B-2
B-2 Logic Diagram of the CRC-16 Decoder	B-3
B-3 A Possible Implementation of the CRC-32 Encoder	B-4
B-4 A Possible Implementation of the CRC-32 Decoder.....	B-5
E-1 Type 1 SPDU Data Field Contents.....	E-2
E-2 SET TRANSMITTER PARAMETERS Directive	E-3
E-3 SET CONTROL PARAMETERS Directive	E-6
E-4 SET RECEIVER PARAMETERS Directive	E-8
E-5 SET V(R) Directive	E-12
E-6 Report Request.....	E-13
E-7 SET PL EXTENSIONS	E-15
E-8 Report Source Spacecraft ID	E-20
E-9 Type 2 SPDU Data Field Contents.....	E-21
F-1 Communications Link Control Word	F-2
F-2 Proximity Link Control Word Fields.....	F-9

CONTENTS (continued)

<u>Table</u>	<u>Page</u>
2-1 Summary of Services Provided by USLP Space Data Link Protocol.....	2-11
4-1 Interpretation of the Virtual Channel Frame Count Length	4-7
4-2 Summary of the TFDZ Construction Rules	4-12
5-1 Managed Parameters for a Physical Channel	5-2
5-2 Managed Parameters for a Master Channel	5-3
5-3 Managed Parameters for a Virtual Channel	5-4
5-4 Managed Parameters for a MAP Channel	5-6
5-5 Managed Parameters for Packet Transfer	5-6
6-1 Additional Managed Parameters for a Virtual Channel when the Unified Space Data Link Protocol Supports SDLS	6-9
F-1 Fixed-Length Supervisory Protocol Data Unit	F-8
F-2 Variable-Length Supervisory Protocol Data Unit	F-11

1 INTRODUCTION

1.1 PURPOSE

The purpose of this Recommended Standard is to specify the Unified Space Data Link Protocol (USLP). This protocol is a Data Link Layer protocol (see reference [1]) to be used over space-to-ground, ground-to-space, or space-to-space communications links by space missions.

1.2 SCOPE

This Recommended Standard defines the USLP in terms of:

- a) the services provided to the users of this protocol;
- b) the protocol data units employed by the protocol; and
- c) the procedures performed by the protocol.

It does not specify:

- a) individual implementations or products;
- b) the implementation of service interfaces within real systems;
- c) the protocol procedures specified in both the COP-1 reference [9] and the COP-P reference [10];
- d) the security services specified in the SDLS protocol (reference [14]);
- e) the flow control;
- f) the methods or technologies required to perform the procedures; or
- g) the management activities required to configure and control the protocol.

1.3 APPLICABILITY

This Recommended Standard applies to the creation of Agency standards and to future data communications over space links between Consultative Committee for Space Data Systems (CCSDS) Agencies in cross-support situations. The Recommended Standard includes comprehensive specification of the services and protocol for inter-Agency cross support. It is neither a specification of, nor a design for, real systems that may be implemented for existing or future missions.

The Recommended Standard specified in this document is to be invoked through the normal standards programs of each CCSDS Agency and is applicable to those missions for which cross support based on capabilities described in this Recommended Standard is anticipated. Where mandatory capabilities are clearly indicated in sections of the Recommended

Standard, they must be implemented when this document is used as a basis for cross support. Where options are allowed or implied, implementation of these options is subject to specific bilateral cross support agreements between the Agencies involved.

1.4 RATIONALE

The CCSDS believes it is important to document the rationale underlying the recommendations chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The USLP Green Book (reference [D11]) contains further details including the applicability to various space links and data rates.

1.5 DOCUMENT STRUCTURE

This document is divided into six numbered sections and seven annexes:

- a) section 1 presents the purpose, scope, applicability, and rationale of this Recommended Standard and lists the conventions, definitions, and references used throughout the Recommended Standard;
- b) section 2 provides an overview of USLP;
- c) section 3 defines the services provided by the protocol entity;
- d) section 4 specifies the protocol data units and procedures employed by the protocol entity;
- e) section 5 specifies the managed parameters used by the protocol entity;
- f) section 6 specifies the protocol entity with support for the Space Data Link Security Protocol;
- g) annex A provides the Protocol Implementation Conformance Statement (PICS) proforma;
- h) annex B specifies Frame Error Control Field procedures;
- i) annex C provides the security, SANA, and patent considerations;
- j) annex D provides a list of informative references;
- k) annex E lists the Proximity-1 variable-length Supervisory Protocol Data Unit (SPDU) formats;
- l) annex F discusses Protocol Data Unit control words and directives;
- m) annex G lists all acronyms used within this document.

1.6 CONVENTIONS AND DEFINITIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from the Open Systems Interconnection (OSI) Basic Reference Model

This Recommended Standard makes use of a number of terms defined in reference [1]. The use of those terms in this Recommended Standard is to be understood in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- a) blocking;
- b) connection;
- c) Data Link Layer;
- d) entity;
- e) flow control;
- f) Network Layer;
- g) peer entities;
- h) Physical Layer;
- i) protocol control information;
- j) protocol data unit;
- k) real system;
- l) segmenting;
- m) service;
- n) Service Access Point (SAP);
- o) SAP address;
- p) service data unit.

1.6.1.2 Definitions from OSI Service Definition Conventions

This Recommended Standard makes use of a number of terms defined in reference [2]. The use of those terms in this Recommended Standard is to be understood in a generic sense, i.e., in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

- a) confirmation;

- b) indication;
- c) primitive;
- d) request;
- e) response;
- f) service provider;
- g) service user.

1.6.1.3 Terms Defined in this Recommended Standard

For the purposes of this Recommended Standard, the following definitions also apply. Many other terms that pertain to specific items are defined in the appropriate sections.

aperiodic: Not *periodic* (see **periodic**).

asynchronous: Not *synchronous* (see **synchronous**).

delimited: Having a known (and finite) length; applies to data in the context of data handling.

Isochronous: Characterized by occurring at equal intervals of time.

Mission Phase: A period of a mission during which specified communications characteristics are fixed. The transition between two consecutive Mission Phases may cause an interruption of the communications services.

periodic: Of or pertaining to a sequence of events in which each event occurs at a fixed time interval (within specified tolerance) after the previous event in the sequence.

Physical Channel: A stream of bits transferred over a space link in a single direction.

space link: A communications link between a spacecraft and its associated ground system or between two spacecraft. A space link consists of one or more Physical Channels in one or both directions.

synchronous: Of or pertaining to a sequence of events occurring in a fixed time relationship (within specified tolerance) to another sequence of events. It should be noted that ‘synchronous’ does not necessarily imply ‘periodic’ or ‘constant rate’.

(USLP) Transfer Frame: The protocol data unit of the Unified Space Data Link (USLP) Protocol.

1.6.1.4 Terms Defined in Other Recommended Standards

forward link: That portion of a space link in which the caller transmits and the responder receives (typically a command link).

port: Identifier of the logical or physical port that is the destination for a user's service data unit.

return link: That portion of a space link in which the responder transmits and the caller receives (typically a telemetry link).

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

The following conventions apply for the normative specifications in this Recommended Standard:

- 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.

1.6.2.2 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.

1.6.3 CONVENTIONS

In this document, the following convention is used to identify each bit in an N -bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'Bit 0'; the following bit is defined to be 'Bit 1' and so on up to 'Bit $N-1$ '.

When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., 'Bit 0' (see figure 1-1).

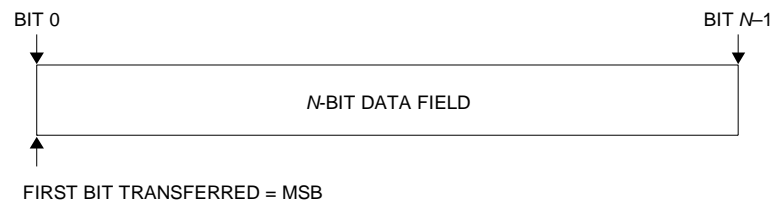


Figure 1-1: Bit Numbering Convention

In accordance with standard data-communications practice, data fields are often grouped into eight-bit 'words' which conform to the above convention. Throughout this Recommended Standard, such an eight-bit word is called an 'octet'. The numbering for octets within a data structure starts with zero. By CCSDS convention, all 'spare' bits shall be permanently set to '0'.

1.7 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.

- [1] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. 2nd ed. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.
- [2] *Information Technology—Open Systems Interconnection—Basic Reference Model—Conventions for the Definition of OSI Services*. International Standard, ISO/IEC 10731:1994. Geneva: ISO, 1994.
- [3] *TM Synchronization and Channel Coding*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.0-B-2. Washington, D.C.: CCSDS, August 2011.
- [4] *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.2-B-1. Washington, D.C.: CCSDS, March 2012.
- [5] *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.3-B-1. Washington, D.C.: CCSDS, March 2013.
- [6] *TC Synchronization and Channel Coding*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-2. Washington, D.C.: CCSDS, September 2010.
- [7] *Proximity-1 Space Link Protocol—Coding and Synchronization Sublayer*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.2-B-2. Washington, D.C.: CCSDS, December 2013.
- [8] “Registries.” Space Assigned Numbers Authority. <http://sanaregistry.org/r/>.
- [9] *Communications Operation Procedure-1*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.1-B-2. Washington, D.C.: CCSDS, September 2010.
- [10] *Proximity-1 Space Link Protocol—Data Link Layer*. Issue 5. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.0-B-5. Washington, D.C.: CCSDS, December 2013.

- [11] *CCSDS Global Spacecraft Identification Field Code Assignment Control Procedures*. Issue 6. Recommendation for Space Data System Standards (Blue Book), CCSDS 320.0-B-6. Washington, D.C.: CCSDS, October 2013.
- [12] *Space Packet Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-1. Washington, D.C.: CCSDS, September 2003.
- [13] *Encapsulation Service*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.1-B-2. Washington, D.C.: CCSDS, October 2009.
- [14] *Space Data Link Security Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.0-B-1. Washington, D.C.: CCSDS, September 2015.
- [15] *Proximity-1 Space Link Protocol—Physical Layer*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.1-B-4. Washington, D.C.: CCSDS, December 2013.
- [16] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.
- [17] *Space Data Link Security Protocol—Extended Procedures*. Recommendation for Space Data System Standards, CCSDS 355.1. Forthcoming.

NOTE — Informative references are listed in annex D.

2 OVERVIEW

2.1 CONCEPT OF UNIFIED SPACE DATA LINK PROTOCOL

2.1.1 ARCHITECTURE

The Unified Space Data Link Protocol is a Data Link Layer protocol (see reference [1]) to be used by space missions. This protocol has been designed to meet the requirements of space missions for efficient transfer of space application data of various types and characteristics over space-to-ground, ground-to-space, or space-to-space communications links (hereafter called space links).

Figure 2-1 illustrates the relationship of this protocol to the reference model of Open Systems Interconnection (reference [1]). Two sublayers of the Data Link Layer are defined for CCSDS space link protocols as shown in reference [D2]. USLP corresponds to the Data Link Protocol Sublayer, and provides functions of transferring various data using a variable-length (or fixed length as a subset) protocol data unit called the Transfer Frame. The Synchronization and Channel Coding Sublayer provides some additional functions necessary for transferring Transfer Frames over a space link. These functions are delimiting/synchronizing Transfer Frames, error-correction coding/decoding, and bit transition generation/removal (optional). Depending on the coding scheme applied some of the above mentioned functions are optional or performed differently as explained later in this document. For the Synchronization and Channel Coding Sublayer, the following set of Synchronization and Channel Coding Recommended Standards (references [3] through [7]) are to be used with USLP with constraints listed later in this document. How USLP is used in overall space data systems is shown in references [D2] through [D4] and [D11].

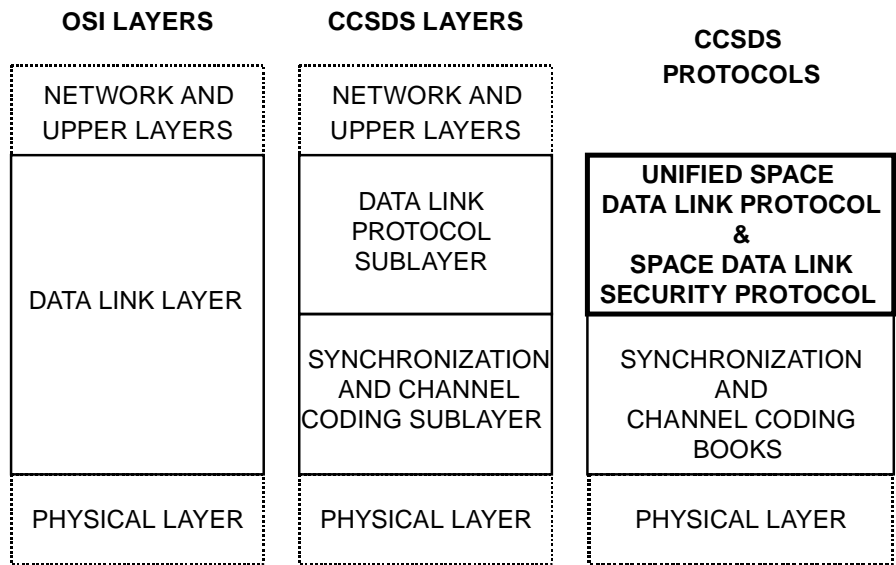


Figure 2-1: Relationship with OSI Layers

2.1.2 PROTOCOL FEATURES

2.1.2.1 Transfer Frames, Virtual Channels, and Multiplexer Access Points

USLP provides the users with several services to transfer service data units over a space link. These protocol data units are known as USLP Transfer Frames (unless otherwise stated, the terms ‘Transfer Frame’ and ‘Frame’ in this document refer to the USLP Transfer Frame). Each Transfer Frame contains a primary header which provides protocol control information, identifying the length of the frame and signaling the inclusion of selected fields. The Transfer Frames carry upper-layer service data units within the Transfer Frame Data Field (TFDF). The TFDF contains a header that identifies both how the data field is organized and identifies the protocol to which the service data unit(s) are associated.

A key feature of USLP is the concept of ‘Virtual Channels’ (VC). The Virtual Channel facility allows one Physical Channel to be shared among multiple higher-layer data streams, each of which may have different service requirements. Of particular importance is the Quality of Service (QoS) attribute associated with a VC. A single Physical Channel may therefore be divided into several separate logical data channels, each known as a ‘Virtual Channel’. Each Transfer Frame transferred over a Physical Channel belongs to one of the Virtual Channels of the Physical Channel.

Moreover, this protocol enables service data units from different sources to be multiplexed together in one Virtual Channel using ‘Multiplexer Access Points’ (MAPs). MAP_ID assignments allow service data units arriving at a Service Access Point (SAP) at the sending end to be transferred to a SAP with the corresponding MAP_ID at the receiving end. Each SAP can be associated with a specific protocol. USLP enables the transfer of CCSDS Packets, SDUs associated with protocols registered in SANA, or user defined data. The set of recognized protocol IDs is a SANA registry (reference [8]).

2.1.2.2 Additional USLP Features

USLP has increased the allowable size of the Transfer Frame in order to reduce the operational frame handling process for high rate missions. The protocol also has increased the capability for identifying (using a larger addressing space) more spacecraft. Moreover, it provides the means to transfer service data units from CCSDS recognized protocols within the frame without the need for additional encapsulation services. USLP provides a configurable sized sequence counter in the frame primary header that is incremented to allow the receiving entity to determine if any frames are missing.

2.1.2.3 Efficient Data Transfer

USLP provides the users with several services to transfer service data units over a space link. USLP performs (1) segmentation and blocking (i.e., aggregation) of service data units and (2) transmission control of service data units.

Because the underlying space link inherently includes a noisy signal path, there is a finite probability that it will introduce an error. It may be desirable to break large service data units into relatively small pieces when the frame error rate for large frames is higher than required by the mission. Under those conditions each piece has a lower probability of being invalidated by transmission error than if the entire service data unit were sent contiguously. System throughput efficiency may be improved because only small pieces have to be retransmitted when errors are detected. However, there may also be situations in which the service data units are very small. For efficient transfer of service data units, it is desirable to group these small units into larger pieces. USLP provides the capability to break large service data units into relatively small pieces (segmentation) and to group small service data units into larger pieces (blocking).

The Communications Operations Procedure (COP) as described in 2.1.2.4 below controls the transmission of service data units through the space link performing retransmissions needed to ensure delivery of service data units in sequence and without gaps or duplication.

In addition, a systematic repeated frame retransmission mechanism for use on deep space links may optionally be provided by the Synchronization and Channel Coding Sublayer when reference [6] is applied (see 2.4.2).

2.1.2.4 Communications Operation Procedure (COP) (optional)

Within this document the term, COP refers to both the Communications Operation Procedure-1 (COP-1) reference [9], used in near Earth and deep space applications, as well as the Communication Operations Procedure for Proximity links (COP-P) within the Proximity-1 Space Data Link Protocol reference [10]. The USLP Green Book (reference [D11]) provides further details.

The use of either the COP-1 or COP-P procedures is optional; both are compatible with USLP. The Protocol Data Units (CLCW for COP-1 and PLCW for COP-P) and operational procedures for COP-1 and COP-P, respectively, are not identical but similar and are transparent to USLP. For this reason the generic terms Frame Acceptance and Reporting Mechanism (FARM) and Frame Operation Procedure (FOP) are used.

CAUTION – The controlling specifications for the logical operations which must be executed to perform both the COP-1 and COP-P are contained in more detailed CCSDS Recommended Standards (references [9] and [10]). In the event of any conflict between the descriptive text contained in this Recommended Standard and the text of reference [9] or [10], the more detailed specifications contained in reference [9] or [10] are normative.

The COP fully specifies the closed-loop procedures executed by the sending and receiving ends of USLP. The COP consists of a pair of synchronized procedures for each Virtual Channel: a FOP that executes within the sending entity; and a FARM that executes within the receiving entity. The sending FOP transmits Transfer Frames to the receiving FARM.

The FARM returns to the FOP reports of the status of Transfer Frame acceptance using protocol control reports and thus closes the loop.

The inclusion of the COP provides a reliable Quality of Service (QoS), i.e., the delivery of service data units to the layer above at the receiving end, correct and without omission or duplication, and in the same sequential order in which they were received from the layer above at the sending end.

Correctness of the delivered service data units is guaranteed (within known error probabilities) by the Frame Error Control Field applied by the Data Link Protocol Sublayer, and by the Frame Validation Checks performed in USLP. However, validation of the completeness, sequentially, and non-duplication of the delivered service data units on a particular Virtual Channel requires that an accounting (i.e., numbering) scheme for Transfer Frames be implemented by the COP.

The COP controls transfer of Sequence-Controlled Transfer Frames so that service data units within Sequence-Controlled Transfer Frames are delivered to the receiving end of the layer above, correct and without omission or duplication, and in the same sequential order in which they were received from the layer above at the sending end.

Expedited Transfer Frames are processed by the COP only to the extent of causing the FARM to increment a counter for Expedited Transfer Frames. Expedited Transfer Frames are also used to send Control Commands from the FOP to the FARM.

2.1.2.5 Space Data Link Security Protocol (optional)

The Space Data Link Security (SDLS) Protocol is specified in reference [14]. The SDLS protocol may provide security, such as authentication and encryption, for USLP Transfer Frames. Support for the SDLS protocol is an optional feature of USLP.

NOTE – The inclusion of the SDLS protocol makes no changes to any requirements in this Recommended Standard except for the inclusion of a mandatory Security Header and optional Security Trailer.

The inclusion of security provided by the SDLS protocol is selectable by Virtual Channel. So, for example, there can be some Virtual Channels with security and some without. The type of security can vary from one Virtual Channel to another.

2.1.3 ADDRESSING

There are four identifier fields in the header of Transfer Frames: Transfer Frame Version Number (TFVN), Spacecraft Identifier (SCID), Virtual Channel Identifier (VCID) and the Multiplexer Access Point Identifier (MAP ID). The concatenation of a TFVN and a SCID is known as a Master Channel Identifier (MCID), and the concatenation of an MCID and a VCID is called a Global Virtual Channel Identifier (GVCID). The concatenation of the GVCID and a MAP ID is called the Global MAP ID (GMAP ID). Therefore

$$\text{MCID} = \text{TFVN} + \text{SCID};$$

$$\text{GVCID} = \text{MCID} + \text{VCID} = \text{TFVN} + \text{SCID} + \text{VCID}.$$

Each Virtual Channel in a Physical Channel is identified by a GVCID. Therefore a Virtual Channel consists of Transfer Frames with the same GVCID.

All Transfer Frames with the same GVCID and MAP ID constitute a MAP Channel. Thus a Virtual Channel consists of one or multiple MAP Channels. The concatenation of a GVCID and a MAP ID is known as a Global MAP ID (GMAP ID). Therefore,

$$\begin{aligned}\text{GMAP ID} &= \text{GVCID} + \text{MAP ID} \\ &= \text{MCID} + \text{VCID} + \text{MAP ID} \\ &= \text{TFVN} + \text{SCID} + \text{VCID} + \text{MAP ID}.\end{aligned}$$

All Transfer Frames with the same MCID on a Physical Channel constitute a Master Channel (MC). A Master Channel consists of one or more Virtual Channels. In most cases, a Physical Channel carries only Transfer Frames of a single MCID, and the Master Channel will be identical with the Physical Channel. However, a Physical Channel may carry Transfer Frames with multiple MCIDs (with the same TFVN). In such a case, the Physical Channel consists of multiple Master Channels. A Physical Channel is identified with a Physical Channel Name, which is set by management and not included in the header of Transfer Frames.

The relationships between these Channels are shown in figure 2-2.

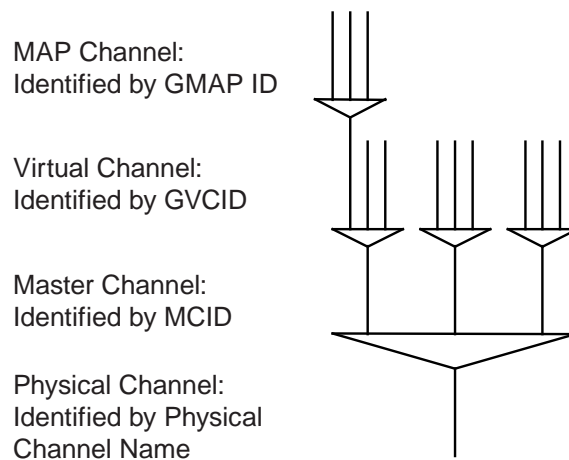


Figure 2-2: Relationships between Channels

2.1.4 PROTOCOL DESCRIPTION

USLP is described in terms of:

- a) the services provided to the users;

- b) the protocol data units; and
- c) the procedures performed by the protocol.

The service definitions are given in the form of primitives, which present an abstract model of the logical exchange of data and control information between the protocol entity and the service user. The definitions of primitives are independent of specific implementation approaches.

The procedure specifications define the procedures performed by protocol entities for the transfer of information between peer entities. The definitions of procedures are independent of specific implementation methods or technologies.

This protocol specification also specifies the requirements for the underlying services provided by the Synchronization and Channel Coding Sublayer.

2.2 OVERVIEW OF SERVICES

2.2.1 COMMON FEATURES OF SERVICES

USLP provides users with data transfer services. The point at which a service is provided by a protocol entity to a user is called a Service Access Point (SAP) (see reference [1]). Each service user is identified by a SAP address that is associated with a specific MAP ID within a specific Virtual Channel i.e., the GMAP ID.

Service data units submitted to a SAP are processed in the order of submission. No processing order is maintained for service data units submitted to different SAPs.

NOTE – Implementations may be required to perform flow control at a SAP between the service user and the service provider. However, CCSDS does not recommend a scheme for flow control between the user and the provider.

The followings are features common to all the services defined by this Recommended Standard:

- a) unidirectional (one way) services: one end of a connection can send, but not receive, data through the space link, while the other end can receive, but not send;
- b) asynchronous services: there are no predefined timing rules for the transfer of service data units supplied by the service user or for the transmission of Transfer Frames generated by the service provider. The user may request data transfer at any time, but there may be restrictions imposed by the service provider on the data generation rate. The timing of data transfer is determined by the provider in accordance with mission-specific rules and may depend on the traffic of the data exchange by service provider/lower layers at the time of transfer.
- c) unconfirmed services: the sending user does not receive confirmation from the receiving end that data has been received;
- d) incomplete services: the services do not guarantee completeness, but some services may signal gaps in the sequence of service data units delivered to the receiving user;

- e) sequence-preserving services: the sequence of service data units supplied by the sending user is preserved through the transfer over the space link, although for the Expedited Service, described below, there may be gaps and duplications in the sequence of service data units delivered to the receiving user.

NOTE – This Recommended Standard assumes that these services are provided at the end points of a space link. However, this Recommended Standard makes no assumptions concerning how these end points are composed or configured either on-board a spacecraft or in a ground system. In a ground system, the services defined by this Recommended Standard may be extended or enhanced with Space Link Extension Services (reference [D5]).

In addition, the following optional features are provided by these external CCSDS standards when selected by the user:

- a) The use of the COP procedures adds the capability for in order delivery of service data units to the layer above without error or omission or duplication at the receiving end.
- b) The use of the SDLS protocol adds the capability to authenticate the sender and/or to encrypt the data to provide security for specific Virtual Channels defined to use the SDLS security services.

2.2.2 SERVICE ATTRIBUTES

2.2.2.1 Overview

USLP in association with the COP-1 (reference [9]), COP-P (reference [10]), and SDLS (reference [14]), provide the following service attributes (sequence-controlled, expedited, authenticated and/or encrypted, asynchronous, synchronous, and periodic) that determine the Quality of Service (QoS) associated with how reliably service data units supplied by the sending user are delivered to the receiving user.

The sequence-controlled QoS is provided by the COP-1 reference [9] or COP-P reference [10] procedures. The user requests with a parameter of the service request primitive whether the Sequence-Controlled or Expedited Service QoS should be applied to each service data unit.

The Space Data Link Security (SDLS) services provide the user with the capability of authenticating the frame originator and/or encrypting the frame contents. SDLS security services are established by the use of the managed parameters in section 6.

The capability to transfer data either in an asynchronous, synchronous, or periodic fashion is determined by the Transfer Frame multiplexing rules, which are specified by the missions.

These service attributes are provided at any Service Access Point except for the Virtual Channel Frame and Master Channel Frame.

For the Virtual Channel Frame and Master Channel Frame Services, the service provider does not make any distinction between Sequence-Controlled and Expedited service QoS applicable to service data units supplied by the user. The user should request necessary procedures to provide Sequence-Controlled and/or Expedited Service QoS for its service data units.

2.2.2.2 Sequence-Controlled Service – Reliable Delivery

The reliable QoS (Sequence-Controlled Service) of either COP-1 (reference [9]) or COP-P (reference [10]) utilizes an Automatic Repeat Queuing (ARQ) procedure of the ‘go-back-*n*’ type with sequence-control mechanisms at both sending and receiving ends and a standard report returned from the receiving end to the sending end.

For Sequence-Controlled Service, service data units supplied by a sending user at a SAP are inserted into the Data Field of Transfer Frames (after MAP multiplexing when applicable) and transmitted on a Virtual Channel in the order in which they are presented at the SAP. The retransmission mechanism ensures with a high probability of success that:

- a) no service data unit is lost;
- b) no service data unit is duplicated;
- c) no service data unit is delivered out of sequence.

2.2.2.3 Expedited Service – Best Effort Delivery

The Best Effort QoS (Expedited Service) is used when ARQ is not required by the Data Link Layer, or when a higher layer protocol provides a retransmission capability.

For Expedited Service, service data units supplied by the sending user are transmitted one or more times (i.e., no retransmission). There is no guarantee that all Expedited service data units are delivered to the receiving user.

NOTE – Although Expedited Service carries the name ‘Expedited’, it is neither a required method nor a faster method for sending urgent data to the receiving end.

2.2.2.4 Security Service

The optional use of the SDLS protocol (reference [14]) provides all its functions (authentication, encryption, authenticated encryption) for the data in the Transfer Frame Data Zone of a USLP Frame. It provides full protection for the service data of the MAP Packet (MAPP) service, the VC Octet Stream Service, and the MAP Access (MAPA) Service (see section 6, Protocol Specification with SDLS option).

SDLS provides authentication for some fields in the Transfer Frame Primary Header and for some auxiliary data fields in a USLP Frame. It does not provide encryption for these fields.

SDLS provides no protection for the protocol control frames generated by either the COP-1 or COP-P.

SDLS provides no protection for the Virtual Channel Operational Control Field (VC_OCF) Service nor for the Insert Service. It also provides no protection for the frames supplied to USLP by external sources such as by the Virtual Channel Frame (VCF) and the Master Channel Frame (MCF) services.

2.2.2.5 Asynchronous Service

In asynchronous service, there are no timing relationships between the transfer of service data units supplied by the service user and the transmission of Transfer Frames generated by the service provider. The user may request data transfer at any time it desires, but there may be restrictions imposed by the provider on the data generation rate. In this service (figure 2-3), each service data unit from a sending user is placed in a queue, the contents of which are sent to a receiving user in the order in which they were presented. Although transmission errors may prevent delivery of some data units, the service provider attempts to transfer all data units provided by the user exactly once. The timing of data transfer is determined by the provider in accordance with mission-specific rules, and may depend on the traffic at the time of transfer. The key feature of this service is that all of the service data units from the sending user are transferred, and transferred only once.

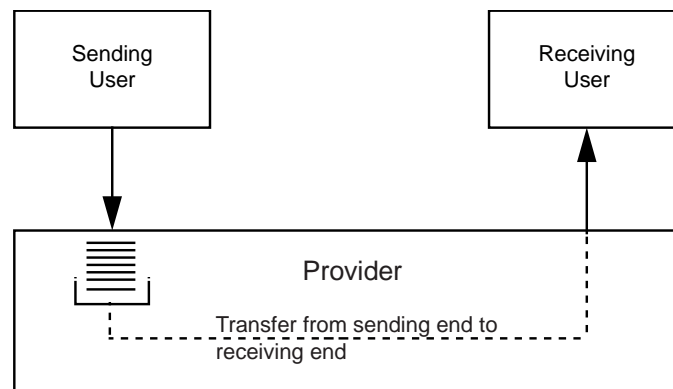


Figure 2-3: Asynchronous Service Model

2.2.2.6 Synchronous Service

In synchronous service, the transfer of service data units is synchronized with the release of either (1) Transfer Frames of a Virtual Channel, (2) Transfer Frames of a Master Channel, or (3) all Transfer Frames of a Physical Channel. The transfer timing may be periodic or aperiodic.

In this service (figure 2-4), each service data unit from a sending user is placed in a buffer that can hold only one service data unit; the content of the buffer is sent to a receiving user at the time when a Transfer Frame is transmitted. The transmission timing of Transfer Frames

is determined by the service provider according to mission-specific rules (usually known to the user). The key feature of this service, which is essentially time-division multiplexing, is that the timing of data transfer is driven by the transfer mechanism, not by individual service requests from the user. Thus a particular service data unit from a user might be sent once, several times (if the 'new' value is not placed in the buffer soon enough), or not at all (if one value is replaced by a second before the service provider can send it).

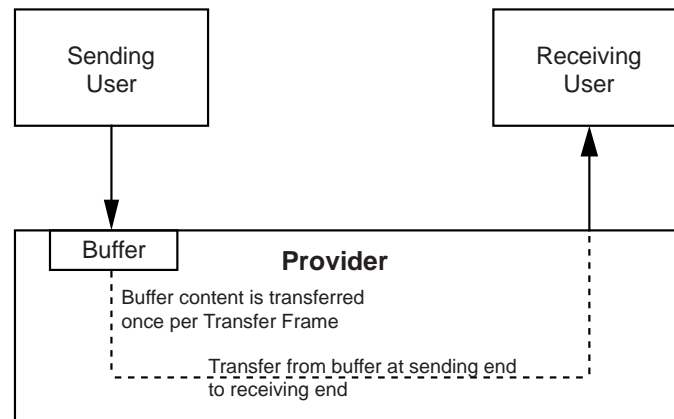


Figure 2-4: Synchronous Service Model

2.2.2.7 Periodic Service

Periodic service is a special case of synchronous service in which service data units are transferred at a constant rate. Periodic transfer from service interface to service interface is provided with a specified maximum delay and a specified maximum jitter at the service interface. There is one case in which synchronous service is periodic: when the service is associated with a Virtual Channel (or a Master Channel) and that Virtual (or Master) Channel produces Transfer Frames at a constant rate.

For periodic services, all service data units are sent only once if the user supplies service data units at the same rate as the rate at which the service provider transfers them.

2.2.3 SUMMARY OF SERVICES

2.2.3.1 Introduction

Eight services are provided by USLP. Three of them (MAP Packet, MAP Access, and MAP Octet Stream) are provided for a MAP Channel. Two of them (Virtual Channel Frame and COPs Management) are provided for a Virtual Channel. Two of them (Master Channel Operational Control Field [OCF] and Master Channel Frame) is provided for a Master Channel. One of them (Insert) is provided for all Transfer Frames on a Physical Channel.

Table 2-1 summarizes these services and shows their characteristics, the Service Data Units (SDUs) that they transfer and the availability of SDLS security features. The optional SDLS protocol can provide security features for the SDUs transferred by some of the services:

- encryption, to provide confidentiality by hiding data content;
- authentication, to confirm the source and integrity of the data.

Table 2-1: Summary of Services Provided by USLP Space Data Link Protocol

Service	Service Type	Service Data Unit	SAP Address	SDLS Security Features
MAP Packet (MAPP)	Sequence-Controlled and Expedited, Asynchronous	Packet	GMAP ID + PVN	All
MAP Access (MAPA)	Sequence-Controlled and Expedited, Asynchronous or Periodic	MAP_SDU	GMAP ID	All
MAP Octet Stream	Asynchronous or Periodic	Octet Stream Data	GMAP ID	All
Master Channel Operational Control Field (MC_OCF)	Synchronous or Periodic	OCF_SDU	GVCID	None
Virtual Channel Frame (VCF)	Asynchronous or Periodic	Transfer Frame	GVCID	None
Master Channel Frame (MCF)	Asynchronous or Periodic	Transfer Frame	MCID	None
Insert	Isochronous	IN_SDU	Physical Channel Name	None
COPs Management	N/A	N/A	GVCID	N/A

2.2.3.2 MAP Packet (MAPP) Service

The MAP Packet (MAPP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link on a specified MAP Channel. The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS. CCSDS PVNs are defined in reference [8].

The service is unidirectional and asynchronous. If the COP is used, then both Sequence-Controlled and Expedited service types are provided for the MAP Packet Service.

The use of a reliable ARQ protocol allows multiple users (i.e., Packets with different PVNs) to share a single GMAP ID. In this case the service provider multiplexes Packets of different PVNs to form a single stream of Packets to be transferred on that GMAP ID.

When no reliable ARQ protocol is used, only Expedited service is provided for a MAP Channel. In this case, a user is identified with a single PVN and a GMAP ID.

2.2.3.3 MAP Access (MAPA) Service

The MAP Access (MAPA) Service provides transfer of a sequence of privately formatted service data units of variable length across a space link. The length of the service data units transferred by this service is not constrained by the length of the Data Field of the Transfer Frame.

For a given service instance, multiple users, each identified with the GMAP ID of the MAP Channel, can use this service on a MAP Channel. Service data units from different users may be multiplexed together within one VC Channel as long as these SDUs are contained in independent Transfer Frame Data Zones (TFDZ).

2.2.3.4 MAP Octet Stream Service

The MAP Octet Stream service provides transfer of a serial string of aligned octets, whose internal structure and boundaries are unknown to the service provider, across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to the receiving user.

Octet Streams from different users may be multiplexed together within one Virtual Channel using different MAP IDs.

2.2.3.5 Master Channel Operational Control Field (MC_OCF) Service

The Master Channel Operational Control Field (MC_OCF) Service provides transfer of fixed-length data units, each consisting of four octets, in the Operational Control Field (OCF) of Transfer Frames of a Virtual Channel. The service is unidirectional and sequence-preserving. The transfer is signaled for inclusion within Transfer Frames of a Virtual Channel by using the OCF Flag in the Transfer Frame Primary Header. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to the receiving user.

For a given service instance, one or more users, identified with the GVCID of the Virtual Channel, can use this service on a Master Channel. Service data units from different users can be multiplexed together within one Master Channel.

2.2.3.6 Virtual Channel Frame (VCF) Service

The Virtual Channel Frame (VCF) Service provides transfer of a sequence of fixed-length USLP Transfer Frames of a Virtual Channel, created by an independent protocol entity,

across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to the receiving user.

For a given service instance, only one user, identified with the GVCID of the Virtual Channel, can use this service on a Virtual Channel. Service data units from different users are not multiplexed together within one Virtual Channel.

The Virtual Channel Frame Service transfers the independently created USLP Transfer Frames through a space link, together with USLP Transfer Frames created by the service provider itself. This service is made available to trusted users who are certified during the design process to ensure that the independently created protocol data units do not violate the operational integrity of the space link.

2.2.3.7 Master Channel Frame (MCF) Service

The Master Channel Frame (MCF) Service provides transfer of a sequence of USLP Transfer Frames of a Master Channel, created by an independent protocol entity, across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to a receiving user.

For a given service instance, only one user, identified with the MCID of the Master Channel, can use this service on a Master Channel. Service data units from different users are not multiplexed together within one Master Channel.

The Master Channel Frame Service transfers the independently created USLP Transfer Frames through the space link, together with USLP Transfer Frames created by the service provider itself. This service is made available to trusted users who are certified during the design process to ensure that the independently created protocol data units do not violate the operational integrity of the space link.

2.2.3.8 Insert Service

The Insert Service provides transfer of private, fixed-length, octet-aligned service data units across a space link in a mode which efficiently utilizes the space link transmission resources at relatively low data rates. The service is isochronous, unidirectional, periodic, and sequence-preserving. The service does not guarantee completeness, but may signal gaps in the sequence of service data units delivered to a receiving user.

For a given service instance, only one user, identified with the Physical Channel Name of the Physical Channel, can use this service on a Physical Channel. Service data units from different users are not multiplexed together within one Physical Channel. The presence of the Insert Zone is signaled by Managed Parameters.

2.2.3.9 COPs Management Service

The COPs Management Service is used by a user at the sending end for managing the operations of either COP-1 or COP-P for a particular Virtual Channel. The user manages the operations of the COPs by invoking Directives defined in reference [9] for COP-1 and reference [10] for COP-P. The user is notified by the service provider of events associated with Directives and events that occur asynchronously with Directives.

A user of this service must be authorized to manage the COPs for a particular Virtual Channel. For a given service instance, only one user, identified with the GVCID of the Virtual Channel, is allowed to use this service on a Virtual Channel.

2.2.4 RESTRICTIONS ON SERVICES

There are some restrictions on the services provided on a Physical Channel, as follows:

- a) For fixed-length Transfer Frames only on one MAP Channel, the MAP Access Service shall not exist simultaneously with the MAP Packet Service.
- b) On one Master Channel, only one Master Channel Operational Control Field (MC_OCF) Service can exist.

2.3 OVERVIEW OF FUNCTIONS

2.3.1 GENERAL FUNCTIONS

USLP transfers various service data units, supplied by sending users, encapsulated in a sequence of protocol data units using services of lower layers. The protocol data units, known as USLP Transfer Frames, either have variable lengths and must be transferred over a Physical Channel asynchronously, or have a fixed length and must be transferred over a Physical Channel at a constant rate.

The protocol entity performs the following protocol functions:

- a) generation and processing of protocol control information (i.e., headers and trailers) to perform data identification, loss detection, and error detection;
- b) segmenting and blocking of service data units to transfer variable-length service data units in fixed-length protocol data units;
- c) multiplexing/demultiplexing and commutation/decommutation in order for various service users to share a single Physical Channel.

If the protocol entity supports the optional SDLS protocol, then it uses the functions of SDLS to apply the configured security features.

If the protocol entity supports the optional COP procedures, then it uses the functions of the COP to supply the sequence controlled features.

This protocol entity does not perform the following protocol functions:

- a) connection establishment and release;
- b) flow control;
- c) retransmission of protocol data units;
- d) management or configuration of the SDLS protocol.

2.3.2 INTERNAL ORGANIZATION OF PROTOCOL ENTITY

Figures 2-5 and 2-6 show the internal organization of the protocol entity of the sending and receiving ends, respectively. Data flow from top to bottom in figure 2-5 and from bottom to top in figure 2-6. These figures identify data-handling functions performed by the protocol entity and show logical relationships among these functions. The figures are not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all of the functions may be present in the protocol entity.

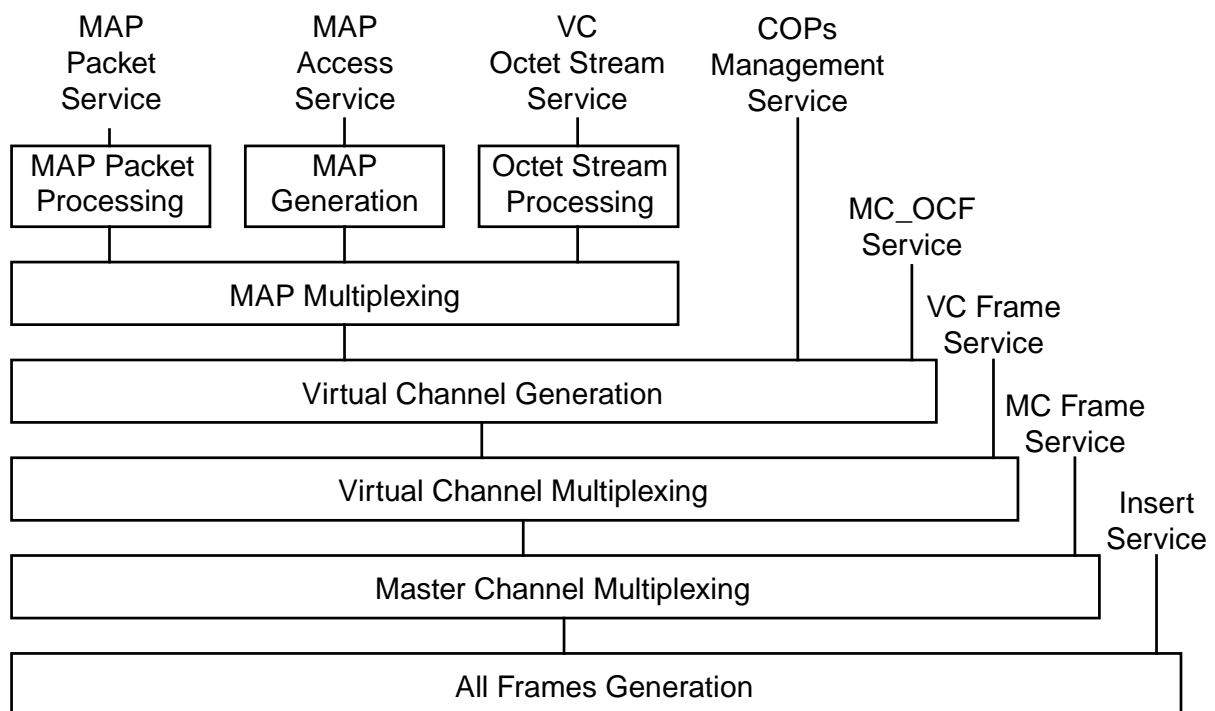


Figure 2-5: Internal Organization of Protocol Entity (Sending End)

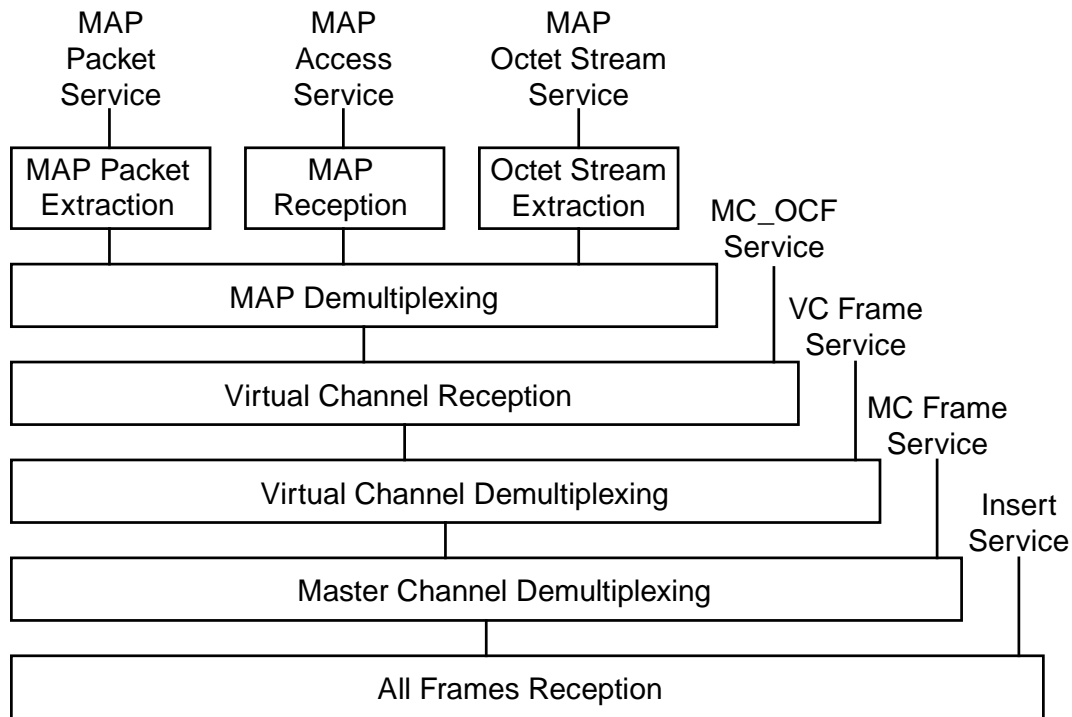


Figure 2-6: Internal Organization of Protocol Entity (Receiving End)

By extracting multiplexing/demultiplexing and commutation/decommutation functions from figures 2-5 and 2-6, the relationship among various data units can be shown as figure 2-7, which is known as the Channel Tree of USLP.

In figure 2-7, multiplexing (shown with a triangle) is a function of mixing, according to an algorithm established by the project, multiple streams of data units, each with a different identifier, to generate a single stream of data units. Commutation (shown with a box) is a function of concatenating, according to the formatting rule specified by the protocol definition, multiple data units, each from a different service, in a single protocol data unit sharing the same identifier.

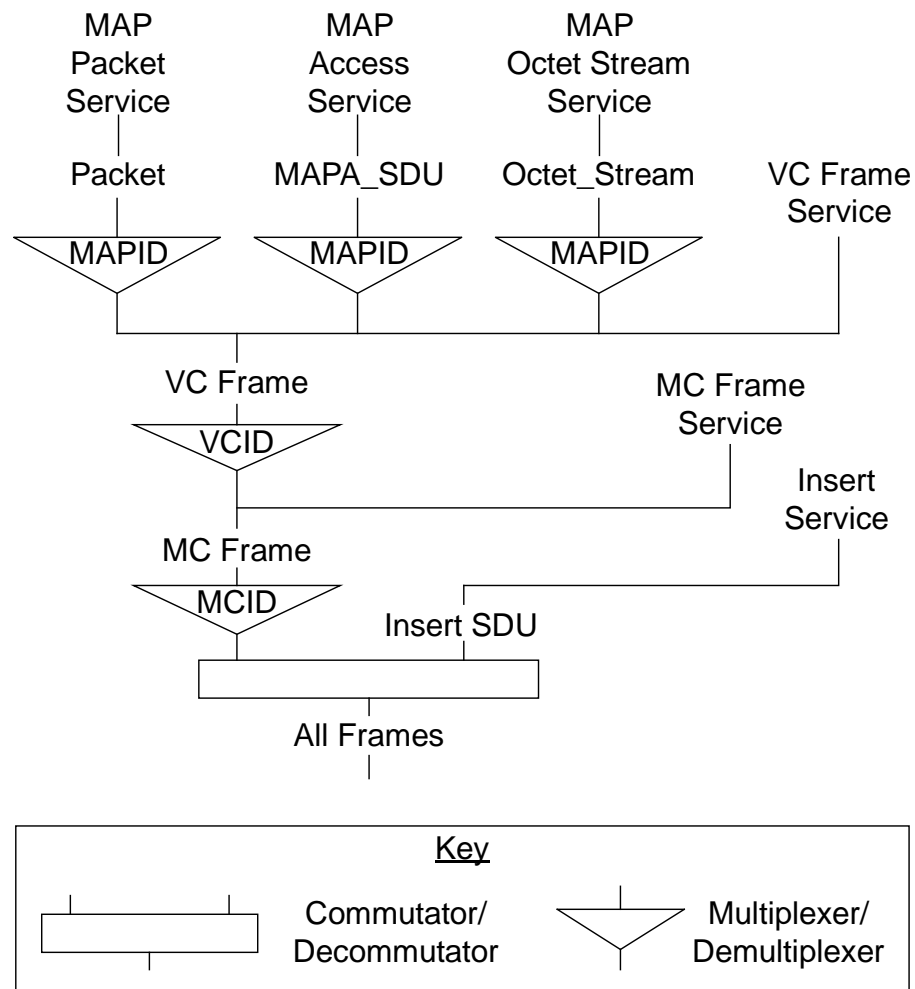


Figure 2-7: Unified Space Data Link Protocol Channel Tree

2.4 SERVICES ASSUMED FROM LOWER LAYERS

2.4.1 SERVICES ASSUMED FROM THE SYNCHRONIZATION AND CHANNEL CODING SUBLAYER

As described in 2.1.1, one of the set of Channel Coding and Synchronization Recommended Standards (references [3], [4], [5], [6] and [7]) are to be used with USLP with the constraints listed in this document as the Synchronization and Channel Coding Sublayer specification. The functions provided by the Synchronization and Channel Coding Recommended Standard are:

- error control encoding and decoding functions (optional when the coding schemes defined in references [3] through [7] are used);
- bit transition generation and removal functions (optional);
- delimiting and synchronizing functions.

When any of the coding schemes defined in references [3], [4], and [5] are used, the Synchronization and Channel Coding Sublayer, then, transfers fixed-length, delimited Transfer Frames as a contiguous stream of bits over a space link using the services of the underlying Physical Layer.

When any of the coding schemes defined in references [6] and [7] are used, the Synchronization and Channel Coding Sublayer, then, transfers variable-length, delimited Transfer Frames as an intermittent stream of bits over a space link using the services of the underlying Physical Layer.

For all coding schemes defined in references [3] through [7], one and only one Transfer Frame at a time is provided by the USLP Data Protocol Sublayer to the Synchronization and Channel Coding Sublayer.

2.4.2 SYSTEMATIC RETRANSMISSIONS

In addition, USLP, when reference [6] is applied, can request the Synchronization and Channel Coding Sublayer to perform systematic retransmissions of the data units submitted to it. The retransmissions can improve the probability of complete delivery for deep space missions on links with long light time delays. This mechanism is not efficient but may be the best approach when frame reporting has significant lag time and thus reduces the efficiency of the COP-1 Go-Back-*N* retransmission process.

The definition of the service interface to the Synchronization and Channel Coding Sublayer specified in reference [6] includes the ChannelAccess.request service primitive, which has an optional Repetitions parameter. The sublayer transfers the data unit the number of times specified by Repetitions. If the value of Repetitions is one, or if the sublayer does not support the Repetitions parameter, then no systematic retransmissions are performed, and the frame is transferred once.

USLP requests the systematic retransmissions in accordance with parameters set by management. For each MAP, management sets the value to be used for the Repetitions parameter when requesting the transfer of frames carrying service data units on the Sequence-Controlled Service. For each Virtual Channel, management sets a similar parameter for frames carrying COP control commands. For a Physical Channel, management sets an upper limit for the value of the Repetitions parameter specified in reference [6].

When requesting the transfer of frames carrying service data units on the Expedited Service, USLP does not limit the value of the Repetitions parameter.

2.4.3 PERFORMANCE REQUIREMENTS TO LOWER LAYERS

The coding options of the Channel Coding and Synchronization Recommended Standard and the performance of the RF link provided by the Physical Layer shall be chosen according to the following criteria:

- a) The probability of loss of a Transfer Frame shall be less than a mission-specified value;
- b) The probability of not identifying a frame that is in error shall be less than a mission-specified value;
- c) The probability of misidentifying the MCID and VCID shall be less than a mission-specified value;
- d) The probability of incorrectly extracting Packets from Transfer Frames using the First Header Pointer and the Packet Length Field shall be less than a mission-specified value.
- e) The probability of incorrectly extracting the Packet Length Field from the Transfer Frame Header shall be less than a mission-specified value.

In order to assure correct decoding at the receiving end, the same coding options must be applied to all Transfer Frames of a Physical Channel.

3 SERVICE DEFINITION

3.1 OVERVIEW

This section provides service definition in the form of primitives, which present an abstract model of the logical exchange of data and control information between the protocol entity and the service user. The definitions of primitives are independent of specific implementation approaches.

The parameters of the primitives are specified in an abstract sense and specify the information to be made available to the user of the primitives. The way in which a specific implementation makes this information available is not constrained by this specification. In addition to the parameters specified in this section, an implementation may provide other parameters to the service user (e.g., parameters for controlling the service, monitoring performance, facilitating diagnosis, and so on).

3.2 SOURCE DATA

3.2.1 SOURCE DATA OVERVIEW

NOTE – This subsection describes the service data units that are transferred from sending users to receiving users by USLP.

The service data units transferred by USLP shall be:

- a) MAP Packet;
- b) MAP Access Service Data Unit (MAPA_SDU);
- c) MAP Octet Stream Data;
- d) Operational Control Field Service Data Unit (OCF_SDU);
- e) USLP Transfer Frame;
- f) Insert Service Data Unit (IN_SDU).

3.2.2 MAP PACKET

3.2.2.1 Packets shall be transferred over a space link with the MAP Packet Service.

3.2.2.2 The Packets transferred by this service must be CCSDS specified, self-delimiting, and have a Packet Version Number (PVN) registered by SANA (reference [8]).

3.2.2.3 The position and length of the Packet Length Field of the Packets must be known to the service provider in order to extract Packets from Transfer Frames at the receiving end. This requirement applies when packets span Transfer Frames using TFDF Construction Rule ‘000’ and also when blocking of Packets is performed using TFDF Construction Rule ‘111’ by the service provider.

NOTES

- 1 Packets are variable-length, delimited, octet-aligned data units.
- 2 Examples of packets are: CCSDS Space Packets, CCSDS Encapsulation Packets.

3.2.2.4 Packets shall be contained either within a single frame or within multiple sequential frames of the same GMAP ID using the TFDZ Construction Rules in 4.1.4.2.1.3.

3.2.3 MAP ACCESS SERVICE DATA UNIT (MAP_SDU)

3.2.3.1 MAP_SDUs shall be transferred over a space link via the MAP Channel Access Service.

3.2.3.2 A single MAP_SDU may be transmitted in the Data Zone of one or multiple Transfer Frame(s), and therefore the length of MAP_SDUs is not constrained by the length of the Transfer Frame Data Zone.

NOTE – MAP Channel Access Service Data Units (MAP_SDUs) are variable-length, octet-aligned data units, the format of which is unknown to the service provider. Their length is provided to the SAP and they are delimited within the TFDZ using the TFDZ construction rules.

3.2.4 MAP OCTET STREAM DATA

3.2.4.1 Octet Stream Data shall be transferred over a space link with the MAP Octet Stream Service.

3.2.4.2 The length of the Octet Stream Data supplied in each Octet Stream service request shall be used to delimit the received data that is to be transferred within the TFDZ.

NOTE – Octet Stream Data are variable-length, octet-aligned, continuous string of octets, the format of which is unknown to the service provider.

3.2.5 OPERATIONAL CONTROL FIELD SERVICE DATA UNIT (OCF_SDU)

3.2.5.1 Operational Control Field Service Data Units (OCF_SDUs) shall be transferred over a space link with the MC_OCF Service.

3.2.5.2 The OCF Flag in the Transfer Frame primary header shall signal the presence or absence of this service.

NOTES

- 1 Although the transfer of OCF_SDUs is carried within the Virtual Channel frame, which by management can provide the OCF service, the creation of OCF_SDUs by the sending user may or may not be synchronized with a single Virtual Channel. Such synchronization, if required for timing or other purposes, is a mission-design issue.
- 2 OCF_SDUs are fixed-length data units, each consisting of four octets, carried in the Operational Control Field (OCF), defined in 4.1.5, from a sending end to a receiving end.
- 3 As defined in 4.1.5, CCSDS specifies the use of the first bit of this field to indicate the type of data carried.

3.2.6 DISCUSSION—USLP TRANSFER FRAME

The USLP Transfer Frame is the variable-length protocol data unit of USLP but also can be used as a service data unit of the Virtual Channel Frame and Master Channel Frame Services. Its format is defined in 4.1 and 6.3 of this Recommended Standard.

3.2.7 INSERT SERVICE DATA UNIT (IN_SDU)

3.2.7.1 Insert Service Data Units (IN_SDUs) shall be isochronous, periodic, octet-aligned data units of fixed length, and their presence within the frame shall be controlled by Managed Parameters.

3.2.7.2 When present, each Transfer Frame on the Physical Channel shall contain one IN_SDU, and the Transfer Frame must be of constant length and be aligned with the codeword in order to maintain periodicity.

3.2.7.3 IN_SDU length shall be established by management and may be of any constant value that is an integral number of octets, between 1 octet and the maximum length of the data-carrying space of the Transfer Frames.

NOTE – The length of IN_SDUs at the sending interface is always equal to the length at the receiving interface.

3.3 MAP PACKET SERVICE

3.3.1 OVERVIEW OF PACKET SERVICE

The MAP Packet (MAPP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link. The Packets transferred by this service must have a Packet Version Number (PVN) recognized by CCSDS. Packet Version Numbers are registered in reference [8]. The service is unidirectional, asynchronous and sequence-preserving. It does not guarantee completeness, nor does it signal gaps in the sequence of service data units delivered to a receiving user.

A user of this service is a protocol entity that sends or receives Packets with a single PVN and identified with the PVN and a GMAP ID. Different users (i.e., Packets with different PVNs) may share a single MAP Channel, and if there are multiple users on a MAP Channel, the service provider multiplexes Packets of different versions to form a single stream of Packets to be transferred on that MAP Channel.

3.3.2 MAP PACKET SERVICE PARAMETERS

3.3.2.1 General

The parameters used by the MAPP Service primitives shall conform to the specifications contained in subsections 3.3.2.2 through 3.3.2.9.

3.3.2.2 Packet

The Packet parameter shall contain a Packet for transfer by the MAP Packet Service.

NOTE – The Packet parameter is the service data unit transferred by the MAP Packet Service. Restrictions on the Packets transferred by the MAP Packet Service are stated in 3.2.2.

3.3.2.3 GMAP ID

The GMAP ID parameter shall contain a GMAP ID that indicates the MAP Channel through which the Packet is to be transferred.

NOTE – The GMAP ID is part of the SAP address of the MAP Packet Service.

3.3.2.4 Packet Version Number

The Packet Version Number (PVN) shall identify the CCSDS protocol entity of the upper layer that uses the MAP Packet Service.

NOTE – The PVN is part of the SAP address of the MAP Packet Service.

3.3.2.5 SDU ID

The SDU ID parameter shall contain a user-supplied sequence number to be used to identify the associated Packet in subsequent MAPP_Notify.indication primitives.

3.3.2.6 Service Type

At the sending end, the Service Type parameter shall indicate whether the Packet should be transferred with the Sequence-Controlled Service type or the Expedited Service type.

3.3.2.7 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with the transfer of a Packet. The values taken by this parameter are defined in reference [9].

3.3.2.8 Packet Quality Indicator

The Packet Quality Indicator is an optional parameter that may be used to notify the user at the receiving end of the Packet Service whether the Packet delivered by the primitive is complete or partial.

3.3.2.9 Verification Status Code

3.3.2.9.1 The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.

3.3.2.9.2 The Verification Status Code parameter shall be used to notify the user at the receiving end of the Packet Service of a verification failure in a Transfer Frame addressed to the MAP Channel.

3.3.2.9.3 A non-zero value shall indicate that the SDLS protocol has detected an error; the values taken by this parameter are defined in reference [14].

NOTE – A non-zero value of the Verification Status Code does not indicate an error in the delivered Packet. Processing of frames failing verification is implementation-specific and depends also on the processing capabilities of the service user for eventual forensic investigation.

3.3.3 MAPP SERVICE PRIMITIVES

3.3.3.1 General

The service primitives associated with the MAPP service are:

- a) MAPP.request;
- b) MAPP_Notify.indication;
- c) MAPP.indication.

3.3.3.2 MAPP.request

3.3.3.2.1 Function

At the sending end, the MAPP Service user shall pass a MAPP.request primitive to the service provider to request that a Packet be transferred to the user at the receiving end through the specified MAP Channel.

3.3.3.2.2 Semantics

The MAPP.request primitive shall provide parameters as follows:

MAPP.request	(Packet, GVCID, MAP ID, Packet Version Number, SDU ID, Service Type)
--------------	---

NOTE – When separate ports are provided for Sequence-Controlled and Expedited Services, the Service Type parameter is not used in this primitive.

3.3.3.2.3 When Generated

The sending-end user shall generate a MAPP.request primitive when a Packet is ready to be transferred.

3.3.3.2.4 Effect On Receipt

Receipt of the MAPP.request primitive shall cause the service provider to transfer the Packet.

3.3.3.3 MAPP_Notify.indication

3.3.3.3.1 Function

At the sending end, the service provider shall pass a MAPP_Notify.indication primitive to the MAPP Service user to notify the user of an event associated with the transfer of a Packet.

3.3.3.3.2 Semantics

The MAPP_Notify.indication primitive shall provide parameters as follows:

MAPP_Notify.indication	(GVCID, MAP ID, Packet Version Number, SDU ID, Service Type, Notification Type)
------------------------	--

NOTE – When separate ports are provided for Sequence-Controlled and Expedited Services, the Service Type parameter is not used in this primitive.

3.3.3.3.3 When Generated

The sending-end service provider shall generate a MAPP_Notify.indication primitive in response to an event associated with the transfer of a Packet.

3.3.3.3.4 Effect On Receipt

The effect of receipt of the MAPP_Notify.indication primitive by the MAPP Service user is undefined.

3.3.3.4 MAPP.indication

3.3.3.4.1 Function

At the receiving end, the service provider shall pass a MAPP.indication to the MAPP Service user to deliver a Packet.

3.3.3.4.2 Semantics

The MAPP.indication primitive shall provide parameters as follows:

MAPP.indication	(Packet, GVCID, MAP ID, Packet Version Number, Service Type (optional), Packet Quality Indicator (optional), Verification Status Code (optional))
-----------------	---

3.3.3.4.3 When Generated

The receiving-end service provider shall generate a MAPP.indication primitive when a Packet is ready to be delivered.

3.3.3.4.4 Effect On Receipt

The effect of receipt of the MAPP.indication primitive by the MAPP Service user is undefined.

3.4 MAP ACCESS SERVICE

3.4.1 OVERVIEW

The MAP Access (MAPA) Service provides transfer of a sequence of privately formatted, octet-aligned, variable-length service data units across a space link. The length of the data unit is unknown to the service provider and must be conveyed to the service provider at the service access point. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but may signal gaps in the sequence of service data units delivered to the receiving user.

A user of this service is a protocol entity that sends or receives a MAP_SDU within one or more frames identified with a USLP Protocol Identifier (UPID) and a GMAP ID. Different users (i.e., MAP_SDU identified with a different UPID) may share a single MAP Channel, but cannot share a single TFDZ. The service provider isolates each MAP_SDU to a single frame or sequential numbered frames using the TFDZ construction flags (see 4.1.4.2.1.3) on the same GMAP ID.

3.4.2 MAPA SERVICE PARAMETERS

3.4.2.1 General

The parameters used by the MAPA Service primitives shall conform to the specifications contained in subsections 3.4.2.2 through 3.4.2.5.

3.4.2.2 MAP_SDU

The parameter MAP_SDU shall be the service data unit transferred by the MAPA Service.

NOTE – Restrictions on the MAP_SDUs transferred by the MAPA Service are stated in 3.2.3.

3.4.2.3 GMAP ID

The GMAP ID parameter shall contain a GMAP ID that indicates the MAP Channel through which the MAP_SDU is to be transferred.

NOTE – The GMAP ID is the SAP address of the MAPA Service.

3.4.2.4 MAP_SDU Loss Flag

3.4.2.4.1 The MAP_SDU Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the MAPA Service that a sequence discontinuity has been detected, and that one or more MAP_SDUs have been lost.

3.4.2.4.2 If implemented, the flag shall be derived by examining the Virtual Channel Frame Count in the Transfer Frames.

3.4.2.5 Verification Status Code

3.4.2.5.1 The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.

3.4.2.5.2 The parameter shall be used to notify the user at the receiving end of the VCA Service of a verification failure in a Transfer Frame addressed to the Virtual Channel.

3.4.2.5.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [14].

NOTE – A non-zero value of the Verification Status Code does not indicate an error in the delivered MAP_SDU. Processing of frames failing verification is implementation-specific and depends also on the processing capabilities of the service user for eventual forensic investigation.

3.4.3 MAPA SERVICE PRIMITIVES

3.4.3.1 General

The service primitives associated with the MAPA service are:

- a) MAPA.request;
- b) MAPA_Notify.indication;
- c) MAPA.indication.

3.4.3.2 MAPA.request

3.4.3.2.1 Function

At the sending end, the MAPA Service user shall pass a MAPA.request primitive to the service provider to request that a MAP_SDU be transferred to the user at the receiving end through the specified MAP Channel.

NOTE – The MAPA.request primitive is the service request primitive for the MAPA Service.

3.4.3.2.2 Semantics

The MAPA.request primitive shall provide parameters as follows:

MAPA.request	(MAP_SDU, GVCID, MAP ID, SDU ID, Service Type)
--------------	--

NOTE – When separate ports are provided for Sequence-Controlled and Expedited Services, the Service Type parameter is not used in this primitive.

3.4.3.2.3 When Generated

The sending-end service user shall generate a MAPA.request when a MAP_SDU is ready to be transferred.

3.4.3.2.4 Effect On Receipt

Receipt of the MAPA.request primitive shall cause the service provider to transfer the MAP_SDU.

3.4.3.3 MAPA_Notify.indication

3.4.3.3.1 Function

At the sending end, the service provider shall pass a MAPA_Notify.indication primitive to the MAPA Service user to notify the user of an event associated with the transfer of a MAP_SDU.

3.4.3.3.2 Semantics

The MAPA.indication primitive shall provide parameters as follows:

MAPA_Notify.indication	(GVCID, MAP ID, SDU ID, Service Type, Notification Type)
------------------------	--

NOTE – When separate ports are provided for Sequence-Controlled and Expedited Services, the Service Type parameter is not used in this primitive.

3.4.3.3.3 When Generated

The sending-end service provider shall generate a MAPA_Notify.indication in response to an event associated with the transfer of a MAP_SDU.

3.4.3.3.4 Effect On Receipt

The effect of receipt of the MAPA_Notify.indication primitive by the MAPA Service user is undefined.

3.4.3.4 MAPA.indication

3.4.3.4.1 Function

At the receiving end, the service provider shall pass a MAPA.indication to the MAPA Service user to deliver a MAP_SDU.

NOTE – The MAPA.indication primitive is the service indication primitive for the MAPA Service.

3.4.3.4.2 Semantics

The MAPA.indication primitive shall provide parameters as follows:

MAPA.indication	(MAP_SDU, GVCID, MAP ID, Service Type (optional), Verification Status Code (optional))
-----------------	--

3.4.3.4.3 When Generated

The receiving-end service provider shall generate a MAPA.indication when a MAP_SDU is ready to be delivered.

3.4.3.4.4 Effect On Receipt

The effect of receipt of the MAPA.indication primitive by the MAPA Service user is undefined.

3.5 MAP OCTET STREAM SERVICE

3.5.1 OVERVIEW OF MAP OCTET STREAM SERVICE

The MAP Octet Stream Service provides transfer of a serial string of octets, whose internal structure and boundaries are unknown to the service provider, across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but may signal gaps in the sequence of service data units delivered to the receiving user.

Octet Streams from different users may be multiplexed together within one Virtual Channel using different MAP IDs.

3.5.2 MAP OCTET STREAM SERVICE PARAMETERS

3.5.2.1 General

The parameters used by the Octet Stream Service primitives shall conform to the specifications contained in subsections 3.5.2.2 through 3.5.2.5.

3.5.2.2 Octet Stream Data

The parameter Octet Stream Data shall be the service data unit transferred by the Octet Stream Service.

NOTE – Restrictions on the Octet Stream Data transferred by the Octet Stream Service are stated in 3.2.3.

3.5.2.3 GMAP ID

The GMAP ID parameter shall contain a GMAP that indicates the MAP through which the Octet Stream Data is to be transferred.

NOTE – The GMAP ID is the SAP address of the Octet Stream Service.

3.5.2.4 Octet Stream Data Loss Flag

The Octet Stream Data Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the Octet Stream Service that a sequence discontinuity has been detected and that some Octet Stream Data may have been lost. If implemented, the flag shall be derived by examining the Virtual Channel Frame Count in the Transfer Frames.

NOTE – As the contents (valid Octet Stream Data or idle data) of lost Transfer Frames cannot be established, the user should be aware that the Octet Stream Data Loss Flag signals a disruption in the Transfer Frames of the specified Virtual Channel, and not necessarily a disruption of the Octet Stream Data itself.

3.5.2.5 Verification Status Code

3.5.2.5.1 The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.

3.5.2.5.2 The Verification Status Code parameter shall be used to notify the user at the receiving end of the Octet Stream Service of a verification failure in a Transfer Frame addressed to the Virtual Channel.

3.5.2.5.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [14].

NOTE – A non-zero value of the Verification Status Code does not indicate an error in the delivered Octet Stream Data. Processing of frames failing verification is implementation-specific and depends also on the processing capabilities of the service user for eventual forensic investigation.

3.5.3 OCTET STREAM SERVICE PRIMITIVES

3.5.3.1 General

The service primitives associated with this service are:

- a) OCTET_STREAM.request;
- b) OCTET_STREAM.indication.

3.5.3.2 OCTET_STREAM.request

3.5.3.2.1 Function

At the sending end, the Octet Stream Service user shall pass an OCTET_STREAM.request primitive to the service provider to request that Octet Stream Data be transferred to the user at the receiving end through the specified Virtual Channel.

NOTE – The OCTET_STREAM.request primitive is the service request primitive for the Octet Stream Service.

3.5.3.2.2 Semantics

The OCTET_STREAM.request primitive shall provide parameters as follows:

OCTET_STREAM.request (Octet Stream Data,
GMAP ID)

3.5.3.2.3 When Generated

The sending-end service provider shall generate an OCTET_STREAM.request when Octet Stream Data is ready to be transferred.

3.5.3.2.4 Effect On Receipt

Receipt of the OCTET_STREAM.request primitive shall cause the service provider to transfer the Octet Stream Data.

3.5.3.2.5 Discussion—Additional Comments

The OCTET_STREAM.request primitive is used to transfer Octet Stream Data across the space link on the specified Virtual Channel.

Since the service interface specification is an abstract specification, the implementation of the Octet Stream Data parameter is not constrained; i.e., it may be continuous Octet Stream, delimited Octet Stream, or individual bits.

3.5.3.3 OCTET_STREAM.indication

3.5.3.3.1 Function

At the receiving end, the service provider shall pass an OCTET_STREAM.indication to the OCTET STREAM Service user to deliver Octet Stream Data.

NOTE – The OCTET_STREAM.indication primitive is the service indication primitive for the Octet Stream Service.

3.5.3.3.2 Semantics

The OCTET_STREAM.indication primitive shall provide parameters as follows:

OCTET_STREAM.indication	(Octet Stream Data, GMAP ID, Octet Stream Data Loss Flag (optional), Verification Status Code (optional))
-------------------------	--

3.5.3.3.3 When Generated

The receiving-end service provider shall generate an OCTET_STREAM.indication when Octet Stream Data ready to be delivered.

3.5.3.3.4 Effect On Receipt

The effect of receipt of the OCTET_STREAM.indication primitive by the Octet Stream Service user is undefined.

3.5.3.3.5 Discussion—Additional Comments

The OCTET_STREAM.indication primitive is used to deliver Octet Stream Data to the Octet Stream Service user identified by the GMAP ID.

The quantity of Octet Stream Data delivered by an implementation of this service primitive is not defined. Therefore it is not necessarily related to the quantity of Octet Stream Data submitted to the service provider by the sending user with the OCTET_STREAM.request primitive.

3.6 MASTER CHANNEL OPERATIONAL CONTROL FIELD (MC_OCF) SERVICE

3.6.1 OVERVIEW

The Master Channel Operational Control Field (MC_OCF) Service provides transfer of fixed-length data units, each consisting of four octets, in the Operational Control Field (OCF) of Transfer Frames in a Virtual Channel. The service is unidirectional and sequence-preserving. The inclusion of an OCF_SDU in a Transfer Frame is signaled in the Transfer Frame Header. The transfer is synchronized with the release of Transfer Frames of a Virtual Channel. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to the receiving user.

Only one Master Channel Operational Control Field (MC_OCF) Service can exist within a Master Channel, and that service may be provided by multiple VCs that are designated by the VC managed parameter to support the service.

Only the COP and SDLS users can use this service on a Virtual Channel, because the OCF service can distinguish between their respective reports. The user is identified with the GVCID of the Virtual Channel. Service data units from different OCF users are multiplexed together for delivery within the OCF of the transmitted Virtual Channel.

3.6.2 MC_OCF SERVICE PARAMETERS

3.6.2.1 General

The parameters used by the MC_OCF Service primitives shall conform to the specifications contained in subsections 3.6.2.2 through 3.6.2.4.

3.6.2.2 OCF_SDU

The parameter OCF_SDU shall be the service data unit transferred by the MC_OCF Service in the Operational Control Field of Transfer Frames of a Virtual Channel.

NOTE – Restrictions on the OCF_SDU transferred by the MC_OCF Service are stated in 3.2.5.

3.6.2.3 GVCID

The GVCID parameter shall contain a GVCID that indicates the Virtual Channel through which the OCF_SDU is to be transferred.

NOTE – The GVCID is the SAP address of the MC_OCF Service.

3.6.2.4 OCF_SDU Loss Flag

3.6.2.4.1 The OCF_SDU Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the MC_OCF Service that a sequence discontinuity has been detected and that one or more OCF_SDUs may have been lost.

3.6.2.4.2 If implemented, the flag shall be derived by examining the Virtual Channel Frame Count in the Transfer Frames.

3.6.3 MC_OCF SERVICE PRIMITIVES

3.6.3.1 General

The service primitives associated with this service are:

- a) MC_OCF.request;
- b) MC_OCF.indication.

3.6.3.2 MC_OCF.request

3.6.3.2.1 Function

At the sending end, the MC_OCF Service user shall pass a MC_OCF.request primitive to the service provider to request that an OCF_SDU be transferred to the user at the receiving end through the specified Virtual Channel.

NOTE – The MC_OCF.request primitive is the service request primitive for the MC_OCF Service.

3.6.3.2.2 Semantics

The MC_OCF.request primitive shall provide parameters as follows:

MC_OCF.request	(OCF_SDU, GVCID)
----------------	---------------------

3.6.3.2.3 When Generated

The MC_OCF.request primitive shall be passed to the service provider to request it to send the OCF_SDU.

3.6.3.2.4 Effect On Receipt

Receipt of the MC_OCF.request primitive shall cause the service provider to transfer the OCF_SDU.

3.6.3.2.5 Discussion—Additional Comments

The MC_OCF.request primitive is used to transfer OCF_SDUs across the space link on the specified Virtual Channel.

3.6.3.3 MC_OCF.indication

3.6.3.3.1 Function

At the receiving end, the service provider shall pass a MC_OCF.indication to the MC_OCF Service user to deliver an OCF_SDU.

NOTE – The MC_OCF.indication primitive is the service indication primitive for the MC_OCF Service.

3.6.3.3.2 Semantics

The MC_OCF.indication primitive shall provide parameters as follows:

MC_OCF.indication	(OCF_SDU, GVCID, OCF_SDU Loss Flag (optional))
-------------------	--

3.6.3.3.3 When Generated

The MC_OCF.indication primitive shall be passed from the service provider to the MC_OCF Service user at the receiving end to deliver an OCF_SDU.

3.6.3.3.4 Effect On Receipt

The effect of receipt of the MC_OCF.indication primitive by the MC_OCF Service user is undefined.

3.6.3.3.5 Discussion—Additional Comments

The MC_OCF.indication primitive is used to deliver OCF_SDUs to the MC_OCF Service user identified by the GVCID.

3.7 VIRTUAL CHANNEL FRAME (VCF) SERVICE

3.7.1 OVERVIEW OF VCF SERVICE

The Virtual Channel Frame (VCF) Service provides transfer of a sequence of variable-length USLP Transfer Frames of a Virtual Channel, created by an independent protocol entity, across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to the receiving user.

Only one user can use this service on a Virtual Channel, and the user is identified with the GVCID of the Virtual Channel. Service data units from different users are not multiplexed together within one Virtual Channel.

3.7.2 VCF SERVICE PARAMETERS

3.7.2.1 General

The parameters used by the VCF Service primitives shall conform to the specifications contained in subsections 3.7.2.2 through 3.7.2.4.

3.7.2.2 Frame

The Frame parameter shall be a USLP Transfer Frame of the Virtual Channel specified by the GVCID parameter.

NOTES

- 1 The parameter Frame is the service data unit transferred by the VCF Service.
- 2 The format of the GVCID parameter is defined in 4.1.

3.7.2.3 GVCID

The GVCID parameter shall contain a GVCID that indicates the Virtual Channel through which the Frame is to be transferred.

NOTE – The GVCID is the SAP address of the VCF Service.

3.7.2.4 Frame Loss Flag

3.7.2.4.1 The Frame Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the VCF Service that a sequence discontinuity has been detected and that one or more Transfer Frames of the specified Virtual Channel have been lost.

3.7.2.4.2 If implemented, the flag shall be derived by examining the Virtual Channel Frame Count in the Transfer Frames.

3.7.3 VCF SERVICE PRIMITIVES

3.7.3.1 General

The service primitives associated with this service are:

- a) VCF.request;
- b) VCF.indication.

3.7.3.2 VCF.request

3.7.3.2.1 Function

At the sending end, the VCF Service user shall pass a VCF.request primitive to the service provider to request that a Frame be transferred to the user at the receiving end through the specified Virtual Channel.

NOTE – The VCF.request primitive is the service request primitive for the VCF Service.

3.7.3.2.2 Semantics

The VCF.request primitive shall provide parameters as follows:

VCF.request	(Frame, GVCID)
-------------	-------------------

3.7.3.2.3 When Generated

The VCF.request primitive shall be passed to the service provider to request it to send the Frame.

3.7.3.2.4 Effect On Receipt

Receipt of the VCF.request primitive causes the service provider to transfer the Frame.

3.7.3.2.5 Discussion—Additional Comments

The VCF.request primitive is used to transfer Transfer Frames of a Virtual Channel across the space link.

3.7.3.3 VCF.indication

3.7.3.3.1 Function

At the receiving end, the service provider shall pass a VCF.indication to the VCF Service user to deliver a frame.

NOTE – The VCF.indication primitive is the service indication primitive for the VCF Service.

3.7.3.3.2 Semantics

The VCF.indication primitive shall provide parameters as follows:

VCF.indication	(Frame, GVCID, Frame Loss Flag (optional))
----------------	--

3.7.3.3.3 When Generated

The VCF.indication primitive is passed from the service provider to the VCF Service user at the receiving end to deliver a Frame.

3.7.3.3.4 Effect On Receipt

The effect of receipt of the VCF.indication primitive by the VCF Service user is undefined.

3.7.3.3.5 Discussion—Additional Comments

The VCF.indication primitive is used to deliver Transfer Frames of a Virtual Channel to the VCF Service user identified by the GVCID.

3.8 MASTER CHANNEL FRAME (MCF) SERVICE

3.8.1 OVERVIEW OF MCF SERVICE

The Master Channel Frame (MCF) Service provides transfer of a sequence of fixed-length USLP Transfer Frames of a Master Channel, created by an independent protocol entity, across a space link. The service is unidirectional, either asynchronous or periodic, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of service data units delivered to a receiving user.

Only one user can use this service on a Master Channel, and the user is identified with the MCID of the Master Channel. Service data units from different users are not multiplexed together within one Master Channel.

3.8.2 MCF SERVICE PARAMETERS

3.8.2.1 General

The parameters used by the MCF Service primitives shall conform to the specifications contained in subsections 3.8.2.2 through 3.8.2.4.

3.8.2.2 Frame

The Frame parameter shall be a USLP Transfer Frame of the Master Channel specified by the MCID parameter.

NOTES

- 1 The parameter Frame is the service data unit transferred by the VCF Service.
- 2 The format of the MCID parameter is defined in 4.1.
- 3 Restrictions on the USLP Transfer Frames transferred by the MCF Service are stated in 3.2.6.

3.8.2.3 MCID

The MCID parameter shall contain the MCID of the Master Channel on which the Frame is to be transferred.

NOTE – The MCID is the SAP address of the MCF Service.

3.8.2.4 Frame Loss Flag

3.8.2.4.1 The Frame Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the MCF Service that a sequence discontinuity has been detected and that one or more Transfer Frames of the specified Master Channel may have been lost.

3.8.2.4.2 If implemented, the flag shall be derived by a signal given by the underlying Synchronization and Channel Coding Sublayer.

3.8.3 MCF SERVICE PRIMITIVES

3.8.3.1 General

The service primitives associated with this service are:

- a) MCF.request;
- b) MCF.indication.

3.8.3.2 MCF.request

3.8.3.2.1 Function

At the sending end, the MCF Service user shall pass an MCF.request primitive to the service provider to request that a Frame be transferred to the user at the receiving end through the specified Master Channel.

NOTE – The MCF.request primitive is the service request primitive for the MCF Service.

3.8.3.2.2 Semantics

The MCF.request primitive shall provide parameters as follows:

MCF.request	(Frame, MCID)
-------------	------------------

3.8.3.2.3 When Generated

The MCF.request primitive shall be passed to the service provider to request it to send the Frame.

3.8.3.2.4 Effect On Receipt

Receipt of the MCF.request primitive shall cause the service provider to transfer the Frame.

3.8.3.2.5 Discussion—Additional Comments

The MCF.request primitive is used to transfer Transfer Frames of a Master Channel across the space link.

3.8.3.3 MCF.indication

3.8.3.3.1 Function

At the receiving end, the service provider shall pass an MCF.indication to the MCF Service user to deliver a Frame.

NOTE – The MCF.indication primitive is the service indication primitive for the MCF Service.

3.8.3.3.2 Semantics

The MCF.indication primitive shall provide parameters as follows:

MCF.indication	(Frame, MCID, Frame Loss Flag (optional))
----------------	---

3.8.3.3.3 When Generated

The MCF.indication primitive shall be passed from the service provider to the MCF Service user at the receiving end to deliver a Frame.

3.8.3.3.4 Effect On Receipt

The effect of receipt of the MCF.indication primitive by the MCF Service user is undefined.

3.8.3.3.5 Discussion—Additional Comments

The MCF.indication primitive is used to deliver Transfer Frames of a Master Channel to the VCF Service user identified by the MCID.

3.9 INSERT SERVICE

3.9.1 OVERVIEW OF INSERT SERVICE

The Insert Service provides transfer of private, fixed-length, octet-aligned service data units across a space link in a mode which efficiently utilizes the space link transmission resources at relatively low data rates. The service is isochronous, unidirectional, periodic, and sequence-preserving. The service does not guarantee completeness, but may signal gaps in the sequence of service data units delivered to a receiving user. The presence of the Insert Zone is controlled by Managed Parameters.

Only one user can use this service on a Physical Channel, and the user is identified with the Physical Channel Name of the Physical Channel. Service data units from different users are not multiplexed together within one Physical Channel.

3.9.2 INSERT SERVICE PARAMETERS

3.9.2.1 General

The parameters used by the Insert Service primitives shall conform to the specifications contained in subsections 3.9.2.2 through 3.9.2.4.

3.9.2.2 IN_SDU

The parameter IN_SDU shall be the service data unit transferred by the Insert Service.

NOTE – Restrictions on the IN_SDUs transferred by the Insert Service are stated in 3.2.7.

3.9.2.3 Physical Channel Name

The Physical Channel Name shall indicate the Physical Channel through which the IN_SDU is to be transferred.

NOTE – The Physical Channel Name is the SAP address of the Insert Service.

3.9.2.4 IN_SDU Loss Flag

3.9.2.4.1 The IN_SDU Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the Insert Service that a sequence discontinuity has been detected and that one or more IN_SDUs have been lost.

3.9.2.4.2 If implemented, the flag shall be derived by a signal given by the underlying Synchronization and Channel Coding Sublayer.

3.9.3 INSERT SERVICE PRIMITIVES

3.9.3.1 General

The service primitives associated with this service are:

- a) INSERT.request;
- b) INSERT.indication.

3.9.3.2 INSERT.request

3.9.3.2.1 Function

At the sending end, the Insert Service user shall pass an INSERT.request primitive to the service provider to request that an IN_SDU be transferred to the user at the receiving end through the specified Physical Channel.

NOTE – The INSERT.request primitive is the service request primitive for the Insert Service.

3.9.3.2.2 Semantics

The INSERT.request primitive shall provide parameters as follows:

INSERT.request	(IN_SDU, Physical Channel Name)
----------------	------------------------------------

3.9.3.2.3 When Generated

The INSERT.request primitive is passed to the service provider to request it to send the IN_SDU.

3.9.3.2.4 Effect On Receipt

Receipt of the INSERT.request primitive causes the service provider to transfer the IN_SDU.

3.9.3.2.5 Discussion—Additional Comments

The INSERT.request primitive is used to transfer IN_SDUs across the space link on the specified Physical Channel.

3.9.3.3 INSERT.indication

3.9.3.3.1 Function

At the receiving end, the service provider shall pass an INSERT.indication to the Insert Service user to deliver an IN_SDU.

NOTE – The INSERT.indication primitive is the service indication primitive for the Insert Service.

3.9.3.3.2 Semantics

The INSERT.indication primitive shall provide parameters as follows:

INSERT.indication	(IN_SDU, Physical Channel Name, IN_SDU Loss Flag (optional))
-------------------	--

3.9.3.3.3 When Generated

The INSERT.indication primitive shall be passed from the service provider to the Insert Service user at the receiving end to deliver an IN_SDU.

3.9.3.3.4 Effect On Receipt

The effect of receipt of the INSERT.indication primitive by the Insert Service user is undefined.

3.9.3.3.5 Discussion—Additional Comments

The INSERT.indication primitive is used to deliver IN_SDUs to the Insert Service user identified by the Physical Channel Name.

3.9.3.4 Directive.request

3.9.3.4.1 Function

At the sending end, the authorized user shall pass a Directive.request primitive to the service provider to invoke a Directive defined in reference [9].

3.9.3.4.2 Semantics

The Directive.request primitive shall provide parameters as follows:

Directive.request	(GVCID, Directive ID, Directive Type, Directive Qualifier)
-------------------	---

3.9.3.4.3 When Generated

The authorized user shall generate a Directive.request primitive when execution of a Directive is required.

3.9.3.4.4 Effect On Receipt

Receipt of the Directive.request primitive shall cause the service provider to execute the Directive.

3.9.3.5 Directive_Notify.indication

3.9.3.5.1 Function

At the sending end, the service provider shall pass a Directive_Notify.indication primitive to the authorized user to notify the user of an event or an action associated with a Directive requested by the user.

3.9.3.5.2 Semantics

The Directive_Notify.indication primitive shall provide parameters as follows:

Directive_Notify.indication (GVCID,
Directive ID,
Notification Type)

3.9.3.5.3 When Generated

The sending-end service provider shall generate a Directive_Notify.indication primitive in response to an event or action associated with a Directive.

3.9.3.6 Async_Notify.indication

3.9.3.6.1 Function

At the sending end, the service provider shall pass an Async_Notify.indication primitive to the authorized user to notify the user of an event that occurs asynchronously with requests.

3.9.3.6.2 Semantics

The Async_Notify.indication primitive shall provide parameters as follows:

Async_Notify.indication (GVCID,
Notification Type,
Notification Qualifier)

3.9.3.6.3 When Generated

The sending-end service provider shall generate an Async_Notify.indication primitive in response to an event that occurs asynchronously with requests.

3.10 COPS MANAGEMENT SERVICE

3.10.1 OVERVIEW

The COPS Management Service is used by a user at the sending end for managing the operations of either COP-1 or COP-P for a particular Virtual Channel or Port ID. The user manages the operations of COP-1 by invoking Directives defined in reference [9] or reference [10] for COP-P. The user is notified by the service provider of events associated with Directives and events that occur asynchronously with Directives.

A user of this service must be authorized to manage COP-1 for a particular Virtual Channel or COP-P for a particular Port ID. Only one user, identified either with the GVCID of the Virtual Channel for COP-1 or with a assigned Port ID for COP-P, is allowed to use this service.

3.10.2 COPS MANAGEMENT SERVICE PARAMETERS

3.10.2.1 General

The parameters used by the COPS Management Service primitives shall conform to the specifications of 3.10.2.2 through 3.10.2.7.

3.10.2.2 GVCID/Port ID

3.10.2.2.1 The GVCID parameter shall contain the GVCID of the Virtual Channel for which the COP-1 is managed.

3.10.2.2.2 The Port ID parameter shall contain the Port Id for which the COP-P is managed.

NOTE – The GVCID consists of an MCID and a VCID and is the SAP address of the COP Management Service.

3.10.2.3 Directive ID

The Directive ID parameter shall contain a user-supplied sequence number to be used to identify the associated Directive.request primitive in subsequent Directive_Notify.indication primitives.

3.10.2.4 Directive Type

The Directive Type parameter shall contain the type of Directive.

NOTE – The values taken by this parameter are defined in reference [9] for COP-1 and reference [10] for COP-P.

3.10.2.5 Directive Qualifier

The Directive Qualifier parameter shall contain a qualifier of the Directive if one is required. The values taken by this parameter are defined in reference [9] for COP-1 and reference [10] for COP-P.

3.10.2.6 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with a Directive.

NOTE – The values taken by this parameter are defined in reference [9] for COP-1 and reference [10] for COP-P.

3.10.2.7 Notification Qualifier

The Notification Qualifier parameter shall contain a qualifier of the notification if one is required.

NOTE – The values taken by this parameter are defined in reference [9] for COP-1 and reference [10] for COP-P.

3.10.3 COPS MANAGEMENT SERVICE PRIMITIVES

3.10.3.1 General

The service primitives associated with the COPS Management Service are:

- a) Directive.request;
- b) Directive_Notify.indication;
- c) Async_Notify.indication.

3.10.3.2 Directive.request

3.10.3.2.1 Function

At the sending end, the authorized user shall pass a Directive.request primitive to the service provider to invoke a Directive defined in reference [9] for COP-1 and reference [10] for COP-P.

3.10.3.2.2 Semantics

The Directive.request primitive shall provide parameters as follows:

Directive.request	(GVCID/Port ID, Directive ID, Directive Type, Directive Qualifier)
-------------------	---

3.10.3.2.3 When Generated

The authorized user shall generate a Directive.request primitive when execution of a Directive is required.

3.10.3.2.4 Effect On Receipt

Receipt of the Directive.request primitive shall cause the service provider to execute the Directive.

3.10.3.3 Directive_Notify.indication

3.10.3.3.1 Function

At the sending end, the service provider shall pass a Directive_Notify.indication primitive to the authorized user to notify the user of an event or an action associated with a Directive requested by the user.

3.10.3.3.2 Semantics

The Directive_Notify.indication primitive shall provide parameters as follows:

Directive_Notify.indication (GVCID/Port ID,
Directive ID,
Notification Type)

3.10.3.3.3 When Generated

The sending-end service provider shall generate a Directive_Notify.indication primitive in response to an event or action associated with a Directive.

3.10.3.3.4 Effect On Receipt

The effect of receipt of the Directive_Notify.indication primitive by the COPs Management Service user is undefined.

3.10.3.4 Async_Notify.indication

3.10.3.4.1 Function

At the sending end, the service provider shall pass an Async_Notify.indication primitive to the authorized user to notify the user of an event that occurs asynchronously with requests.

3.10.3.4.2 Semantics

The Async_Notify.indication primitive shall provide parameters as follows:

Async_Notify.indication	(GVCID/Port ID, Notification Type, Notification Qualifier)
-------------------------	--

3.10.3.4.3 When Generated

The sending-end service provider shall generate an Async_Notify.indication primitive in response to an event that occurs asynchronously with requests.

3.10.3.4.4 Effect On Receipt

The effect of receipt of the Async_Notify.indication primitive by the COPs Management Service user is undefined.

4 **PROTOCOL SPECIFICATION WITHOUT SDLS OPTION**

NOTE – This section specifies the protocol data unit and the procedures of USLP without support for the SDLS protocol. Section 6 specifies the protocol with the SDLS option.

4.1 **PROTOCOL DATA UNIT**

4.1.1 **USLP TRANSFER FRAME**

A USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

- a) Transfer Frame Primary Header (4 to 14 octets, mandatory);
- b) Transfer Frame Insert Zone (integral number of octets, optional);
- c) Transfer Frame Data Field (integral number of octets, mandatory);
- d) Operational Control Field (4 octets, optional);
- e) Frame Error Control Field (2 octets, optional).

NOTES

- 1 The protocol data unit of USLP is the USLP Transfer Frame. In this Recommended Standard, the USLP Transfer Frame is also called the Transfer Frame or Frame for simplicity.
- 2 The Transfer Frame Primary Header can be limited in size to 4 octets if the End of Transfer Frame Primary Header Flag is set to ‘1’ (see 4.1.2.6 and the USLP Green Book reference [D11] for the motivation of this exceptional case).
- 3 The Frame Error Control Field is controlled by Managed Parameters (see section 5).
- 4 The structural components of the USLP Transfer Frame are shown in figure 4-1.

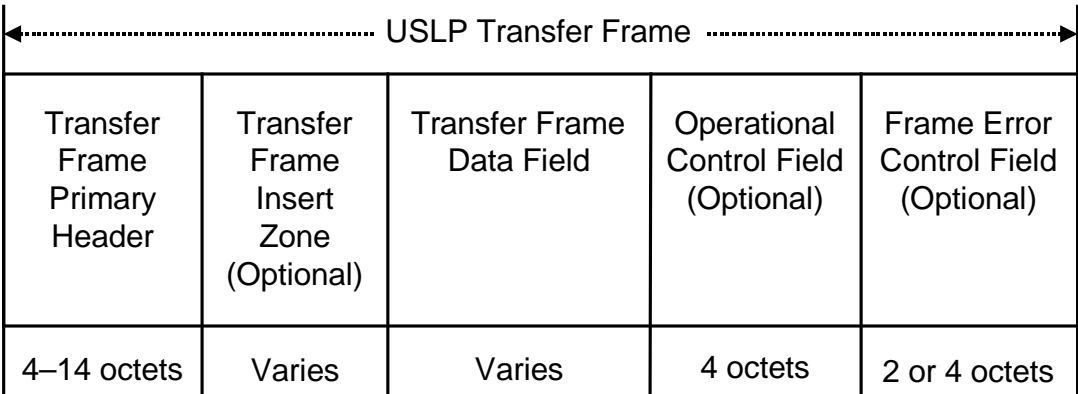


Figure 4-1: USLP Transfer Frame Structural Components

4.1.2 TRANSFER FRAME PRIMARY HEADER

4.1.2.1 General

The Transfer Frame Primary Header is mandatory and shall consist of up to 12 fields, positioned contiguously, in the following sequence:

- a) Transfer Frame Version Number (4 bits, mandatory);
- b) Spacecraft Identifier (16 bits, mandatory);
- c) Source or Destination Identifier (1 bit, mandatory);
- d) Virtual Channel Identifier (6 bits, mandatory);
- e) Multiplexer Access Point (MAP) Identifier (4 bits, mandatory);
- f) End of Transfer Frame Primary Header Flag (1 bit, mandatory);
- g) Frame Length (16 bits, mandatory);
- h) Bypass/Sequence Control Flag (1 bit, mandatory);
- i) Reserve Spares (3 bits, mandatory);
- j) Operational Control Field Flag (1 bit, mandatory);
- k) Virtual Channel Frame Count Length (3 bits, mandatory);
- l) Virtual Channel Frame Count (0 to 7 octets, optional).

The format of the Transfer Frame Primary Header is shown in figure 4-2.

Master Channel ID											
Transfer Frame Version Number	Spacecraft ID	Source or Destination ID	Virtual Channel ID	MAP ID	End of Transfer Frame Primary Header Flag	Frame Length	Bypass/Sequence Control Flag	Reserve Spares	Operational Control Field Flag	VC Frame Count Length	VC Frame Count
4 bits	16 bits	1 bit	6 bits	4 bits	1 bit	16 bits	1 bit	3 bits	1 bit	3 bits	0–56 bits

Figure 4-2: Transfer Frame Primary Header

4.1.2.2 Master Channel Identifier

4.1.2.2.1 General

4.1.2.2.1.1 Bits 0–19 of the Transfer Frame Primary Header shall contain the Master Channel Identifier (MCID).

4.1.2.2.1.2 The Master Channel Identifier shall consist of:

- a) Transfer Frame Version Number (4 bits, mandatory);
- b) Spacecraft Identifier (16 bits, mandatory).

4.1.2.2.2 Transfer Frame Version Number

4.1.2.2.2.1 Bits 0–3 of the Transfer Frame Primary Header shall contain the (Binary Encoded) Transfer Frame Version Number.

4.1.2.2.2.2 This 4-bit field shall identify the data unit as a Transfer Frame defined by this Recommended Standard; it shall be set to ‘1100’ binary.

4.1.2.2.3 Spacecraft Identifier

4.1.2.2.3.1 Bits 4–19 of the Transfer Frame Primary Header shall contain the Spacecraft Identifier (SCID).

4.1.2.2.3.2 The Spacecraft Identifier is assigned by CCSDS and shall provide the identification of the spacecraft that is associated with the data contained in the Transfer Frame.

4.1.2.2.3.3 The Spacecraft Identifier shall be static throughout all Mission Phases.

NOTE – The Secretariat of the CCSDS assigns Spacecraft Identifiers according to the procedures in reference [11].

4.1.2.3 Source-or-Destination Identifier

4.1.2.3.1 Bit 20 of the Transfer Frame Primary Header shall contain the Source-or-Destination Identifier.

4.1.2.3.2 The Source-or-Destination Identifier shall be used to identify the association of the data contained in the Transfer Frame.

4.1.2.3.3 The Source-or-Destination Identifier is interpreted as follows:

- a) ‘0’ = SCID refers to the *source* of the Transfer Frame;
- b) ‘1’ = SCID refers to the *destination* of the Transfer Frame.

NOTE – Although not limited to Proximity environments, a discussion of the assignment of source vs. destination Spacecraft ID is provided in reference [D14].

4.1.2.4 Virtual Channel Identifier

4.1.2.4.1 Bits 21–26 of the Transfer Frame Primary Header shall contain the Virtual Channel Identifier (VCID).

4.1.2.4.2 The Virtual Channel Identifier shall be used to identify the Virtual Channel.

NOTES

- 1 VCID 63 is used for transmission of Only Idle Data (OID) Transfer Frames (see 4.1.4.1.7).
- 2 There are no restrictions on the selection of any other Virtual Channel Identifiers except the rules described above. In particular, Virtual Channels are not required to be numbered consecutively.
- 3 A Transfer Frame on the ‘Idle’ Virtual Channel may not contain any valid user data within its Transfer Frame Data Zone, but it may contain an Insert Zone (managed) and/or an OCF if signaled within the Frame Primary Header.

4.1.2.5 Multiplexer Access Point Identifier

4.1.2.5.1 Bits 27–30 of the Transfer Frame Primary Header shall contain the Multiplexer Access Point Identifier (MAP ID).

4.1.2.5.2 If only one MAP exists, then the MAP ID shall be set to a constant value for all data placed into the Transfer Frame Data Zone for that VC.

NOTES

- 1 The MAP Identifier provides the identification from 1 to up to 16 possible MAP Channels within a Virtual Channel.
- 2 There are no restrictions on the selection of MAP Identifiers. In particular, MAPs are not required to be numbered consecutively.
- 3 MAPs identify SAPs for the Virtual Channel and may be designated to receive Sequence Controlled or Expedited SDUs.

4.1.2.6 End of Transfer Frame Primary Header Flag

4.1.2.6.1 Bit 31 of the Transfer Frame Primary Header shall contain the End of Transfer Frame Primary Header Flag.

4.1.2.6.2 The End of Transfer Frame Primary Header Flag shall signal the use of a truncated Transfer Frame Primary Header. It shall be ‘1’ if the Transfer Frame Primary

Header is limited to 4 octets; it shall be '0' if the Transfer Frame Primary Header is not truncated.

NOTE – The truncated Transfer Frame Primary Header is useful for supporting mission control commands, e.g., emergency hardware commands. Together with the minimum size of the Transfer Frame Data Field (TFDF) Header of 1 octet, this limits the protocol overhead to 5 octets.

4.1.2.7 Frame Length

4.1.2.7.1 Bits 32–47 of the Transfer Frame Primary Header shall contain the Frame Length.

4.1.2.7.2 This 16-bit field shall contain a length count C which equals one fewer than the total octets in the Transfer Frame.

4.1.2.7.3 The count shall be measured from the first bit of the Transfer Frame Primary Header to the last bit of the Frame Error Control Field (if present), or to the last bit of the Operational Control Field (if present), or to the last bit of the Transfer Frame Data Field (if both the FECF and the OCF is omitted).

NOTES

1 The length count C is expressed as:

$$C = (\text{Total Number of Octets in the Transfer Frame}) - 1$$

2 The size of this field limits the total number of octets in the Transfer Frame to 65536 octets.

4.1.2.8 Bypass/Sequence Control Flag

4.1.2.8.1 Bit 48 of the Transfer Frame Primary Header shall contain the Bypass/Sequence Control Flag.

4.1.2.8.2 The single-bit Bypass/Sequence Control Flag shall be used to control the application of Frame Acceptance Checks within the COP-1 reference [9] and COP-P reference [10] procedures by the receiving end:

- a) setting the Bypass/Sequence Control Flag to value '0' shall specify that this Transfer Frame is a Sequence Controlled Transfer Frame, and acceptance of this Transfer Frame by the receiving end shall be subject to the normal Frame Acceptance Checks of the FARM;
- b) setting the Bypass/Sequence Control Flag to value '1' shall specify that this Transfer Frame is an Expedited Transfer Frame, and the Frame Acceptance Checks of the FARM by the receiving end shall be bypassed/expedited.

NOTES

- 1 The Frame Acceptance and Reporting Mechanism (FARM) associated with the COP-1 or COP-P can be made to operate in a normal Acceptance/Sequence controlled mode (for Sequence Controlled Transfer Frames) or a Expedited mode (for Expedited Transfer Frames), according to the setting of the Bypass/Sequence Control Flag.
- 2 All Transfer Frames received by the receiving end first undergo a basic standard set of Frame Validation Checks, which are applied regardless of the setting of the Bypass/Sequence Control Flag (see 4.3.9.9).
- 3 The equivalent of the Telecommand protocol Control Flag required by the COP is defined in USLP within the Protocol ID field contained in the Transfer Frame Data Field Header (see 4.1.4.2).

4.1.2.9 Reserve Spares

4.1.2.9.1 Bits 49-51 of the Transfer Frame Primary Header shall contain the reserved spares.

4.1.2.9.2 This 3-bit field is reserved for future definition by CCSDS and shall be set to '000'.

4.1.2.10 Operational Control Field Flag

4.1.2.10.1 Bit 52 of the Transfer Frame Primary Header shall contain the Operational Control Field Flag.

4.1.2.10.2 The Operational Control Field Flag shall indicate the presence or absence of the Operational Control Field. It shall be '1' if the Operational Control Field is present; it shall be '0' if the Operational Control Field is not present.

4.1.2.11 Virtual Channel Frame Count Length

4.1.2.11.1 Bits 53-55 of the Transfer Frame Primary Header shall contain the length of the Virtual Channel Frame Count field.

4.1.2.11.2 This 3-bit field shall define the length of the Virtual Channel Frame Count field.

4.1.2.11.3 The value of this field shall be interpreted as shown in table 4-1.

Table 4-1: Interpretation of the Virtual Channel Frame Count Length

Value of VC Frame Count Length (binary)	Length of Virtual Channel Frame Count Field	Maximum Virtual Channel Frame Count
000	0	N/A
001	1 octet	255
010	2 octets	65,535
011	3 octets	16,777,215
100	4 octets	4,294,967,295
101	5 octets	1,099,511,627,775
110	6 octets	281,474,976,710,655
111	7 octets	7.20575940379279E16

NOTES

- 1 The VC Frame Count Length of '000' indicates that the VC Frame Count field is absent for the Transfer Frame Header.
- 2 The Frame Count Length (i.e., Sequence Number) enables the FARM to check the sequentiality of incoming Sequence-Controlled Transfer Frames. The Frame Count Length is Virtual Channel-dependent; i.e., this protocol maintains a separate Frame Count for each of the Virtual Channels.
- 3 A VC Frame Count Length of '000' indicates that the VC Frame Count field is absent from the Transfer Frame Header.

4.1.2.12 Virtual Channel Frame Count

4.1.2.12.1 If present, the Virtual Channel Frame Count shall be the final field in the Transfer Frame Header.

4.1.2.12.2 The VC Frame Count shall contain a sequential binary count (modulo maximum Virtual Channel Frame Count +1) of each Sequence-Controlled Transfer Frame transmitted within a specific Virtual Channel (see table 4-2).

4.1.2.12.3 The VC Frame Count shall increment monotonically and independently for the set of Sequence-Controlled Frames; i.e., the Bypass/Sequence Control Flag is set to '0'. In this case, the Frame Count is called the Sequence-Controlled Counter.

4.1.2.12.4 The VC Frame Count shall increment monotonically and independently for the set of Expedited frames; i.e., the Bypass/Sequence Control Flag is set to '1'. In this case, the Frame Count is called the FARM-B (Expedited Frame) Counter.

4.1.2.12.5 A resetting of the Virtual Channel Frame Count before reaching the maximum Virtual Channel Frame Count shall not take place unless it is unavoidable.

NOTES

- 1 The purpose of this field is to provide individual accountability for each Virtual Channel. It enables both systematic Packet extraction from the Transfer Frame Data Field for Expedited Frames as well as the verification of the sequentiality of incoming Sequence Controlled Frames by the FARM.
- 2 The procedure for assigning the VC Frame Count (i.e., Frame Sequence Number) to Transfer Frames for COP-1 is defined in reference [9] and for COP-P in reference [10].
- 3 If the Virtual Channel Frame Count is reset because of an unavoidable re-initialization, the completeness of a sequence of Transfer Frames in the related Virtual Channel cannot be determined.
- 4 The FARM-B (Expedited Frame) Counter is not used in the frame validation process but is required for correlations associated with Proximity-1 timing services (reference [15]).

4.1.3 TRANSFER FRAME INSERT ZONE

4.1.3.1 If included, the Transfer Frame Insert Zone shall follow, without gap, the Transfer Frame Primary Header.

4.1.3.2 If the Physical Channel supports the Insert Service for transfer of isochronous data, then the Insert Zone shall exist in every Transfer Frame transmitted within the same Physical Channel, including OID Transfer Frames.

4.1.3.3 For Insert Service, the length of the Insert Zone shall be set by management to be equal to the constant length of the Insert Service Data Unit (IN_SDU) for that Physical Channel. The Insert Zone shall contain precisely one octet-aligned IN_SDU.

4.1.3.4 Once set by management, the length of the Insert Zone shall be static throughout a Mission Phase.

NOTE — The presence or absence of the Insert Zone is controlled by Managed Parameters.

4.1.4 TRANSFER FRAME DATA FIELD

4.1.4.1 General

4.1.4.1.1 If present, the Transfer Frame Data Field (TFDF) shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone, if present.

4.1.4.1.2 The Transfer Frame Data Field, which shall contain an integer number of octets, shall have a length which varies and is equal to the Frame Length minus the length of the Transfer Frame Primary Header minus the length of the Transfer Frame Insert Zone and/or the Operational Control Field and/or Frame Error Control Field (if any of these are present).

4.1.4.1.3 The Transfer Frame Data Field shall consist of a mandatory TFDF Header followed by an optional Transfer Frame Data Zone (TFDZ) (see figure 4-3 below).

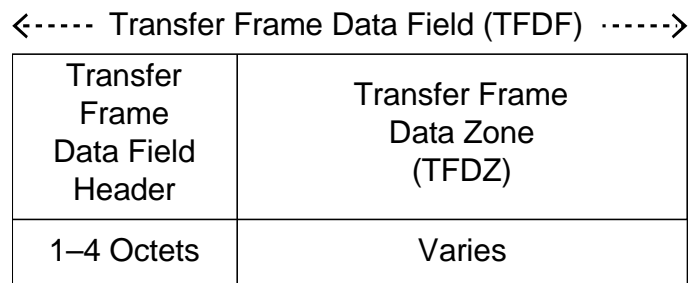


Figure 4-3: Transfer Frame Data Field

NOTE – The length of the TFDZ associated with a VC may be restricted by the associated VC Managed Parameters to be either fixed or variable (see 5.4).

4.1.4.1.4 The Transfer Frame Data Field, which shall contain an integral number of octets, may vary in length up to a maximum of 65514 octets.

4.1.4.1.5 The Transfer Frame Data Field shall contain any of the following SDUs determined by management: either an integral number of octets of user data or an integral number of octets of Protocol Directives as defined in 4.1.4.2.1.4.3.

4.1.4.1.6 In the case where no valid Transfer Frame Data Field is available for transmission at release time for a Transfer Frame, a Transfer Frame with a Data Field containing Only Idle Data shall be transmitted.

NOTE – Such a Transfer Frame is called an OID (Only Idle Data in its Data Field) Transfer Frame.

4.1.4.1.7 The Virtual Channel ID of an OID Transfer Frame shall be set to the value of ‘all ones’ (i.e., 63 decimal) and a project-specified ‘idle’ pattern shall be inserted into the Transfer Frame Data Field.

4.1.4.1.8 It is required that a separate Virtual Channel be dedicated to carry OID Transfer Frames.

NOTES

- 1 OID Transfer Frames are not sent on Virtual Channels that also contain valid Packets.
- 2 Transfer Frames containing Only Idle Data in their Data Fields are sent to maintain synchronization at the receiver and also to transmit data in the Transfer Frame Insert Zone when there is no Data Field to send.
- 3 Idle Data in the Transfer Frame Data Field of an OID Transfer Frame is not to be confused with the Idle Packet specified in reference [12].
- 4 The idle pattern used in the OID Transfer Frame is project-specific, but a random pattern is preferred. Problems with the reception of frames have been encountered because of insufficient randomization.

4.1.4.2 Transfer Frame Data Field Header

4.1.4.2.1 General

4.1.4.2.1.1 The Transfer Frame Data Field Header shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone if present.

4.1.4.2.1.2 The TFDF Header shall contain the following fields:

- a) Transfer Frame Data Zone (TFDZ) Construction Rules (3 bits, mandatory);
- b) USLP Protocol Identifier (5 bits, mandatory);
- c) USLP Protocol Identifier Extension (8 bits, optional);
- d) Pointer Field (First Header Pointer or Last Valid Octet Pointer) (16 bits, optional).

NOTE – The format of the TFDF Header is shown in figure 4-4.

Transfer Frame Data Field Header			
TFDZ Construction Rules 3 bits	USLP Protocol Identifier 5 bits	USLP Protocol Identifier Extension 8 bits	Pointer Field (First Header Pointer or Last Valid Octet Pointer) 16 bits

Figure 4-4: Transfer Frame Data Field Header

4.1.4.2.1.3 TFDZ Construction Rules

4.1.4.2.1.3.1 Bits 0–2 of the Transfer Frame Data Field Header shall contain the TFDZ Construction Rules.

4.1.4.2.1.3.2 The TFDZ Construction Rules shall be used to identify how the protocol organizes the user data within the TFDZ in order to transport it.

4.1.4.2.1.3.3 A MAP_SDU and/or a single Packet SDU may be segmented with portions thereof placed within the TFDZ of successive frames within the same MAP ID.

4.1.4.2.1.3.4 In all cases, the SDU must begin in the first octet of the TFDZ contained in the first frame carrying that SDU.

4.1.4.2.1.3.5 The TFDZ Construction Rules shall be interpreted as specified in 4.1.4.2.1.3.6–4.1.4.2.1.3.13.

4.1.4.2.1.3.6 TFDZ Construction Rule ‘000’ shall indicate a fixed-length TFDZ whose contents are CCSDS Packets that span Transfer Frame boundaries. The First Header Pointer is required; its function is to enable the packet extraction process to restart whenever a Transfer Frame is lost on that VC.

4.1.4.2.1.3.7 TFDZ Construction Rule ‘001’ shall indicate a fixed-length TFDZ that contains either a portion of or a complete MAP_SDU that starts in the first octet of the TFDZ. The Last Valid Octet Pointer shall be set to binary ‘all ones’ if the complete MAP_SDU is not contained within the TFDZ. Otherwise the value in the Last Valid Octet Pointer shall be the delimiter of the MAP_SDU.

Table 4-2: Summary of the TFDZ Construction Rules

TFDZ Construction Rules	Interpretation
000	Spanning Data Units (a fixed-length TFDZ that contains CCSDS Packets that span one or more frames). The TFDZ Header contains the First Header Pointer.
001	The MAPA_SDU contained in this fixed-length TFDZ starts in the first octet of this TFDZ. The Last Valid Octet Pointer contains all binary ones if the MAP_SDU is not fully contained in this TFDZ.
010	This fixed-length TFDZ contains a segment of a MAP_SDU started in a prior TFDZ of this GMAP. The Last Valid Octet Pointer contains all binary 'ones' if this MAP_SDU is not completed in this TFDZ.
011	Octet Stream (User defined octet-aligned) Data contained in a variable-length TFDZ.
100	Starting Segment (i.e., MAP_SDU or CCSDS Packet) starts in the first octet but does not end in this variable-length TFDZ).
101	Continuation Segment (i.e., continuing portion of a MAP_SDU or CCSDS Packet contained in this variable-length TFDZ).
110	Ending Segment (i.e., This variable-length TFDZ contains the final segment of a MAP_SDU or CCSDS Packet that started in a previous TFDZ).
111	No Segmentation (one complete MAP_SDU or multiple CCSDS Packets are contained in this variable-length TFDZ).

NOTE – TFDZ Construction Rules '000' or '001' or '010' apply to fixed-length TFDZs. '011', '100', '101', '110' and '111' apply to variable-length TFDZs.

4.1.4.2.1.3.8 The TFDZ Construction Rule '010' shall indicate a fixed-length TFDZ containing a portion of a MAPA_SDU that was started in a previous TFDZ. The Last Valid Octet Pointer is required to delimit the end of the MAP_SDU and shall contain binary 'all ones' if the end of the MAP_SDU is not contained in this TFDZ.

NOTE – The MAP_SDU begins in the first octet of a TFDZ (Rule '001'), and portions of that MAP_SDU are contained in the TFDZs of the following Transfer Frames of that VC (Rule '010'), with the last frame completing the MAP_SDU (Rule '010').

4.1.4.2.1.3.9 The TFDZ Construction Rule '011' shall indicate a variable-length TFDZ containing an octet-aligned stream that is intended to be continuous without beginning or end.

NOTE – An example is a Video Stream.

4.1.4.2.1.3.10 The TFDZ Construction Rule '100' shall indicate a variable-length TFDZ containing the starting segment of a service data unit (either a MAP_SDU or CCSDS Packet).

NOTE – The SDU is segmented and starts but does not end in this TFDZ.

4.1.4.2.1.3.11 The TFDZ Construction Rule ‘101’ shall indicate a variable-length TFDZ containing a continuation segment (i.e., continuing portion of an SDU contained in the previous TFDZ in that VC).

4.1.4.2.1.3.12 The TFDZ Construction Rule ‘110’ shall indicate a variable-length TFDZ containing the ending segment of the SDU being transferred within this VC.

4.1.4.2.1.3.13 The TFDZ Construction Rule ‘111’ shall indicate a variable-length TFDZ that is not segmented, containing either one MAP_SDU or one or more complete CCSDS Packets.

4.1.4.2.1.4 USLP Protocol Identifier within USLP Data Zone

4.1.4.2.1.4.1 Bits 3–7 of the Transfer Frame Data Field Header shall contain the USLP Protocol Identifier (UPID).

4.1.4.2.1.4.2 The USLP Protocol Identifier shall identify the CCSDS recognized protocol of the data contained within the TFDZ.

4.1.4.2.1.4.3 The USLP Protocol Identifier shall be interpreted as follows:¹

- a) The value ‘00000’ in the Protocol ID field signals that COP-1 directives are contained within the TFDZ.
- b) The value ‘00001’ in the Protocol ID field signals that COP-P directives are contained within the TFDZ.
- c) The value ‘00010’ in the Protocol ID field signals that SDLS directives are contained within the TFDZ.
- d) The value ‘00011’ in the Protocol ID field signals that user-defined stream data are contained within the TFDZ.
- e) The value ‘00100’ in the Protocol ID field signals that mission-unique directives are contained within the TFDZ.
- f) The value ‘10101’ in the Protocol ID field signals that the entire TFDZ is filled with idle octets.
- g) The value ‘11111’ in the Protocol ID field signals that the 8-bit Protocol Identifier Extension field is used for protocol identification. This extension provides for the optional inclusion of an additional 256 Protocol IDs that can be assigned.

¹ When this Recommended Standard is finalized, the USLP Protocol IDs will be registered and maintained within SANA.

4.1.4.2.1.5 USLP Protocol Identifier Extension within USLP Data Zone

4.1.4.2.1.5.1 If present, the USLP Protocol Identifier Extension field shall be 8 bits in length and shall follow, without gap, the USLP Protocol Identifier field.

4.1.4.2.1.5.2 The USLP Protocol Identifier Extension shall identify the CCSDS recognized protocol of the Protocol Data Unit (PDU) contained within the TFDZ.

4.1.4.2.1.5.3 If the USLP Protocol Identifier field contains the value ‘11111’, then the USLP Protocol Identifier Extension field shall be used to identify the protocol whose PDU is contained within the TFDZ.²

4.1.4.2.1.6 Pointer Field (First Header Pointer or Last Valid Octet Pointer)

4.1.4.2.1.6.1 If present, the Pointer Field shall follow, without gap, the Protocol Identifier Extension field and be 16 bits in length and shall contain the offset to a specific octet within the TFDZ.

4.1.4.2.1.6.2 The Pointer Field shall be used only for TFDZ Construction Rules ‘000’, ‘001’, and ‘010’.

NOTE – The first octet in the TFDZ has an offset of ‘0’.

4.1.4.2.1.6.3 When the value in the TFDZ construction rule is ‘000’ binary, the Pointer Field shall contain the offset within the TFDZ to the first octet of the first packet header that starts within the TFDZ. In this case, the Pointer Field is designated as the First Header Pointer.

4.1.4.2.1.6.4 When the value in the TFDZ construction rule is ‘000’ binary, and when no packet starts in the TFDZ, then the First Header Pointer shall be set to binary ‘all ones’.

NOTE – The above situation can occur if a long PDU extends across more than one Transfer Frame.

4.1.4.2.1.6.5 When the value in the TFDZ construction rule is ‘000’ binary, and if the TFDZ contains Only Idle Data, then the First Header Pointer shall be set to binary ‘all ones minus one’.

NOTES

- 1 The purpose of the First Header Pointer is to facilitate delimiting of variable-length packets contained within the TFDZ, by pointing directly to the location of the first packet from which its length may be determined.

² When this Recommended Standard is finalized, the extended Protocol IDs will be registered and maintained within SANA.

- 2 The locations of any subsequent packets within the same TFDZ will be determined by calculating the locations using the length field of these packets.
- 3 If the last packet in the TFDZ of Transfer Frame *M* spills over into Frame *N* of the same Virtual Channel (where $N=M+1$), then the First Header Pointer in Frame *N* points to the start of the packet within Frame *N*.
- 4 If the last packet in the TFDZ of Transfer Frame *M* spills over into Frame *N* of the same Virtual Channel but Frame *N* is missing, then the First Header Pointer in the next Frame received in that VC ignores the residue of the split packet and points to the start of the next packet that begins in that frame.

4.1.4.2.1.6.6 When the value in the TFDZ Construction Rules is binary ‘001’ or ‘010’, the Pointer field shall contain the offset to the last octet of the MAP_SDU being transferred, with the remaining octets being fill octets.

NOTE – In this case the Pointer field is designated as the Last Valid Octet Pointer.

4.1.4.2.1.6.7 If the MAP_SDU does not complete within this fixed-length TFDZ then the value contained within the Last Valid Octet Pointer shall be set to binary ‘all ones’.

NOTE – If the length of the TFDZ is fixed and the MAP_SDU has an insufficient number of user data octets remaining to fill the TFDZ, then the Pointer field indicates the location of the last valid user data octet within the TFDZ (i.e., the boundary between user data and any inserted fill data).

4.1.4.2.1.6.8 When the value in the TFDZ construction rule is ‘010’ binary, and if there are no valid user data in the TFDZ (i.e., the TFDZ contains Only Idle Data), the Last Valid Octet Pointer shall be set to the value ‘all ones minus one’.

4.1.4.3 Transfer Frame Data Zone (TFDZ)

4.1.4.3.1 The Transfer Frame Data Zone shall follow, without gap, the TFDZ Header.

4.1.4.3.2 The data contained within any given TFDZ shall be associated with one and only one MAP ID and Protocol ID.

4.1.4.3.3 The TFDZ shall contain either CCSDS Packets, MAP_SDUs, Octet Stream, or Idle Data (a project-specified ‘idle’ pattern).

NOTES

- 1 The idle pattern used in the TFDZ is project-specific and can be fixed or variable length. A random pattern is preferred. Problems with the reception of frames have been encountered because of insufficient randomization.
- 2 Idle data is used only with fixed-length TFDZs.

- 3 The first and last packet of the TFDZ are not necessarily complete, since the first packet may be a continuation of a packet begun in the previous TFDZ, and the last packet may continue in the subsequent TFDZ of the same Virtual Channel.

4.1.4.3.4 When a fixed-length TFDZ is partially completed with Packets when the required release time for a Transfer Frame of a Virtual Channel has been reached, an Encapsulation Idle Packet (reference [13]) shall be added to fill the remainder of the TFDZ.

NOTES

- 1 This event can occur based on the timeliness criteria contained in the managed parameters.
- 2 TFDZs that contain only Idle Data are sent to maintain synchronous transmission of fixed-length Transfer Frames.
- 3 A TFDZ that contains only Idle Data is not to be confused with the OID Transfer Frame defined in 4.1.4.1.6.
- 4 A TFDZ that contains only Idle Data can be generated whenever it is necessary (even in the middle of transmission of a packet that is split into multiple TFDZs).
- 5 Idle Data in the TFDZ is not to be confused with the Idle Packet specified in reference [12].

4.1.5 OPERATIONAL CONTROL FIELD

4.1.5.1 If present, the Operational Control Field shall occupy the four octets following, without gap, the Transfer Frame Data Field.

4.1.5.2 The Operational Control Field is optional; its presence or absence shall be signaled by the Operational Control Field Flag for each Virtual Channel.

4.1.5.3 Bit 0 of the Operational Control Field shall contain a Type Flag with the following meanings:

- a) Type Flag '0': the Operational Control Field holds a Type-1-Report which contains either a Communications Link Control Word (CLCW) (see F1) or a Proximity Link Control Word (PLCW) (see F2.3.2);
- b) Type Flag '1': the Operational Control Field holds a Type-2-Report.

NOTE – The Type Flag can vary between Transfer Frames on the same Virtual Channel that carries this field.

4.1.5.4 In a Type-2 Report, bit 1 of the Operational Control Field shall indicate the use of this report as follows:

- a) if this bit is '0', the contents of the report are project-specific;
- b) if this bit is '1', the contents of the report contain an SDLS Common Link Security Report (CLSR) defined in reference [17].

NOTES

- 1 The Operational Control Field is provided to support the reporting mechanism of the COP-1/COP-P and/or the SDLS protocol.
- 2 In Type-2 Reports, the value of bit 1 of the Operational Control Field may vary between Transfer Frames on the same Virtual Channel that carries this field.
- 3 The purpose of this field is to provide a standardized mechanism for reporting a small number of real-time functions (such as retransmission control or spacecraft clock calibration); currently the use for retransmission control (Type-1 Reports) has been defined by CCSDS in reference [D6].

4.1.6 FRAME ERROR CONTROL FIELD

4.1.6.1 General

4.1.6.1.1 If present, the Frame Error Control Field shall occupy two or four octets following, without gap, the Operational Control Field, if present, or the Transfer Frame Data Field, if an Operational Control Field is not present.

4.1.6.1.2 The Frame Error Control Field is optional; its presence or absence shall be established by management.

4.1.6.1.3 If present, the Frame Error Control Field shall occur within every Transfer Frame transmitted within the same Physical Channel throughout a Mission Phase.

NOTES

- 1 The purpose of this field is to provide a capability for detecting errors that may have been introduced into the Transfer Frame during the transmission and data handling process.
- 2 Whether this field should be used on a particular Physical Channel is determined based on the mission requirements for data quality and the selected options for the underlying Synchronization and Channel Coding Sublayer. This field may be mandatory depending on the selected options for the Synchronization and Channel Coding Sublayer as specified in references [3], [6], or [7], or by management for other reasons besides coding.

- 3 CCSDS provides two coding procedure options: 16-bit or 32-bit. The CRC-16 procedure provides an undetected bit error rate of approximately 10^{-5} compared to approximately 10^{-11} for the CRC-32 procedure (see annex B for their definition).

4.2 PROTOCOL PROCEDURES AT THE SENDING END

4.2.1 OVERVIEW

This subsection describes procedures at the sending end associated with each of the functions shown in figure 4-5. In this figure, data flow from top to bottom of the figure. This figure identifies data-handling functions performed by the protocol entity at the sending end, and shows logical relationships among these functions. This figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all of the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.

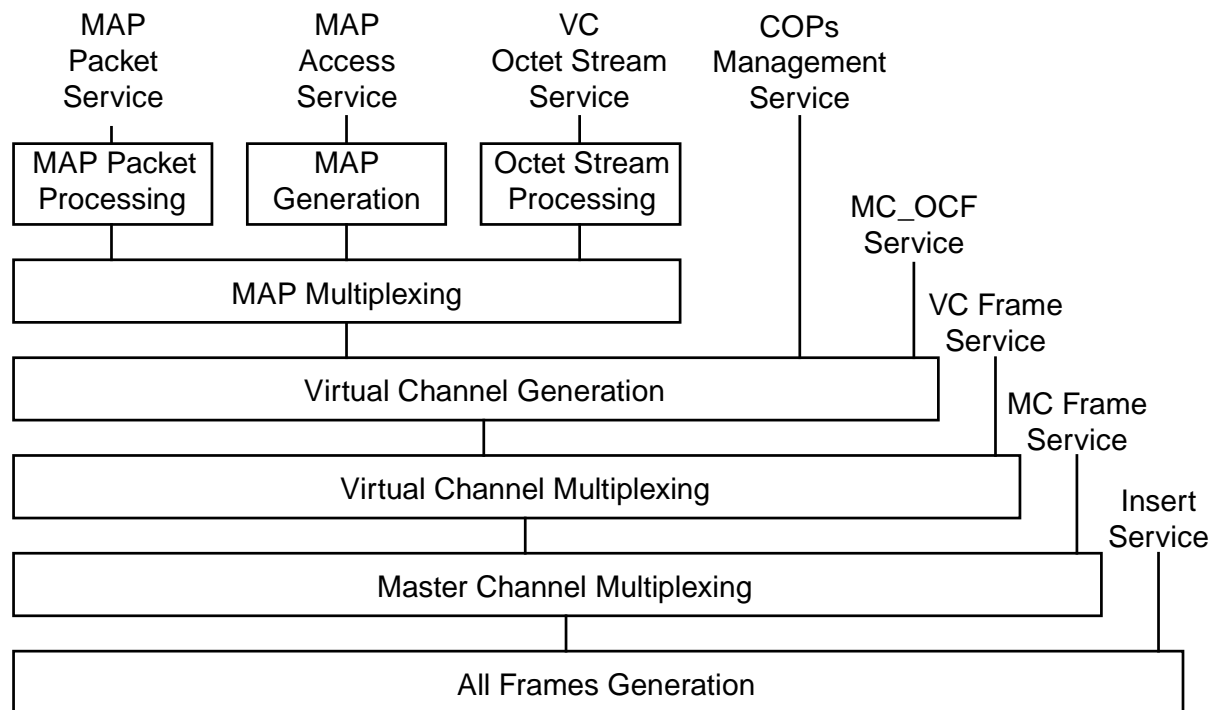


Figure 4-5: Internal Organization of Protocol Entity (Sending End)

4.2.2 MAP PACKET PROCESSING FUNCTION

4.2.2.1 MAP Packet Processing for Fixed-Length TFDZ

4.2.2.1.1 The MAP Packet Processing Function for a fixed-length TFDZ shall be used to transfer variable-length Packets in fixed-length TFDZs of Transfer Frames.

NOTES

- 1 When a packet spans a Transfer Frame, the associated TFDZ Construction Rule used is '000'.
- 2 There is an instance of the Packet Processing Function for each MAP Channel that carries Packets.

4.2.2.1.2 The fixed-length TFDZ that carries Packets shall be constructed by concatenating Packets together until the maximum TFDZ length is exceeded. The Packet whose contents which exceeds the maximum TFDZ length shall be split, filling the TFDZ completely, and the remainder of the Packet shall be placed in a new TFDZ on the same MAP Channel. Construction of the next and the following TFDZs shall continue with the concatenation of Packets until each TFDZ overflows.

4.2.2.1.3 If Packets of multiple versions are to be transferred on a MAP Channel, Packets of these versions shall be multiplexed into a contiguous string of Packets before constructing TFDZs.

NOTE – The associated TFDZ Construction Rule used is '111'.

4.2.2.1.4 The 'First Header Pointer' field shall be set to indicate the location of the first octet of the first Packet occurring within the TFDZ.

4.2.2.1.5 In the absence of sufficient Packets supplied from the users at release time, one Encapsulation Idle Packet of appropriate length shall be inserted.

NOTE – An Encapsulation Idle Packet is defined by reference [13]. The shortest Encapsulation Idle Packet defined is one octet in length (i.e., a one-octet header).

4.2.2.1.6 If it is necessary, the Packet Processing Function may generate an 'idle' TFDZ by setting the First Header Pointer to 'all ones minus one'.

NOTE – An abstract model of the MAP Packet Processing Function for fixed TFDZs is illustrated in figure 4-6.

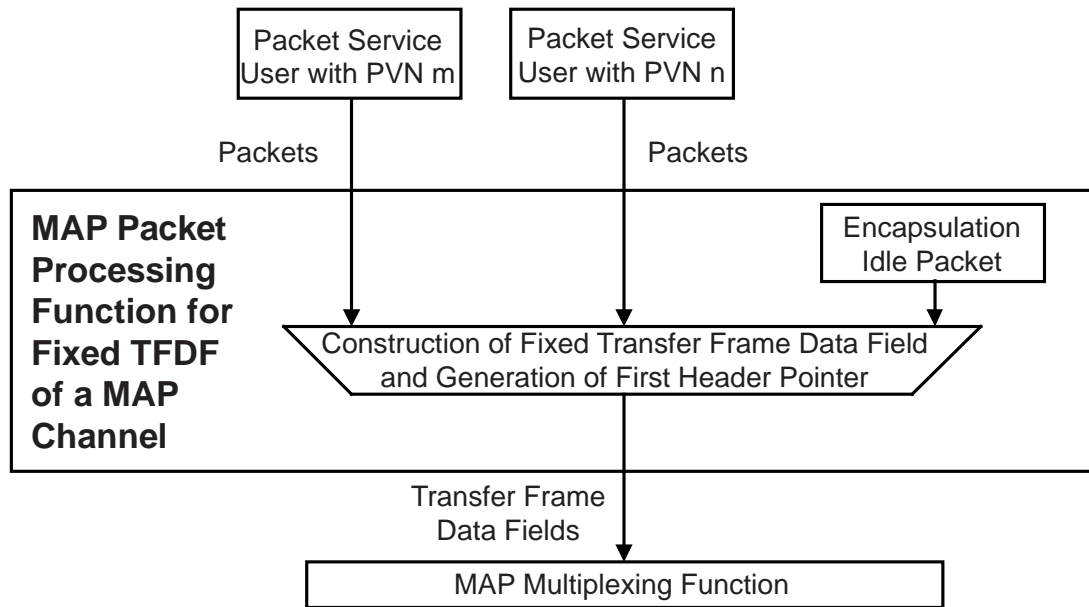


Figure 4-6: Abstract Model of Packet Processing Function for Fixed TFDZs

4.2.2.2 MAP Packet Processing for Variable-Length TFDZ

4.2.2.2.1 The MAP Packet Processing Function shall be used to transfer variable-length Packets in the variable-length Data Zone of Transfer Frames of a MAP Channel.

NOTE – There is an instance of the Packet Processing Function for each MAP Channel that carries Packets.

4.2.2.2.2 If the Packet to be transferred exceeds the maximum transmission unit size of the TFDZ, the MAP Packet Processing Function shall divide it into portions that are compatible with insertion into the TFDZ and attach a TFDZ Header to each portion, forming a TFDZ.

4.2.2.2.3 The first octet of the Packet shall appear, without gap, after the TFDZ Header, in the first octet of the TFDZ.

NOTE – The associated TFDZ Construction Rule used is ‘100’.

4.2.2.2.4 The TFDZs containing the first and continuing portions of the Packet may each have a length equal to the maximum allowable length of the TFDZ on that particular MAP Channel.

NOTE – The associated TFDZ Construction Rule used for continuing segments is ‘101’.

4.2.2.2.5 The TFDZ containing the last portion of the Packet shall contain the TFDZ Header and the remainder of the Packet.

NOTE – The associated TFDZ Construction Rule used is ‘110’.

4.2.2.2.6 The portions of a Packet shall be transferred in consecutive Transfer Frames of the MAP Channel without being interlaced with any other Packets or portions in the same MAP Channel.

4.2.2.2.7 If blocking of Packets is permitted on a particular MAP Channel, then:

- a) multiple complete Packets may be placed into a TFDF with a single TFDF Header preceding them;
- b) the blocked Packets plus the TFDF Header must fit within the maximum size TFDF permitted for the MAP Channel.

NOTE – The associated TFDF Construction Rule used is ‘111’.

4.2.2.2.8 If Packets of multiple versions are to be transferred on a MAP Channel, Packets of these versions are multiplexed into a contiguous string of Packets before they are placed in the TFDF.

NOTE – An abstract model of the MAP Packet Processing Function for variable TFDFs is illustrated in figure 4-7.

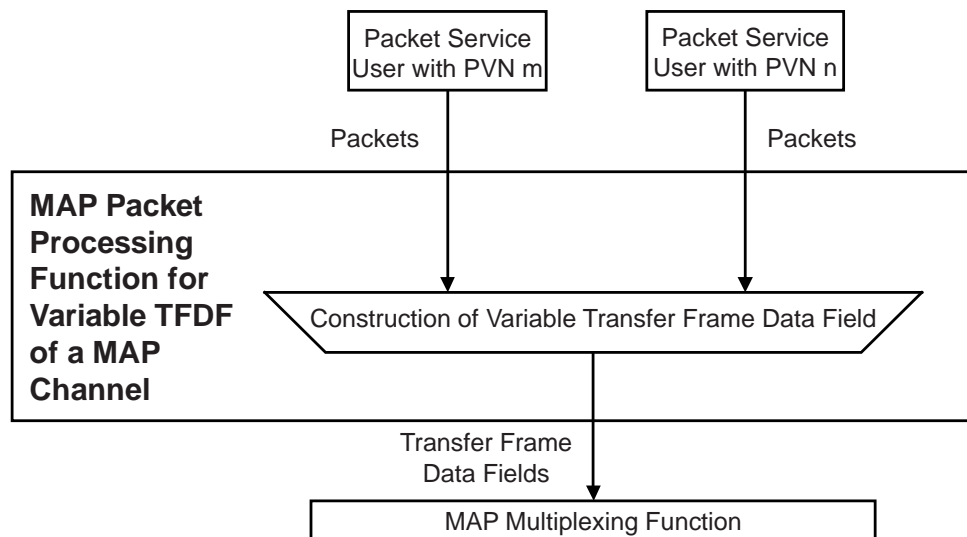


Figure 4-7: Abstract Model of MAP Packet Processing Function for a Variable-Length TFDF

4.2.3 MAP GENERATION FUNCTION

4.2.3.1 MAP Generation Function for Fixed-Length TFDZ

4.2.3.2 The MAP Generation Function shall be used to transfer variable-length user-defined service data units (MAP_SDUs) in fixed-length TFDZs of Transfer Frames of a MAP Channel.

NOTE – There is an instance of the MAP Generation Function for each MAP Channel that carries MAP_SDUs.

4.2.3.3 If the MAP_SDU to be transferred exceeds the maximum transmission unit size of the TFDZ, the MAP Generation Function shall divide it into portions that are compatible with insertion into the TFDZ and generate a Last Valid Octet Pointer, forming the TFDF.

4.2.3.4 The first octet of the MAP_SDU shall appear, without gap, in the first octet of the TFDZ.

NOTE – The associated TFDF Construction Rule used is ‘001’.

4.2.3.5 If a MAP_SDU did not complete in a previous TFDZ then the next portion of that MAP_SDU shall start in the first octet of the TFDZ of the following frame in that VC in the same MAP Channel.

4.2.3.6 The Last Valid Octet Pointer shall point to the last octet of the MAP_SDU if it completes within the TFDZ.

4.2.3.7 The Last Valid Octet Pointer shall point to the last octet of the MAP_SDU if it completes within the TFDZ. If the MAPA_SDU does not complete within the TFDZ then the contents of the Last Valid Octet Pointer shall be set to binary ‘all ones’.

NOTES

- 1 There can be multiple instances of the MAP Generation Function for each Virtual Channel identified by different MAPs. This is possible because a TFDZ that contains a MAPA_SDU can only contain a single MAPA_SDU thus satisfying the one MAP and one UPID requirement.
- 2 An abstract model of the MAP Generation Function for fixed-length TFDFs is illustrated in figure 4-8.

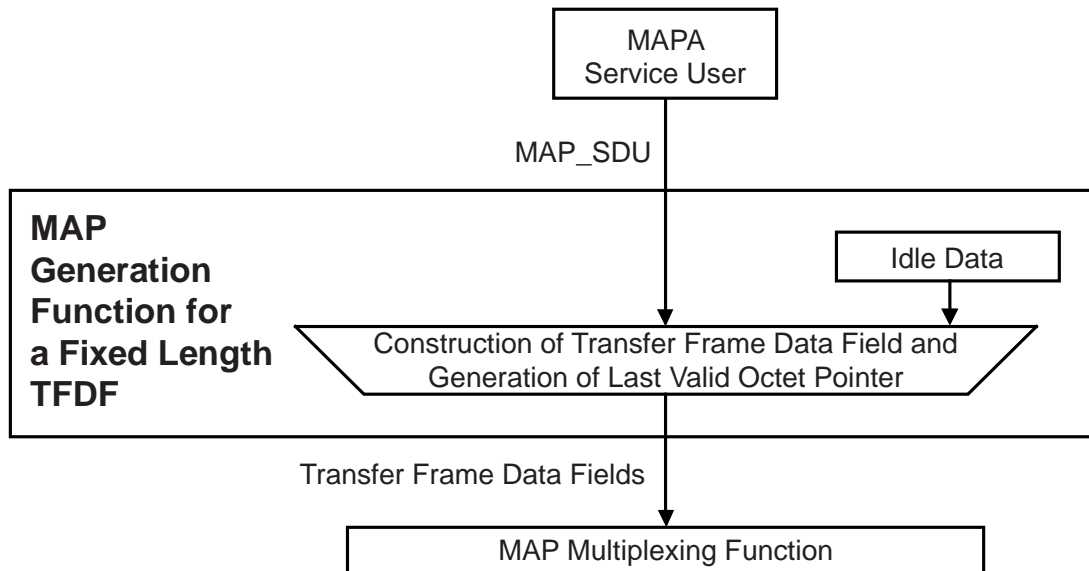


Figure 4-8: Abstract Model of MAP Generation Function for Fixed-Length TFDF

4.2.3.8 MAP Generation Function for a Variable-Length TFDZ

4.2.3.8.1 The MAP Generation Function shall be used to transfer variable-length user-defined service data units (MAP_SDUs) in variable-length Data Zones of Transfer Frames of a MAP Channel.

NOTE – There is an instance of the MAP Generation Function for each MAP Channel that carries MAP_SDUs.

4.2.3.8.2 If the MAP_SDUs to be transferred exceeds the maximum transmission unit size of the TFDZ, the MAP Generation Function shall divide it into portions that are compatible with insertion into the TFDZ and attach a TFDF Header to each portion, forming a TFDF.

4.2.3.8.3 The first octet of the MAP_SDUs shall appear, without gap, after the TFDF Header, in the first octet of the TFDZ.

NOTE – The associated TFDF Construction Rule used is ‘100’.

4.2.3.8.4 The TFDFs containing the first and continuing portions of the MAP_SDUs may each have a length equal to the maximum allowable length of the TFDF on that particular MAP Channel.

NOTE – The associated TFDF Construction Rule used for continuing segments is ‘101’.

4.2.3.8.5 The TFDF containing the last portion of the MAP_SDUs shall contain the TFDF Header and the remainder of the Packet.

NOTE – The associated TFDF Construction Rule used is ‘110’.

4.2.3.8.6 The portions of a MAP_SDU shall be transferred in consecutive Transfer Frames of the MAP Channel without being interlaced with any other Packets or portions in the same MAP Channel.

NOTE – An abstract model of the MAP Generation Function for variable-length TFDZs is illustrated in figure 4-9.

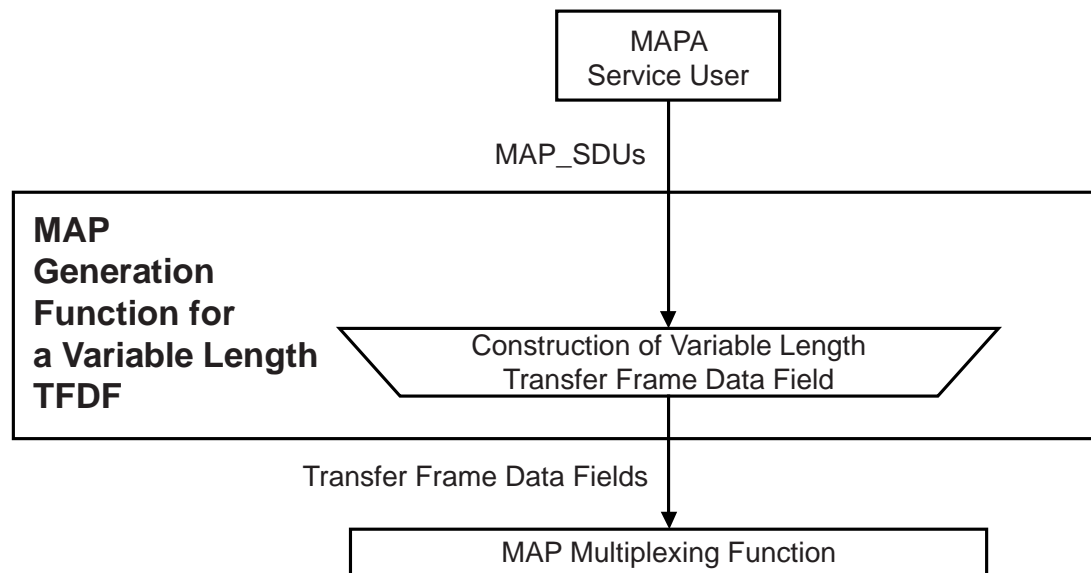


Figure 4-9: Abstract Model of MAP Generation Function for Variable-Length TFDZ

4.2.4 MAP OCTET STREAM PROCESSING FUNCTION

4.2.4.1 The MAP Octet Stream Processing Function shall be used to transfer variable-length user-defined Octet Stream Data in the variable-length TFDZ of Transfer Frames.

NOTE – There is an instance of the Octet Stream Processing Function for each MAP Channel that carries Octet Stream Data.

4.2.4.2 The MAP Octet Stream Processing Function shall be used to fill the TFDZ with the Octet Stream Data supplied by the user.

4.2.4.3 Each octet shall be placed sequentially, and unchanged, into the TFDZ.

4.2.4.4 When the Octet Stream Data have filled one particular TFDZ, the continuation of the Octet Stream Data shall be placed in the next TFDZ on the same MAP Channel.

NOTES

- 1 The associated TFDZ Construction Rule used is '011'.

- 2 The release of an Octet Stream TFDF may be subject to release timing constraints contained in the managed parameters.
- 3 An abstract model of the MAP Octet Stream Processing Function is illustrated in figure 4-10.

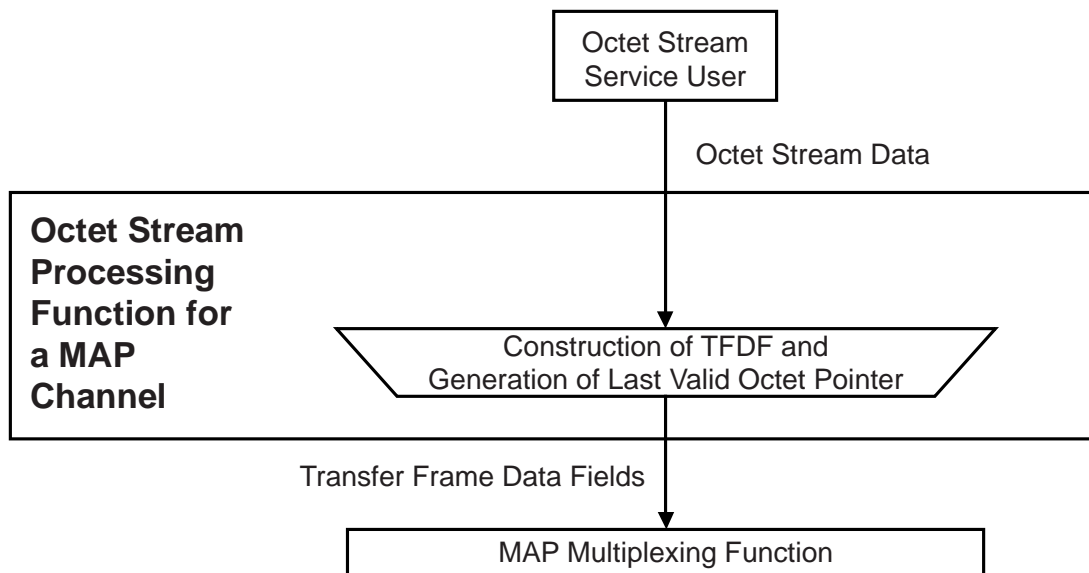


Figure 4-10: Abstract Model of the MAP Octet Stream Processing Function

4.2.5 MAP MULTIPLEXING FUNCTION

4.2.5.1 The MAP Multiplexing Function shall be used to multiplex the contents of the Transfer Frame Data Zone of different MAP Channels of a Virtual Channel.

NOTE – There can be up to 16 instances of the MAP Multiplexing Function for each Virtual Channel.

4.2.5.2 The MAP Multiplexing Function shall multiplex TFDFs as long as their contents is associated with a single MAP ID and a single UPID. These TFDFs shall be put into a queue in an appropriate order set by management.

4.2.5.3 The algorithm to be used to order the service data units is not specified by CCSDS, but shall be defined by project organizations considering factors such as priority, release rate, isochronous timing requirements, etc.

NOTE – An abstract model of the MAP Multiplexing Function is illustrated in figure 4-11.

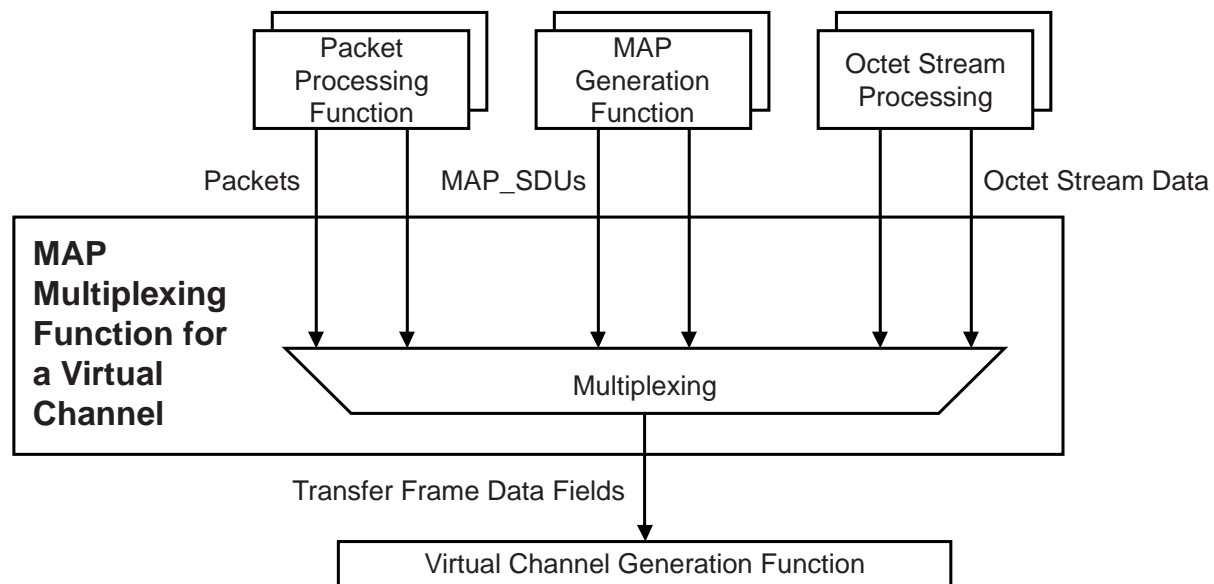


Figure 4-11: Abstract Model of MAP Multiplexing Function

4.2.6 VIRTUAL CHANNEL GENERATION FUNCTION

NOTE – The Virtual Channel Generation Function is used to build the basic structure of Transfer Frames. It is also used to build the structure and the Primary Header of the Transfer Frames for transmission on each Virtual Channel. It also performs most of the operations required to move service data units reliably from the sending end to the receiving end. There is an instance of the Virtual Channel Generation Function for each Virtual Channel.

4.2.6.1 The Virtual Channel Generation Function shall perform the following three procedures in the following order:

- a) Maintain, increment, and include the VC frame count in the Transfer Frame Primary Header based upon the contents of the Bypass/Sequence Control Flag field. This function also maintains the VC Frame Count Length field.
- b) The Frame Operation Procedure (FOP), which is a sub-procedure of the Communications Operation Procedure (COP); and
- c) the Frame Generation Procedure in this order.

4.2.6.2 The FOP shall accept SDUs from the Packet Processing Function, Octet Stream Processing Function, or a MAPA Service User, and shall control transmission and retransmission of frames by examining the report contained in the CLCWs and generating Control Commands.

4.2.6.3 The FOP shall also accept Directives from a COP Management Service User.

NOTE – The detailed specification of the FOP are given in references [9] and [10].

4.2.6.4 The Frame Generation Procedure shall generate Transfer Frames by attaching a Transfer Frame Primary Header to each TFDZ or SPDU delivered by the FOP.

4.2.6.5 Transfer Frames shall be assembled by placing a single TFDZ unchanged into the Transfer Frame Data Field and generating both the Transfer Frame Primary and TFDZ Header fields. A Virtual Channel Frame Count shall be generated independently for each Virtual Channel and placed into the Primary Header.

4.2.6.6 If there is a user of the MC_OCF Service for a particular Virtual Channel, an OCF_SDU supplied by the user shall be placed in the Operational Control Field.

4.2.6.7 The Insert Zone and the Frame Error Control Field of Transfer Frames, if present for a particular Physical Channel, shall be kept empty by the Virtual Channel Generation Function.

NOTE – An abstract model of the Virtual Channel Generation Function is illustrated in figure 4-12.

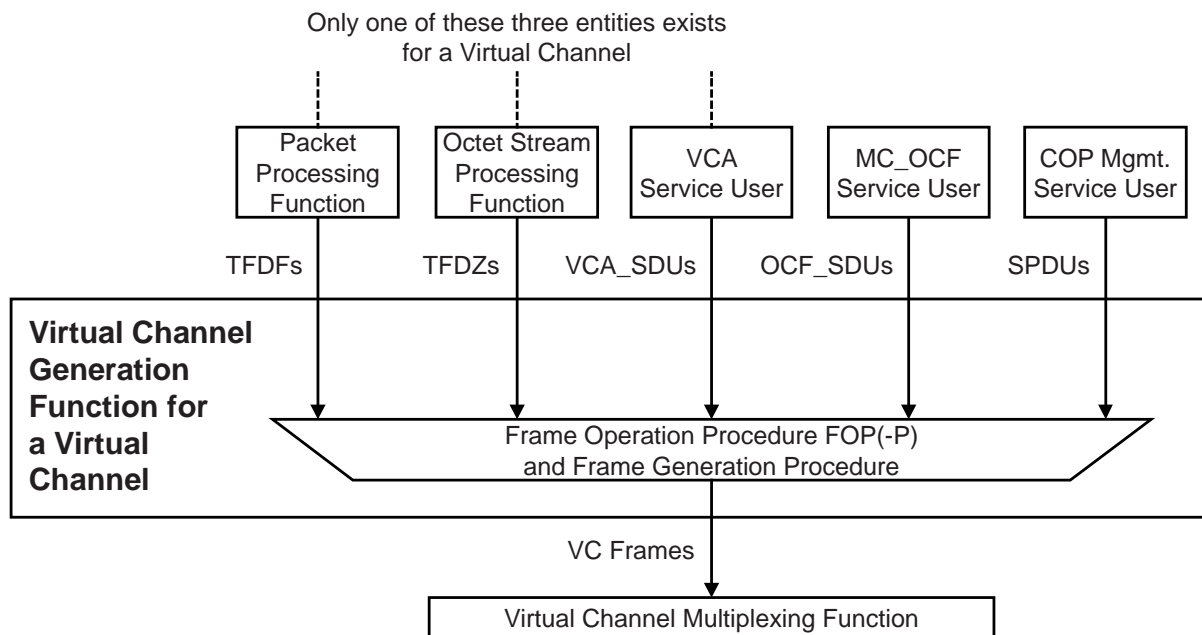


Figure 4-12: Abstract Model of Virtual Channel Generation Function

4.2.7 VIRTUAL CHANNEL MULTIPLEXING FUNCTION

4.2.7.1 The Virtual Channel Multiplexing Function shall be used to multiplex Transfer Frames of different Virtual Channels of a Master Channel.

NOTE – There is an instance of the Virtual Channel Multiplexing Function for each Master Channel that has multiple Virtual Channels.

4.2.7.2 The Virtual Channel Multiplexing Function shall multiplex Transfer Frames received from the instances of the Virtual Channel Generation Function and, if present, the Virtual Channel Frame Service users, in an appropriate order that is set by management.

NOTE – The Virtual Channel Multiplexing Function can put the multiplexed Transfer Frames into a queue.

4.2.7.3 The algorithm used to order the Transfer Frames is not specified by CCSDS, but shall be defined by project organizations considering factors such as priority, release rate, isochronous timing requirements, etc.

4.2.7.4 If there is only one Master Channel on the Physical Channel and when any of the coding schemes defined in references [3], [4], and [5] are used, then the Virtual Channel Multiplexing Function shall create an OID Transfer Frame to preserve the continuity of the transmitted stream in the event that there are no valid Transfer Frames available for transmission at a release time.

NOTE – When any of the coding schemes defined in references [6] and [7] are used, then the Virtual Channel Multiplexing Function does not create an OID Transfer Frame.

4.2.7.5 The OID Transfer Frame shall have its VCID set to the reserved value of ‘all ones’. It is not required to maintain a Virtual Channel Frame Count for OID Transfer Frames.

NOTE – An abstract model of the Virtual Channel Multiplexing Function is illustrated in figure 4-13.

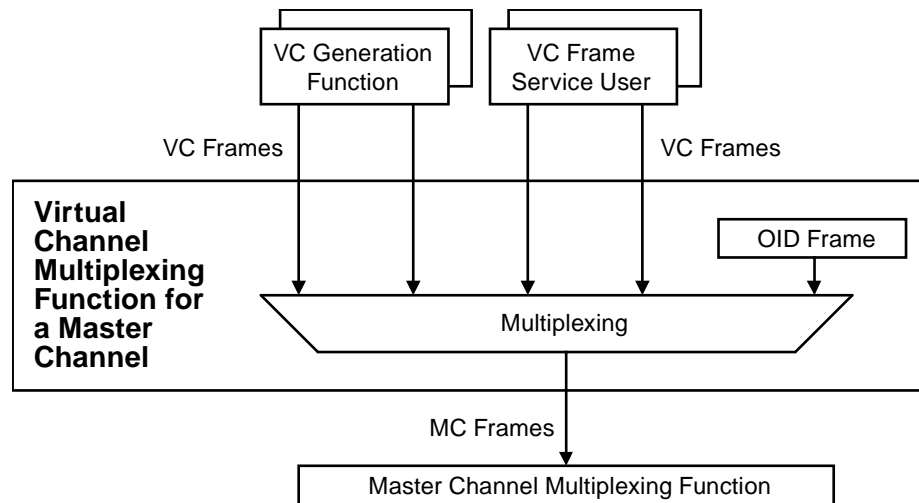


Figure 4-13: Abstract Model of Virtual Channel Multiplexing Function

4.2.8 MASTER CHANNEL MULTIPLEXING FUNCTION

4.2.8.1 The Master Channel Multiplexing Function shall be used to multiplex Transfer Frames of different Master Channels of a Physical Channel.

NOTE – There is an instance of the Master Channel Multiplexing Function for each Physical Channel that has multiple Master Channels.

4.2.8.2 The Master Channel Multiplexing Function shall multiplex Transfer Frames received from the instances of the Virtual Channel Multiplexing Function and, if present, the Master Channel Frame Service users, in an appropriate order that is set by management.

NOTE – The Master Channel Multiplexing Function can put the multiplexed Transfer Frames into a queue.

4.2.8.3 The algorithm to be used to order the Transfer Frames is not specified by CCSDS, but shall be defined by project organizations considering factors such as priority, release rate, isochronous timing requirements, etc.

4.2.8.4 When any of the coding schemes defined in references [3], [4], and [5] are used, the Master Channel Multiplexing Function shall create an OID Transfer Frame to preserve the continuity of the transmitted stream in the event that there are no valid Transfer Frames available for transmission at a release time. The OID Transfer Frame shall have its VCID set to the reserved value of ‘all ones’ and its MCID set to one of the allowable values.

NOTES

- 1 It is not required to maintain a Virtual Channel Frame Count for OID Transfer Frames.

- 2 When any of the coding schemes defined in references [6] and [7] are used, then the Master Channel Multiplexing Function does not create an OID Transfer Frame.
- 3 An abstract model of the Master Channel Multiplexing Function is illustrated in figure 4-14.

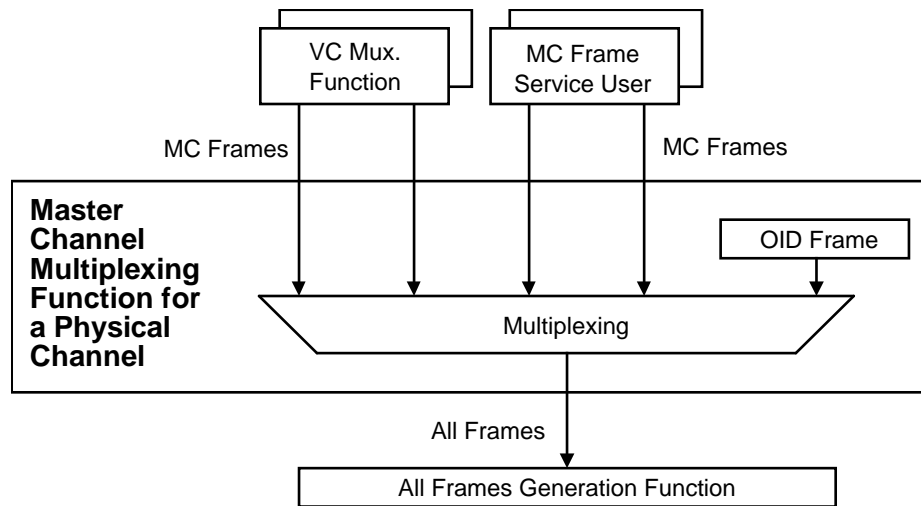


Figure 4-14: Abstract Model of Master Channel Multiplexing Function

4.2.9 ALL FRAMES GENERATION FUNCTION

4.2.9.1 The All Frames Generation Function shall be used to place Insert service data units into Transfer Frames of a Physical Channel. It shall also be used to perform error control encoding defined by this Recommended Standard.

NOTE – There is an instance of the All Frames Generation Function for each Physical Channel.

4.2.9.2 If the optional Insert Service is activated, a fixed-length Insert Zone shall exist in every Transfer Frame that is transmitted in a particular Physical Channel. The IN_SDUs shall be timed to arrive at a constant interval that corresponds to the release time of the Transfer Frames onto the Physical Channel. The All Frames Generation Function shall place the IN_SDUs, received from the Insert Service user, into the Insert Zone of the Transfer Frames, preserving octet alignment.

4.2.9.3 If the Frame Error Control Field is present, check bits shall be generated using the encoding procedure described in annex B and inserted into the Transfer Frame Trailer.

4.2.9.4 If the Frame Error Control Field is present, it must be present in all the Transfer Frames transmitted in a particular Physical Channel.

4.2.9.5 Externally generated Transfer Frames associated with the Virtual Channel Frame and Master Channel Frame Services shall always bypass the error control encoding functions specified above.

NOTE – The users of these services therefore need to ensure that the Transfer Frames contain an error control option which conforms with that used by the service provider for the same Physical Channel.

4.2.9.6 The All Frames Generation Function shall deliver data units to the underlying Synchronization and Channel Coding Sublayer.

4.2.9.7 Each data unit delivered by the All Frames Generation Function shall consist of one Transfer Frame.

NOTE – USLP does not use the feature in TC Synchronization and Channel Coding, i.e., reference [6] that allows a CLTU to contain multiple Transfer Frames.

4.2.9.8 When reference [6] is used as the Synchronization and Channel Coding Sublayer, the All Frames Generation Function may request the Synchronization and Channel Coding Sublayer to perform systematic retransmissions of a data unit as described in 2.4.2, unless the data unit contains a frame carrying service data on the Expedited Service.

NOTES

- 1 When systematic retransmissions of a data unit are requested, the additional delay for the retransmissions can be taken into account when deciding the delivery time for the following data unit.
- 2 An abstract model of the All Frames Generation Function is illustrated in figure 4-15.

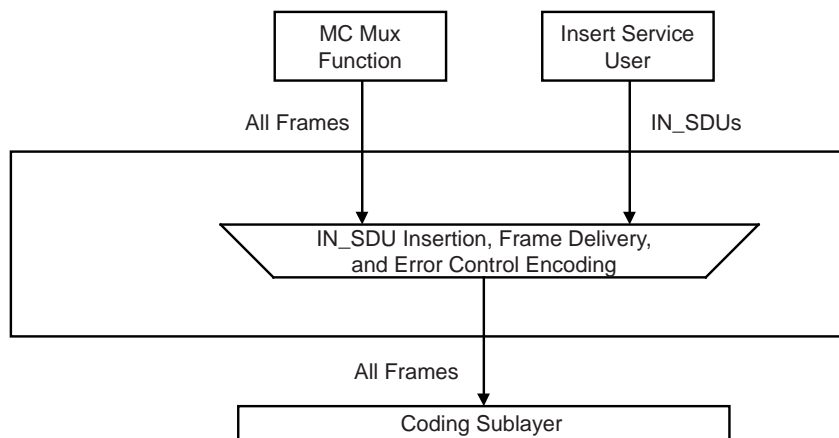


Figure 4-15: Abstract Model of All Frames Generation Function

4.3 PROTOCOL PROCEDURES AT THE RECEIVING END

4.3.1 OVERVIEW

This subsection describes procedures at the receiving end associated with each of the functions shown in figure 4-16. In this figure, data flow from bottom to top of the figure. This figure identifies data-handling functions performed by the protocol entity at the receiving end, and shows logical relationships among these functions. This figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all of the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.

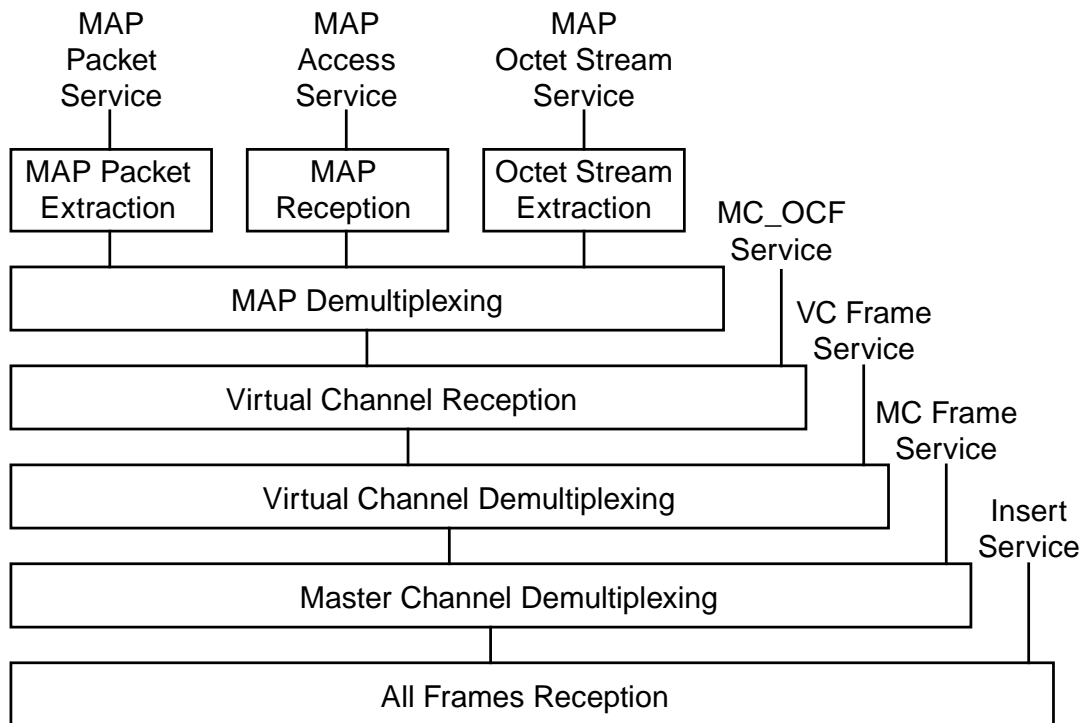


Figure 4-16: Internal Organization of Protocol Entity (Receiving End)

4.3.2 MAP PACKET EXTRACTION FUNCTION

4.3.2.1 Overview

The extraction of Packets from fixed-length TFDFs is different than from variable-length TFDFs. The construction rules identify the method used to load the Packets in the TFDF and the value in the Construction Rules field identify the set of procedures that shall be used to extract the packets.

NOTE – There is an instance of the Packet Extraction Function for each MAP Channel that carries Packets.

4.3.2.2 MAP Packet Extraction Function for a Fixed-Length TFDZ

NOTE – The MAP Packet Extraction Function used to extract variable-length Packets from the fixed-length TFDZs is associated with TFDF Construction Rule ‘000’ when a packet spans multiple frames.

4.3.2.2.1 The Packet Extraction Function shall extract Packets from TFDFs received from the Virtual Channel Reception Function.

4.3.2.2.2 The First Header Pointer of each TFDF shall be used in conjunction with the length field of each Packet contained within the TFDZ to provide the delimiting information needed to extract Packets.

NOTE – Extraction of multiple integral packets from the TFDF utilizes the TFDF Construction Rule ‘111’.

4.3.2.2.3 If the last Packet removed from the TFDF is incomplete, the Packet Extraction Function shall retrieve its remainder from the beginning of the next TFDF received on the same Virtual Channel.

4.3.2.2.4 The First Header Pointer for the next TFDF shall be used to determine the length of the remainder and, hence, the beginning of the next Packet to be extracted.

4.3.2.2.5 If the calculated location of the beginning of the first Packet is not consistent with the location indicated by the First Header Pointer, the Packet Extraction Function shall assume that the First Header Pointer is correct, and shall continue the extraction based on that assumption.

4.3.2.2.6 Extracted Packets shall be delivered to the users on the basis of their MAP ID, VCID, and their Application Processing ID contained in the packet’s header.

NOTES

- 1 Incomplete Packets are not required to be delivered in cross support situations. Idle Packets are discarded. TFDFs that contain only Idle Data are also discarded.
- 2 An abstract model of the MAP Packet Extraction Function for fixed-length TFDFs is illustrated in figure 4-17.

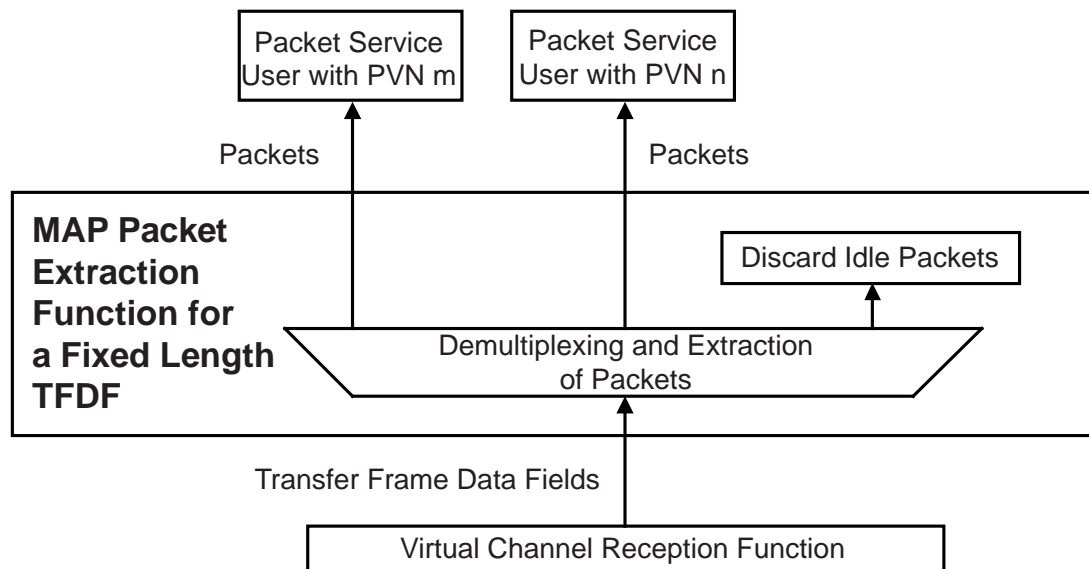


Figure 4-17: Abstract Model of MAP Packet Extraction Function for Fixed-Length TFDFs

4.3.2.3 MAP Packet Extraction Function for a Variable-Length TFDZ

NOTE – The MAP Packet Extraction Function used to extract variable-length Packets from variable-length TFDFs on a MAP Channel are associated with Construction rules ‘100’, ‘101’, ‘110’ and ‘111’.

4.3.2.3.1 The MAP Packet Extraction Function shall extract Packets from variable-length TFDFs received from the MAP Demultiplexing Function.

4.3.2.3.2 A Segmented Packet within a variable-length TFDF shall be extracted and reconstructed from the TFDF using the TFDF Construction Rules ‘100’, ‘101’, and ‘110’.

4.3.2.3.3 If blocking of Packets is permitted within a variable-length TFDF, these packets or one complete packet shall be extracted and reconstructed from the TFDF using the TFDF Construction Rule ‘111’ and the length field of each Packet.

4.3.2.3.4 Extracted Packets shall be delivered to the users on the basis of their GMAP ID and the Application ID contained in the Packet’s header.

NOTES

- 1 Incomplete Packets are not required to be delivered in cross-support situations.
- 2 An abstract model of the MAP Packet Extraction Function for a variable-length TFDZ is illustrated in figure 4-18.

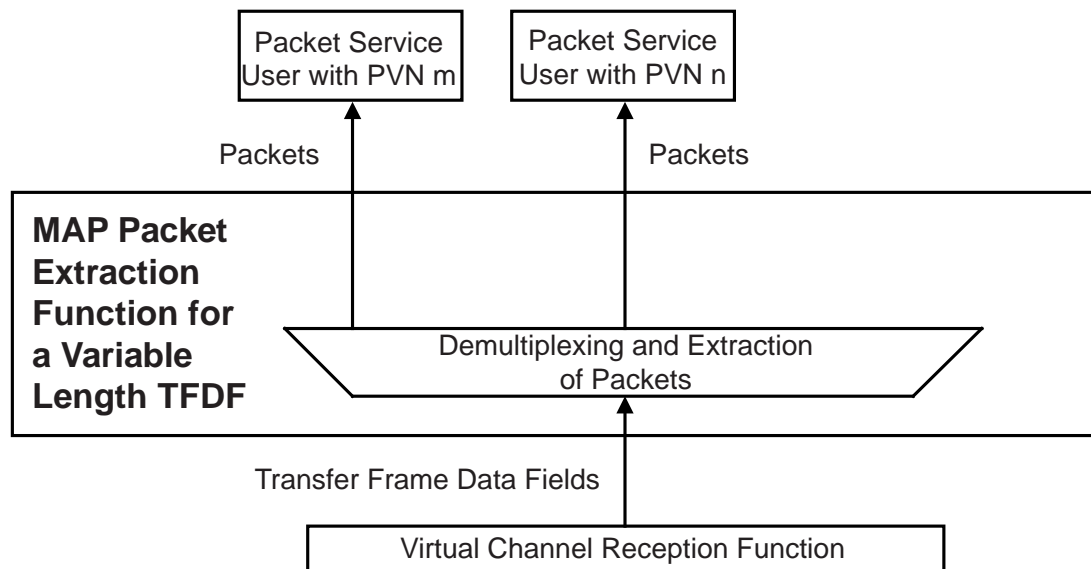


Figure 4-18: Abstract Model of MAP Packet Extraction Function for Variable-Length TFDFs

4.3.3 MAP RECEPTION FUNCTION

4.3.3.1 Overview

The extraction of MAP_SDUs from fixed-length TFDFs is different than from variable-length TFDFs. The construction rules identify the method used to load the packets in the TFDF and the value in the TFDF Construction Rules field Identify the set of procedures to be used to extract the MAP_SDUs.

There is an instance of the MAP Reception Function for each MAP Channel that carries a MAP_SDU.

4.3.3.2 MAP Reception Function for Fixed-Length TFDFs

NOTE – The MAP Reception Function used to extract variable-length MAP_SDUs from fixed-length TFDFs are associated with Construction Rules ‘001’ and ‘010’.

4.3.3.2.1 The MAP Reception Function shall extract the MAP_SDU from fixed-length TFDFs received from the MAP Demultiplexing Function.

4.3.3.2.2 The MAP_SDU shall be extracted and reconstructed from TFDFs using the TFDF Construction Rules ‘001’ and ‘010’ and the Last Valid Octet Pointer within each of the TFDF Headers.

4.3.3.2.3 The Last Valid Octet Pointer that contains a value that is not ‘all ones’ signals the end of the MAP_SDU, and any octets that are pass the pointer value till the end of the frame shall be discarded.

4.3.3.2.4 The extracted and reconstructed MAP_SDU shall be delivered to the MAPA Service user associated with its VCID and MAP ID.

NOTE – An abstract model of the MAP Reception Function for fixed-length TFDFs is illustrated in figure 4-19.

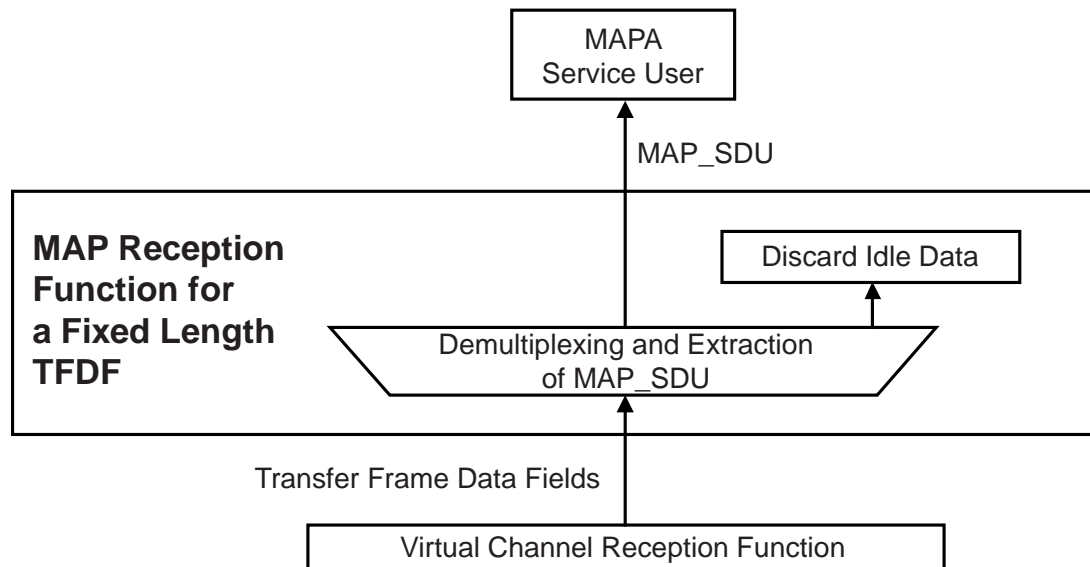


Figure 4-19: Abstract Model of MAP Reception Function for Fixed-Length TFDFs

4.3.3.3 MAP Reception Function for Variable-Length TFDFs

4.3.3.3.1 Discussion

The MAP Reception Function used to extract a variable-length MAP_SDU from a variable-length TFDZs are associated with Construction rules ‘100’, ‘101’, ‘110’ and ‘111’.

The Map Reception Function for variable-length TFDFs is identical to the packet extraction function for variable-length TFDFs. The UPID identifies the contents to either be a packet or a MAP_SDU.

4.3.3.3.2 Requirements

4.3.3.3.2.1 The MAP shall extract the MAP_SDU from variable-length TFDFs received from the MAP Demultiplexing Function.

4.3.3.3.2.2 A Segmented MAP_SDU within a variable-length TFDF shall be extracted and reconstructed from the TFDF using the TFDF Construction Rules ‘100’, ‘101’, and ‘110’.

4.3.3.3.2.3 An incomplete MAP_SDU shall be discarded in cross-support situations.

4.3.3.3.2.4 The extracted and reconstructed MAP_SDU shall be delivered to the MAPA Service user associated with its GMAP ID.

NOTE – An abstract model of the MAP Reception Function for fixed-length TFDFs is illustrated in figure 4-20.

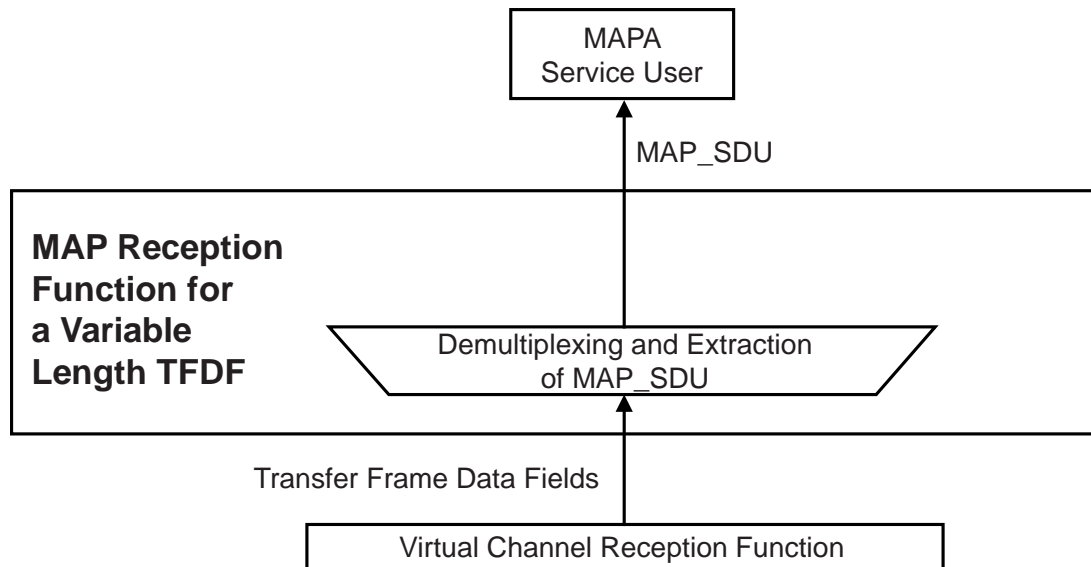


Figure 4-20: Abstract Model of MAP Reception Function for Variable-Length TFDFs

4.3.4 MAP OCTET STREAM EXTRACTION FUNCTION

4.3.4.1 The Octet Stream Extraction Function shall be used to extract variable-length user defined Octet Stream Data from variable-length TFDZs on a MAP Channel.

NOTE – There is an instance of the Octet Stream Extraction Function for each MAP Channel that carries Octet Stream Data.

4.3.4.2 The Octet Stream Extraction Function shall extract user-defined Octet Stream Data from TFDZs received from the MAP Demultiplexing Function. The TFDF Construction Rule used shall be ‘011’.

4.3.4.3 Extracted Octet Stream Data shall be delivered to the Octet Stream Service user identified by the MAP ID.

NOTE – An abstract model of the MAP Octet Stream Extraction Function is illustrated in figure 4-21.

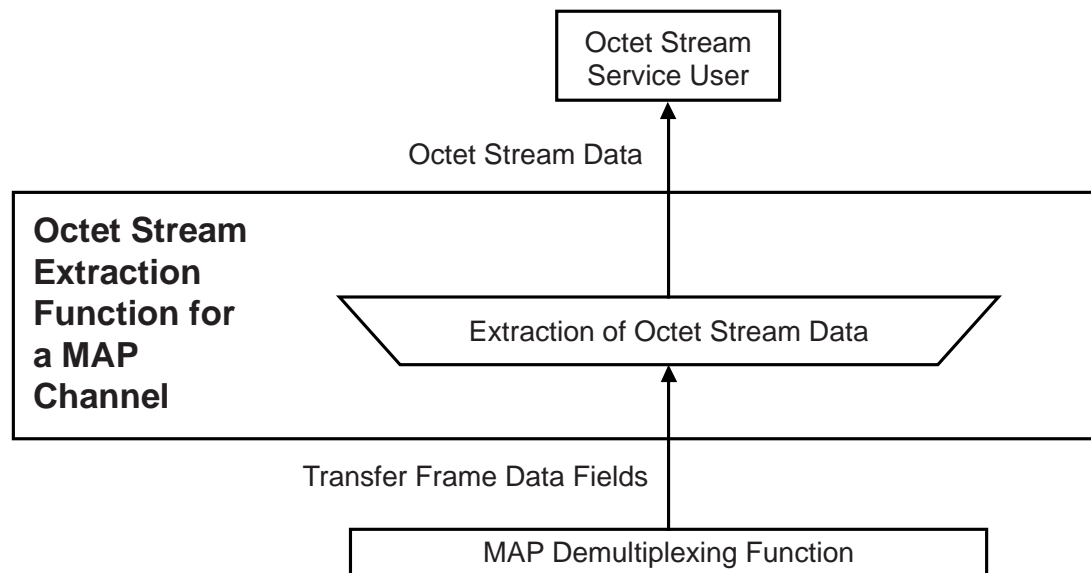


Figure 4-21: Abstract Model of MAP Octet Stream Extraction Function

4.3.5 MAP DEMULTIPLEXING FUNCTION

4.3.5.1 The MAP Demultiplexing Function shall be used to demultiplex service data units from different MAP Channels on a Virtual Channel.

NOTE – There is an instance of the MAP Demultiplexing Function for each Virtual Channel that has multiple MAP Channels.

4.3.5.2 The MAP Demultiplexing Function shall examine the MAP ID in the Transfer Frame Header of the incoming service data units and shall route them to the instances of the Packet Extraction, MAP Reception, and/or Octet Stream Extraction Functions.

4.3.5.3 Service data units associated with an invalid MAP ID shall be discarded.

NOTE – An abstract model of the MAP Demultiplexing Function is illustrated in figure 4-22.

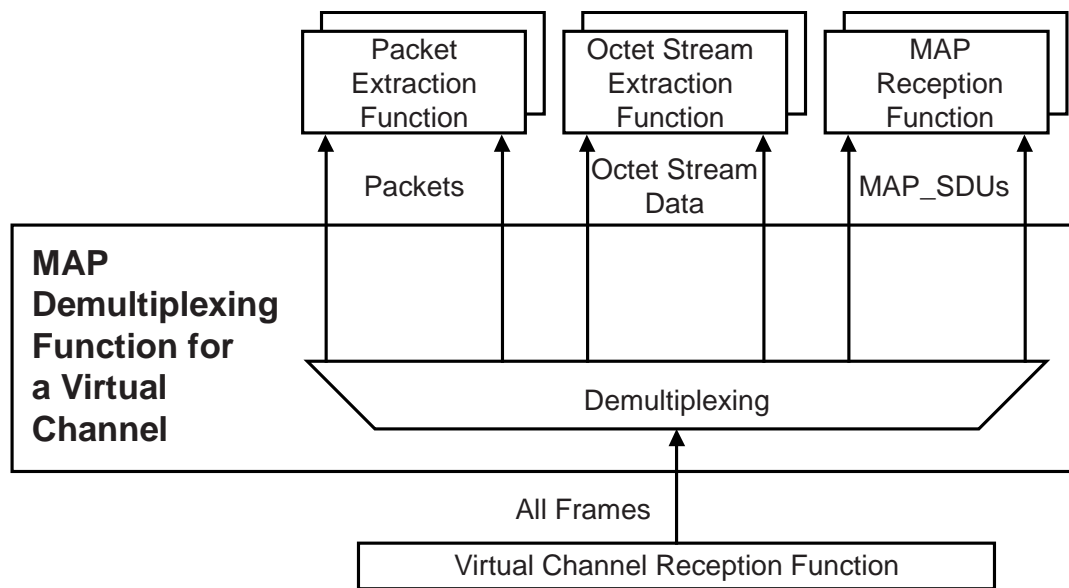


Figure 4-22: Abstract Model of MAP Demultiplexing Function

4.3.6 VIRTUAL CHANNEL RECEPTION FUNCTION

4.3.6.1 The Virtual Channel Reception Function shall be used to decommutate fields of Transfer Frames of a Virtual Channel.

NOTE – There is an instance of the Virtual Channel Reception Function for each Virtual Channel.

4.3.6.2 The Virtual Channel Reception Function shall extract data units contained in the Data Field of the Transfer Frames and deliver them to the user (i.e., the Packet Extraction Function, the Octet Stream Extraction Function, or the MAPA Service user).

4.3.6.3 If there is a user of the MC_OCF Service for a particular Virtual Channel, OCF_SDUs contained in the Operational Control Field of the Transfer Frames shall be extracted and delivered to the user.

4.3.6.4 If a gap in the Virtual Channel Frame Count is detected, a Loss Flag may (optionally) be delivered to the users.

NOTE – An abstract model of the Virtual Channel Reception Function is illustrated in figure 4-23.

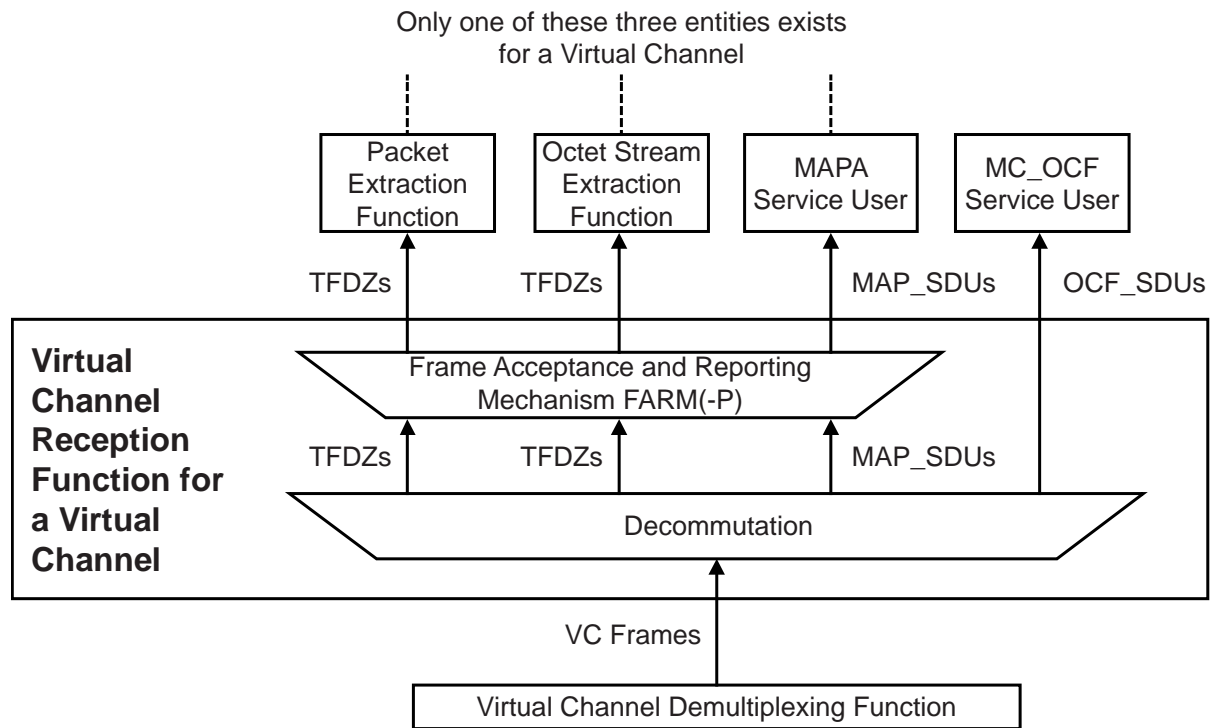


Figure 4-23: Abstract Model of Virtual Channel Reception Function

4.3.7 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION

4.3.7.1 The Virtual Channel Demultiplexing Function shall be used to demultiplex Transfer Frames of different Virtual Channels of a Master Channel.

NOTE – There is an instance of the Virtual Channel Demultiplexing Function for each Master Channel that has multiple Virtual Channels.

4.3.7.2 The Virtual Channel Demultiplexing Function shall examine the VCID in the incoming stream of Transfer Frames and route them to the instances of the Virtual Channel Reception Function and, if present, to the Virtual Channel Frame Service users.

4.3.7.3 If a gap in the Virtual Channel Frame Count is detected, a Loss Flag may (optionally) be delivered to the users.

4.3.7.4 When a frame from any VC's header signals that an OCF_SDU is contained, that OCF_SDU shall be extracted and delivered via the OCF service.

NOTES

- 1 The OCF service is provided by the Virtual Channel Demultiplexing Function.

- 2 OID Transfer Frames are discarded, when any of the coding schemes defined in references [3], [4], and [5] are used. Transfer Frames with an invalid VCID are also discarded.
- 3 When any of the coding schemes defined in references [6] and [7] are used, then this function does not discard an OID Transfer Frame.
- 4 An abstract model of the Virtual Channel Demultiplexing Function is illustrated in figure 4-24.

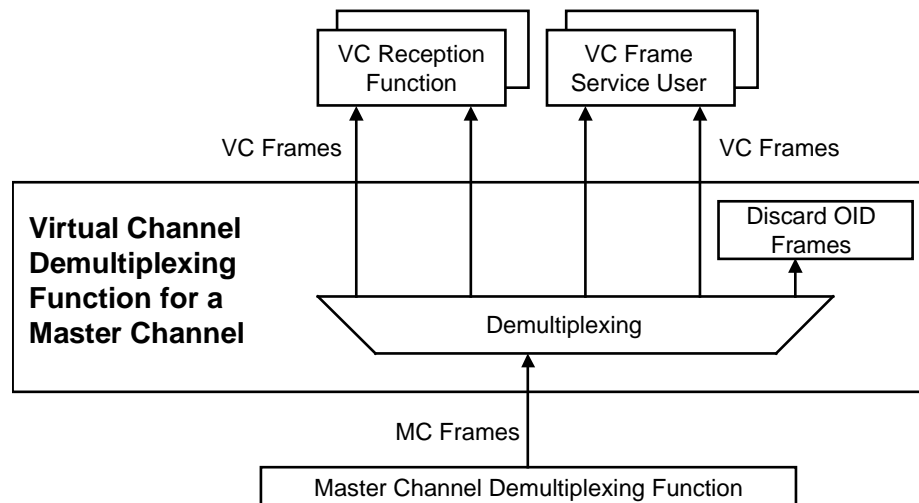


Figure 4-24: Abstract Model of Virtual Channel Demultiplexing Function

4.3.8 MASTER CHANNEL DEMULTIPLEXING FUNCTION

4.3.8.1 The Master Channel Demultiplexing Function shall be used to demultiplex Transfer Frames of different Master Channels of a Physical Channel.

NOTE – There is an instance of the Master Channel Demultiplexing Function for each Physical Channel that has multiple Master Channels.

4.3.8.2 The Master Channel Demultiplexing Function shall examine the MCID in the incoming stream of Transfer Frames and route them to the instances of the Virtual Channel Demultiplexing Function and, if present, to the Master Channel Frame Service users.

4.3.8.3 If frame loss is signaled by the underlying Synchronization and Channel Coding Sublayer, a Loss Flag may (optionally) be delivered to the users.

4.3.8.4 Transfer Frames with an invalid MCID shall be discarded.

NOTE – An abstract model of the Master Channel Demultiplexing Function is illustrated in figure 4-25.

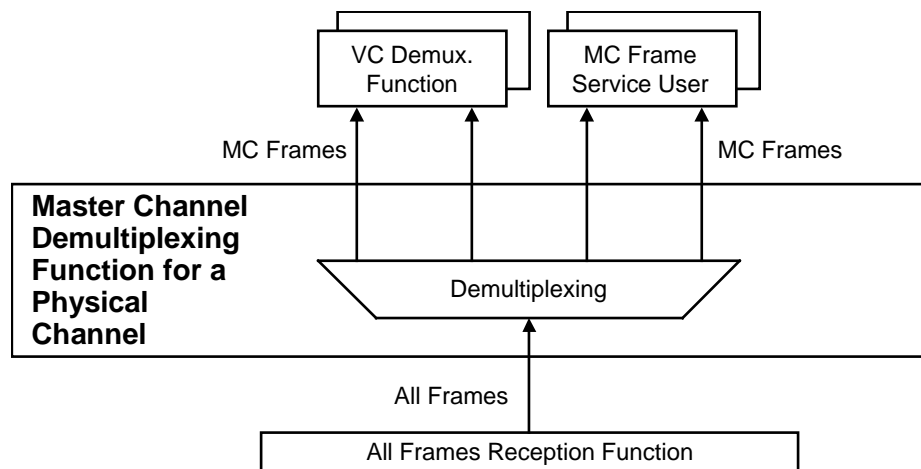


Figure 4-25: Abstract Model of Master Channel Demultiplexing Function

4.3.9 ALL FRAMES RECEPTION FUNCTION

4.3.9.1 The All Frames Reception Function shall be used to reconstitute Transfer Frames from the data stream provided by the specific Synchronization and Channel Coding Sublayer (references [3]–[7]) selected by the user and to perform checks to determine whether the reconstituted Transfer Frames are valid.

4.3.9.2 If the Synchronization and Channel Coding Sublayer used is TC (reference [6]), then the All Frames Reception Function shall perform the following two procedures in the following order:

- a) Frame Delimiting and Fill Removal Procedure; and
- b) Frame Validation Check Procedure.

4.3.9.3 The Frame Delimiting and Fill Removal Procedure shall be used to reconstitute Transfer Frames from the data stream provided by the TC Synchronization and Channel Coding Sublayer and to remove any Fill Data transferred from the TC Synchronization and Channel Coding Sublayer.

4.3.9.4 The Frame Validation Check Procedure shall be used to perform standard Frame Validation Checks on all Transfer Frames reconstituted by the Frame Delimiting and Fill Removal Procedure.

NOTE – There is an instance of the All Frames Reception Function for each Physical Channel.

4.3.9.5 If the Frame Error Control Field is present in the Transfer Frame, the All Frames Reception Function shall recompute the CRC value for the Transfer Frame and compare it to the content of the Frame Error Control field to determine if the Transfer Frame contains a detected error.

4.3.9.6 A Transfer Frame that contains a detected error may be discarded in cross support situations.

4.3.9.7 If the optional Insert Service is activated, the All Frames Reception Function shall extract the IN_SDUs from the Insert Zone of the incoming stream of Transfer Frames, regardless of their GMAP ID, and deliver them to the Insert Service user. If error protection of the IN_SDUs is not required, this function may be performed prior to decoding of Frame Error Control Field described above.

NOTE – An abstract model of the All Frames Reception Function is illustrated in figure 4-26.

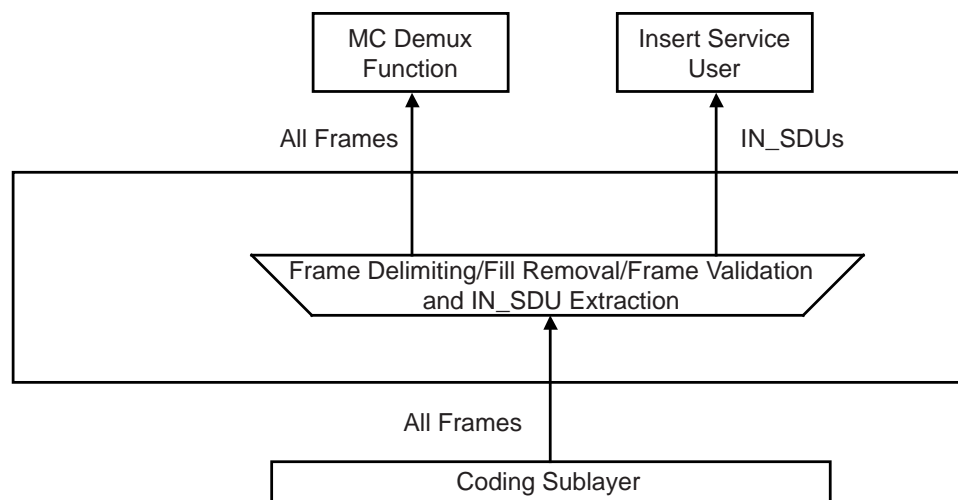


Figure 4-26: Abstract Model of All Frames Reception Function

4.3.9.8 Frame Delimiting and Fill Removal Procedure

4.3.9.8.1 The Channel Coding and Synchronization Recommended Standard (reference [6]) shall be used as the specification for the Synchronization and Channel Coding Sublayer immediately below this procedure.

4.3.9.8.2 At the sending end, USLP shall pass one Transfer Frame to the TC Synchronization and Channel Coding Sublayer.

NOTE – The TC Synchronization and Channel Coding Sublayer encodes the Transfer Frames to protect them from errors that may be introduced as they are transmitted through the space link. Fill Data may have to be inserted by the TC Synchronization and Channel Coding Sublayer to ensure correct transmission of all valid data.

4.3.9.8.3 The receiving end of this protocol shall receive as an input from the TC Synchronization and Channel Coding Sublayer a series of data octets, corresponding to the

decoded Transfer Frame(s), which have been declared ‘clean’ by the TC Synchronization and Channel Coding Sublayer insofar as they contain no detected errors.

NOTE – The TC Synchronization and Channel Coding Sublayer provides a ‘Data Start’ signal to this protocol, indicating that data are being transferred. The Data Start signal is set to ‘true’ while the TC Synchronization and Channel Coding Sublayer is in the process of actively transferring data octets. Since the first octet transferred after Data Start goes ‘true’ corresponds to the first octet of the first Transfer Frame, this Procedure may delimit this Transfer Frame—and each of any successive Transfer Frames—by reading the Frame Length field in the first Transfer Frame Header, and then successively reading the Frame Length field in each subsequent Header. The Data Start signal is set to ‘false’ (indicating ‘Data Stop’) when the TC Synchronization and Channel Coding Sublayer stops transferring octets because of a decoder failure or channel deactivation. Decoding failure may be caused by the normal end of the transmitted Transfer Frame(s) or by a genuine channel-induced error.

4.3.9.8.4 If one valid Frame Length field is detected by the Frame Delimiting and Fill Removal Procedure and the number of octets received when the Data Stop condition occurs equals the number of octets specified by the Frame Length, then the Transfer Frame shall be passed on to the Frame Validation Check Procedure (see 4.3.9.9) as it is delimited.

4.3.9.8.5 If a valid Frame Length field is detected by the Frame Delimiting and Fill Removal Procedure but the number of octets received when the Data Stop condition occurs is fewer than the number of octets specified by that Frame Length, then all those octets shall be discarded.

NOTE – Receipt of fewer octets than specified in Frame Length field indicates that a failure has occurred, possibly resulting from a channel error detected during reception of the data stream within the TC Synchronization and Channel Coding Sublayer.

4.3.9.8.6 If a valid Frame Length field is detected by the Frame Delimiting and Fill Removal Procedure but the number of octets received when the Data Stop condition occurs is greater than the number of octets specified by that Frame Length, the procedure shall

- a) assume that the octets following the final expected octet of the frame are Fill Data appended by the sending end of the Synchronization and Channel Coding Sublayer to complete the last codeword (see reference [6]);
- b) discard that Fill Data;
- c) pass the Frame to the Frame Validation Check Procedure (see 4.3.9.9).

NOTES

- 1 Because the receiving end of the TC Synchronization and Channel Coding Sublayer cannot distinguish between valid data and Fill Data, the Fill Data needs to be stripped by this protocol.
- 2 The characteristics of the BCH codeword structure are such that no more than six octets of Fill Data can occur. If fewer than five trailing octets of Fill Data are present, then they cannot possibly form a Transfer Frame Header, and they will be immediately discarded by this Procedure. If five or six trailing octets of Fill Data exist, this procedure might attempt to interpret the Fill Data as a new Transfer Frame Header; however, the subsequent Frame Validation Checks (see 4.3.9.9) will prevent this from happening because the fill pattern of '01010101' appearing in each octet will violate at least one of the validation tests. In particular, this pattern appearing where the Frame Length field might be expected will indicate a frame length that exceeds the number of octets received from the Synchronization and Channel Coding Sublayer, thus failing a test and causing the trailing five or six octets to be discarded.
- 3 After each Transfer Frame is reconstituted by the Frame Delimiting and Fill Removal Procedure, it will next be subjected to a set of standard tests called Frame Validation Checks.

4.3.9.9 Frame Validation Check Procedure

4.3.9.9.1 The TC Channel Coding and Synchronization Recommended Standard (reference [6]) shall be used as the specification for the Synchronization and Channel Coding Sublayer immediately below this procedure.

4.3.9.9.2 The Frame Validation Checks shall be applied to all incoming Transfer Frames.

4.3.9.9.3 Failure to pass any test within the Frame Validation Checks shall cause the Transfer Frame to be rejected (discarded).

4.3.9.9.4 The Frame Validation Checks shall consist of the following tests:

- a) The Transfer Frame must have an expected Transfer Frame Version Number.
- b) The Transfer Frame must have one of the expected MCIDs (Transfer Frame Version Number and Spacecraft IDs).
- c) The Transfer Frame Header must not contain any values which are not consistent with the implemented features for that spacecraft.
- d) The value of the Frame Length must be consistent with the number of octets that are present.
- e) If the Frame Error Control Field is present, the recomputed CRC value for the Transfer Frame must match the content of the Frame Error Control Field.

5 MANAGED PARAMETERS WITHOUT SDLS OPTION

5.1 OVERVIEW

In order to conserve bandwidth on the space link, some parameters associated with USLP are handled by management rather than by inline communications protocol. The managed parameters are those which tend to be static for long periods of time, and whose change generally signifies a major reconfiguration of the protocol entities associated with a particular mission. Through the use of a management system, management conveys the required information to the protocol entities.

In this section, the managed parameters used by USLP are listed for each of the Channels and for Packet transfer. These parameters are defined in an abstract sense and are not intended to imply any particular implementation of a management system.

NOTE – This section specifies managed parameters for USLP without support for the SDLS protocol. Additional managed parameters for USLP with the SDLS option are specified in 6.6.

5.2 MANAGED PARAMETERS FOR A PHYSICAL CHANNEL

The managed parameters associated with a Physical Channel shall conform to the definitions in table 5-1.

Table 5-1: Managed Parameters for a Physical Channel

Managed Parameter	Allowed Values
Physical Channel Name	Character String
Transfer Frame Type	Fixed Length or Variable Length
Maximum Transfer Frame Length (octets)	Integer (maximum 65536 octets)
Transfer Frame Version Number	'1100' binary
Valid Spacecraft IDs	16-bit Integer
MC Multiplexing Scheme	Mission Specific
Presence of Insert Zone	Present, Absent
Insert Zone Length (octets)	Integer
Presence of Frame Error Control	Present, Absent
Frame Error Control Length (octets)	2 or 4
Maximum Number of Transfer Frames Given to the Coding Sublayer as a Single Data Unit. NOTE – This constraint applies even if reference [6] is used.	1
Maximum Value of the Repetitions Parameter to the Coding & Synchronization Sublayer	Integer
NOTES 1 The Transfer Frame Type shall be fixed-length when USLP is used over references [3] and [5]. It can be variable-length when USLP is used over references [6] and [10]. 2 The value of the Transfer Frame Length shows either the frame length (when Transfer Frame Type is fixed-length) or the maximum allowed frame length (when Transfer Frame Type is variable-length). 3 The number of allowed values of the Transfer Frame Length may be constrained by the actual coding scheme selected when USLP is used over references [3], [4], and [5]. 4 The Maximum Number of Transfer Frames given to the Coding & Synchronization Sublayer applies only when USLP is used over reference [6].	

5.3 MANAGED PARAMETERS FOR A MASTER CHANNEL

The managed parameters associated with a Master Channel shall conform to the definitions in table 5-2.

Table 5-2: Managed Parameters for a Master Channel

Managed Parameter	Allowed Values
Maximum Transfer Frame Length (octets)	Integer (maximum 65536 octets)
Transfer Frame Type	Fixed Length or Variable Length
Spacecraft ID	16-bit Integer
Valid VCIDs	Selectable Set of Integers (from 0 to 62) (in addition to VCID 63)
VC Multiplexing Scheme	Mission Specific
Generate OID Frame	True, False
<p>NOTES</p> <p>1 The value of the Transfer Frame length shows either the frame length (when Transfer Frame length is fixed-length) or the maximum allowed frame length (when Transfer Frame length is variable-length).</p> <p>2 The value of the Transfer Frame Version Number is the same for all Transfer Frames on a Physical Channel.</p> <p>1 For VCID the binary value of 'all ones' (i.e., 63) is always valid as it is reserved for OID Transfer Frames by 4.1.4.1.6; i.e., the number of valid VCIDs always includes value 63 and the Selectable Set of Integers defined above.</p> <p>2 Transfer Frame Type must be fixed, if the Physical Channel Transfer Frame Type is fixed.</p>	

5.4 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

The managed parameters associated with a Virtual Channel shall conform to the definitions in table 5-3.

Table 5-3: Managed Parameters for a Virtual Channel

Managed Parameter	Allowed Values
Maximum Transfer Frame Length (octets)	Integer (maximum 65536 octets)
Transfer Frame Type	Fixed Length or Variable Length
Spacecraft ID	16-bit Integer
VCID	0, 1, ..., 62 (63 reserved)
Frame Sequence Control Supported	True, False
VC Count Size for Sequence Control	Integer
VC Count Size for Expedited Integer	Integer
COP in Effect	COP-1, COP-P, None
CLCW Version Number	1
CLCW Reporting Rate	as required to support COP
Valid MAP IDs	Set of integers (from 0 to 31)
MAP Multiplexing Scheme	Mission Specific
TFDF Length	Integer (maximum 65514 octets)
Truncated Primary Transfer Frame Header Length	7 or 8 octets
Generate OID Frame	True, False
Inclusion of OCF Allowed	True, False
Inclusion of OCF Required	True, False
Value for the Repetitions parameter to the Coding Sublayer when transferring frames carrying service data on the Sequence-Controlled Service	Integer
Value for the Repetitions parameter to the Coding Sublayer when transferring frames carrying COP control commands	Integer
Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released	Integer

Managed Parameter	Allowed Values
Maximum delay in milliseconds between releases of frames of the same VC	Integer
NOTES	
1	The value of the Transfer Frame Length shows either the frame length (when Transfer Frame type is fixed-length) or the maximum allowed frame length (when Transfer Frame Type is Variable-Length).
2	VCID value 63 (i.e., the binary ‘all ones’) is reserved for OID Transfer Frames by 4.1.4.1.6 Generation of OID frames applies to all CCSDS space data links where USLP is used.
3	Transfer Frame Type must be fixed, if either the Physical Channel or Master Channel Transfer Frame Type is fixed.
4	Only when the MC allows variable-length frames, then the length of a given Transfer Frame is set by the VC Transfer Frame Length above.

5.5 MANAGED PARAMETERS FOR A MAP CHANNEL

The managed parameters associated with a MAP Channel shall conform to the definitions in table 5-4.

Table 5-4: Managed Parameters for a MAP Channel

Managed Parameter	Allowed Values
Spacecraft ID	16-bit Integer
VCID	0, 1, ..., 63
MAP ID	0, 1, ..., 31
Frame Sequence Control Supported	True, False
Minimum TFDF Length	Integer (minimum 0 octets)
Maximum TFDF Length	Integer (maximum 65514 octets)
Service Data Unit Type	Packet, MAP_SDU, Stream Data
USLP Protocol ID supported	Integer (registered in SANA)

5.6 MANAGED PARAMETERS FOR PACKET TRANSFER

The managed parameters associated with a Virtual Channel used for the Virtual Channel Packet Service shall conform to the definitions in table 5-5.

Table 5-5: Managed Parameters for Packet Transfer

Managed Parameter	Allowed Values
Valid Packet Version Numbers	Set of Integers
Maximum Packet Length (octets)	Integer
Whether incomplete Packets are required to be delivered to the user at the receiving end	Required, Not required

6 PROTOCOL SPECIFICATION WITH SDLS OPTION

6.1 OVERVIEW

This section specifies the protocol data unit and the procedures of USLP with support for the Space Data Link Security Protocol (reference [14]). If the USLP protocol entity supports SDLS, it has managed parameters for each Virtual Channel to indicate whether SDLS is in use for that channel (see 6.6). Section 4 contains the specification of the protocol without the SDLS option.

6.2 USE OF SDLS PROTOCOL

If SDLS as defined in reference [14] is required over the USLP space data link, then the SDLS protocol shall be used.

NOTE – The SDLS protocol provides a security header and trailer along with associated procedures that may be used with USLP to provide data authentication and data confidentiality at the Data Link Layer.

6.3 USLP TRANSFER FRAME WITH SDLS

6.3.1 OVERVIEW

To support the use of the SDLS security features, a Security Header and a Security Trailer are defined for a USLP Transfer Frame. The use of SDLS can vary between Virtual Channels, so a managed parameter indicates the presence of the Security Header (see 6.6). If the Security Header is present, then SDLS is in use for the Virtual Channel. This subsection specifies the USLP Transfer Frames on a Virtual Channel that is using SDLS.

If a Virtual Channel is not using SDLS, then the frames are as specified in 4.1.

The Security Header and Security Trailer are placed before and after the Transfer Frame Data Field, and they reduce the length of the Transfer Frame Data Field compared to a frame without SDLS. Figure 6-1 compares the frame fields for a frame without SDLS and a frame with SDLS. The upper part of figure 6-1 shows the USLP Transfer Frame without the SDLS fields and is the same as figure 4-1.

NOTE – The Frame Error Control Field is controlled by the managed parameters (see section 5).

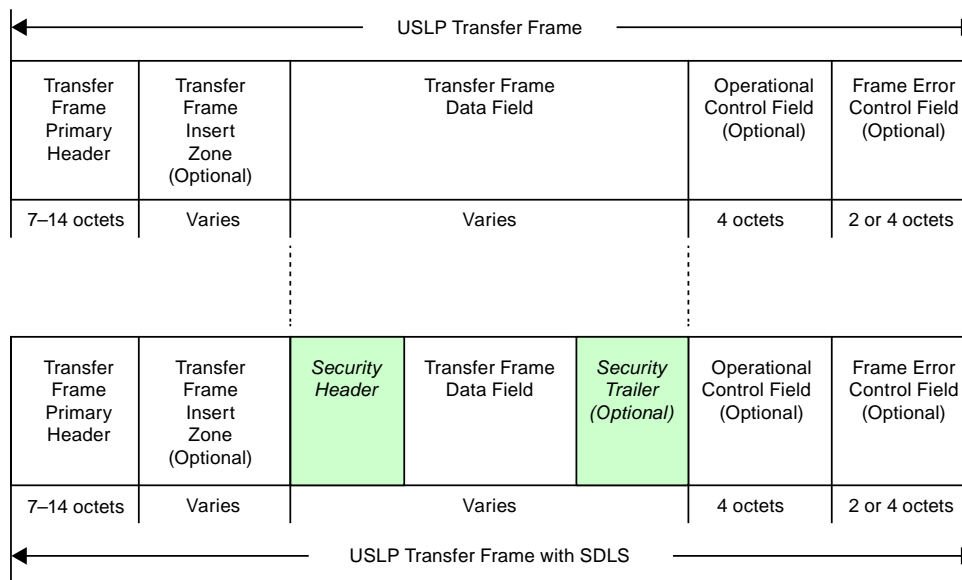


Figure 6-1: Frame without SDLS Compared to Frame with SDLS

6.3.2 TRANSFER FRAME PRIMARY HEADER IN A FRAME WITH SDLS

The Transfer Frame Primary Header for a frame with SDLS shall conform to the specifications of 4.1.2.

NOTE – The Transfer Frame Primary Header is the same for a frame without SDLS and a frame with SDLS.

6.3.3 TRANSFER FRAME INSERT ZONE IN A FRAME WITH SDLS

The Transfer Frame Insert Zone shall conform to the specifications of 4.1.3.

NOTE – The Transfer Frame Insert Zone is the same for a frame without SDLS and a frame with SDLS.

6.3.4 SECURITY HEADER

If present, the Security Header shall follow, without gap, the Transfer Frame Insert Zone if a Transfer Frame Insert Zone is present, or the Transfer Frame Primary Header if a Transfer Frame Insert Zone is not present.

NOTES

- 1 The presence of the Security Header is a managed parameter of the Virtual Channel (see 6.6). If the Security Header is not present, the Transfer Frame has the format specified in 4.1.

- 2 The requirements for the length and contents of the Security Header are specified in reference [14].
- 3 The length of the Security Header is an integral number of octets and is a managed parameter of the Virtual Channel.

6.3.5 TRANSFER FRAME DATA FIELD IN A FRAME WITH SDLS

6.3.5.1 The Transfer Frame Data Field of a frame with SDLS shall conform to the specifications of 4.1.4.1.1 through 4.1.4.1.2 as modified by 6.3.5.2.

6.3.5.2 In a Transfer Frame with SDLS, the Transfer Frame Data Field shall

- a) follow, without gap, the Security Header;

NOTE – Therefore in this case the data unit that is placed into the Transfer Frame Data Field follows, without gap, the Security Header.

- b) contain an integer number of octets equal to the Transfer Frame length, minus
 - the lengths of the Transfer Frame Primary Header and of the Security Header;
 - the lengths of the Transfer Frame Insert Zone, of the Security Trailer, Operational Control Field, and of the Frame Error Control Field, if any of these are present.

6.3.6 SECURITY TRAILER

If present, the Security Trailer shall follow, without gap, the Transfer Frame Data Field.

NOTES

- 1 The Security Trailer is optional in a USLP Transfer Frame with SDLS. The presence of the Security Trailer is a managed parameter of the Virtual Channel (see 6.6).
- 2 The requirements for the length and contents of the Security Trailer are specified in reference [14].
- 3 The length of the Security Trailer is an integral number of octets and is a managed parameter of the Virtual Channel.

6.3.7 OPERATIONAL CONTROL FIELD IN A FRAME WITH SDLS

6.3.7.1 The Operational Control Field of a frame with SDLS shall conform to the specifications of 4.1.5.2 through 4.1.5.4 as modified by 6.3.7.2.

6.3.7.2 In a Transfer Frame with SDLS, the Operational Control Field, if present, shall occupy the four octets following, without gap, the Security Trailer if this is present, or the Transfer Frame Data Field if a Security Trailer is not present.

6.3.8 FRAME ERROR CONTROL FIELD IN A FRAME WITH SDLS

6.3.8.1 The Frame Error Control Field of a frame with SDLS shall conform to the specifications of 4.1.6.1.1, 4.1.6.1.2, B1, and B1.2, as modified by 6.3.8.2.

6.3.8.2 In a Transfer Frame with SDLS, the Frame Error Control Field, if present, shall occupy from two to four octets following, without gap,

- the Operational Control Field if this is present;
- the Security Trailer if this is present and the Operational Control Field is not present;
- the Transfer Frame Data Field if the Operational Control Field and the Security Trailer are not present.

6.4 SENDING END PROTOCOL PROCEDURES WITH SDLS

6.4.1 OVERVIEW

When a secure USLP link is required, USLP supports the use of the SDLS protocol. In this case, USLP contains differences in the sending end procedures compared to the procedures described in 4.2. This subsection defines those differences.

The SDLS ApplySecurity Function may interface with USLP at either the Virtual Channel Generation Function (4.2.6) or the Virtual Channel Multiplexing Function (4.2.7). The choice of where to apply security within the USLP Data Link Layer depends upon several factors, such as the number of Security Associations (SAs), their type (one VC or more than one VC per SA), and the corresponding source and termination of the security function(s), key management, and the use of the anti-replay feature.

There can be security configurations in which, for example, one or several SAs covering just one VC each are present. The physical location of the security processing may not be the same for all Virtual Channels, at the sending end or at the receiving end. This case can be supported by placing the SDLS interface in the Virtual Channel Generation Function where the greatest flexibility in managing the security function occurs.

Conversely, with the SDLS interface in the Virtual Channel Multiplexing Function, the security configuration can include multiple Virtual Channels (not necessarily all) sharing an SDLS Security Association. The call to the SDLS ApplySecurity function follows the Virtual Channel multiplexing, so that the SDLS processing is applied to the multiplexed stream of frames.

6.4.2 PACKET PROCESSING FUNCTION WITH SDLS

6.4.2.1 The Packet Processing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.2 and 6.4.2.2.

6.4.2.2 When handling Packets on a Virtual Channel that uses SDLS, the Packet Processing Function shall apply the Transfer Frame Data Field specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The Packet Processing Function generates a TFDZ to fit exactly within the Transfer Frame Data Field (see 4.1.4).

6.4.3 OCTET STREAM PROCESSING FUNCTION WITH SDLS

6.4.3.1 The Octet Stream Processing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.4 and 6.4.3.2.

6.4.3.2 When handling Octet Stream Data on a Virtual Channel that uses SDLS, the Octet Stream Processing Function shall apply the Transfer Frame Data Field specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The Octet Stream Processing Function generates TFDZs to fit exactly within the Transfer Frame Data Field (see 4.1.4).

6.4.4 VIRTUAL CHANNEL GENERATION FUNCTION WITH SDLS

6.4.4.1 When assembling a Transfer Frame, the Virtual Channel Generation Function shall conform to the specifications of 4.2.6, 6.3, and 6.4.4.2 through 6.4.4.3.

6.4.4.2 The Security Header, and the Security Trailer if it is present for the Virtual Channel, shall be kept empty.

NOTES

- 1 The SDLS ApplySecurity Function specified in reference [14] provides the contents of these security fields as necessary and may modify the contents of the Transfer Frame Data Field by encrypting the data.
- 2 The lengths of the Security Header and Security Trailer are managed parameters of the Virtual Channel (see 6.6).

6.4.4.3 If the Virtual Channel Generation Function contains the interface to the SDLS protocol,

- a) it shall call the SDLS ApplySecurity function for the Transfer Frames that it assembles for Virtual Channels that use SDLS;

- b) the order of processing between the functions of the USLP and SDLS protocols shall occur as follows in the Virtual Channel Generation Function:

- 1) the frame assembly processing by the Virtual Channel Generation Function;
- 2) the call by the Virtual Channel Generation Function to the SDLS ApplySecurity Function.

NOTE – The way in which Transfer Frame data is passed between the Virtual Channel Generation Function and the SDLS ApplySecurity Function is implementation-dependent.

6.4.5 VIRTUAL CHANNEL MULTIPLEXING FUNCTION WITH SDLS

6.4.5.1 The Virtual Channel Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.7 and 6.4.5.2.

6.4.5.2 If the Virtual Channel Multiplexing Function contains the interface to the SDLS protocol,

- a) it shall call the SDLS ApplySecurity function for Transfer Frames on Virtual Channels that use SDLS after the frames have been selected by the multiplexing algorithm;
- b) the order of processing between the functions of the USLP and SDLS protocols shall occur as follows in the Virtual Channel Multiplexing Function:
 - 1) the Virtual Channel multiplexing processing of the Virtual Channel Multiplexing Function;
 - 2) the call by the Virtual Channel Multiplexing Function to the SDLS ApplySecurity Function.

6.4.6 MASTER CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The Master Channel Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.8.

6.4.7 ALL FRAMES GENERATION FUNCTION WITH SDLS

The All Frames Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.9.

NOTE – There is no interface between the SDLS ApplySecurity function with the USLP ‘All Frames Generation’ function in order to guarantee that the Frame Error Control field is computed after the SDLS function has processed the frame.

6.5 RECEIVING END PROTOCOL PROCEDURES WITH SDLS

6.5.1 OVERVIEW

When the USLP Transfer Frame Protocol supports the use of the SDLS protocol, there are differences in the receiving end procedures compared to the procedures described in 4.3. This subsection defines those differences.

The position of the SDLS interface is generally selected to reflect the position of the corresponding interface at the sending end. These choices include the Virtual Channel Demultiplexing Function or the Virtual Channel Reception Function, corresponding to the options discussed in 6.4.1.

6.5.2 ERROR REPORTING

6.5.2.1 Discussion

Depending on the security features in use, the SDLS ProcessSecurity function specified in reference [14] can verify the authenticity of the frame and it can decrypt the contents of the Transfer Frame Data Field. If the SDLS ProcessSecurity Function detects any errors, these are reported to either the Virtual Channel Demultiplexing Function or the Virtual Channel Reception Function. The way that Transfer Frame data is passed between either of these Functions and the SDLS ProcessSecurity Function is implementation-dependent.

6.5.2.2 Requirements

6.5.2.2.1 If the SDLS ProcessSecurity Function does not report an error, the Virtual Channel Reception Function shall extract the contents of the Transfer Frame Data Field from the frame and deliver it to its user (or Function).

6.5.2.2.2 If the SDLS ProcessSecurity Function reports an error, either the Virtual Channel Demultiplexing Function or the Virtual Channel Reception Function shall discard the frame (depending on the interface point).

NOTE – In this case, the optional Verification Status Code parameter can be used to inform the user of the relevant service (see 3.3.2.9, 3.4.2.5, and 3.5.2.5).

6.5.3 PACKET EXTRACTION FUNCTION WITH SDLS

6.5.3.1 The Packet Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.2 and 6.5.3.2.

6.5.3.2 When handling Packets on a Virtual Channel that uses SDLS, the Packet Extraction Function shall apply the Transfer Frame Data Field specification in 6.3.5 to determine the expected length of the TFDZs that it receives.

NOTE – The Packet Extraction Function receives TFDZs that fit exactly within the Transfer Frame Data Field (see 4.1.4).

6.5.4 OCTET STREAM EXTRACTION FUNCTION WITH SDLS

6.5.4.1 The Octet Stream Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.4 and 6.5.4.2.

6.5.4.2 When handling Octet Stream Data on a Virtual Channel that uses SDLS, the Octet Stream Extraction Function shall apply the Transfer Frame Data Field specification in 6.3.5 to determine the length of the TFDZs that it receives.

NOTE – The Octet Stream Extraction Function receives TFDZs that fit exactly within the Transfer Frame Data Field (see 4.1.4).

6.5.5 VIRTUAL CHANNEL RECEPTION FUNCTION WITH SDLS

6.5.5.1 The Virtual Channel Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.6 and 6.5.5.2 through 6.5.5.3.

6.5.5.2 If the Virtual Channel Reception Function contains the interface to the SDLS protocol, it shall call the SDLS ProcessSecurity function for the Transfer Frames that it handles for Virtual Channels that use SDLS.

6.5.5.3 When handling a Transfer Frame on a Virtual Channel that uses SDLS, the Virtual Channel Reception Function shall apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame.

6.5.6 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

6.5.6.1 The Virtual Channel Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.7 and 6.5.6.2.

6.5.6.2 If the Virtual Channel Demultiplexing Function contains the interface to the SDLS protocol, it shall call the SDLS ProcessSecurity function for Transfer Frames on Virtual Channels that use SDLS, before the demultiplexing is applied.

6.5.7 MASTER CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The Master Channel Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.8.

6.5.8 ALL FRAMES RECEPTION FUNCTION WITH SDLS

The All Frames Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.9.

6.6 MANAGED PARAMETERS WITH SDLS

6.6.1 OVERVIEW

Managed parameters for the SDLS protocol are specified in reference [14].

6.6.2 ADDITIONAL MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

The managed parameters associated with a Virtual Channel for USLP that supports the SDLS protocol shall conform to the definitions in table 5-5 and the additional definitions in table 6-1.

Table 6-1: Additional Managed Parameters for a Virtual Channel when the Unified Space Data Link Protocol Supports SDLS

Managed Parameter		Allowed Values
Presence of Space Data Link Security Header		Present / Absent
Presence of Space Data Link Security Trailer		Present / Absent
Length of Space Data Link Security Header (octets)		Integer
Length of Space Data Link Security Trailer (octets)		Integer
NOTES		
1	If the Security Header is present then SDLS is in use for the Virtual Channel.	
2	The valid lengths for the Security Header and Security Trailer are specified in reference [14].	

ANNEX A

**IMPLEMENTATION CONFORMANCE
STATEMENT (ICS) PROFORMA**

(NORMATIVE)

[To be supplied.]

ANNEX B

FRAME ERROR CONTROL FIELD CODING PROCEDURES

(NORMATIVE)

B1 CRC-16 FRAME ERROR CONTROL FIELD CODING PROCEDURES

B1.1 CRC-16 FECF ENCODING PROCEDURE

B1.1.1 The Frame Error Control Field shall be computed by applying Cyclic Redundancy Check (CRC) techniques.

B1.1.2 The Encoding Procedure shall accept an $(n-16)$ -bit Transfer Frame, excluding the Frame Error Control Field, and generate a systematic binary $(n,n-16)$ block code by appending a 16-bit Frame Error Control Field as the final 16 bits of the codeword, where n is the length of the Transfer Frame.

NOTE – The Bit Numbering Convention as specified in 1.6.3 is applicable below.

B1.1.3 The equation for the contents of the Frame Error Control Field is:

$$\begin{aligned} \text{FECF} &= [(X^{16} \cdot M(X)) + (X^{(n-16)} \cdot L(X))] \text{ modulo } G(X) \\ &= P_0 \cdot X^{15} + P_1 \cdot X^{14} + P_2 \cdot X^{13} + \dots + P_{14} \cdot X^1 + P_{15} \cdot X^0 \end{aligned}$$

where

all arithmetic is modulo 2;

FECF is the 16-bit Frame Error Control Field with the first bit transferred being the most significant bit P_0 taken as the coefficient of the highest power of X ;

n is the number of bits in the encoded message;

$M(X)$ is the $(n-16)$ -bit information message to be encoded expressed as a polynomial with binary coefficients, with the first bit transferred being the most significant bit M_0 taken as the coefficient of the highest power of X ;

$L(X)$ is the presetting polynomial given by

$$L(X) = \sum_{i=0}^{15} X^i ;$$

$G(X)$ is the generating polynomial given by

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

NOTES

- 1 The $X^{(n-16)} \cdot L(X)$ term has the effect of presetting the shift register to all '1' state prior to encoding.
- 2 A possible FECF generator implementation is shown in figure B-1. For each frame, the shift register cells are initialized to '1'. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the sixteen FECF bits.

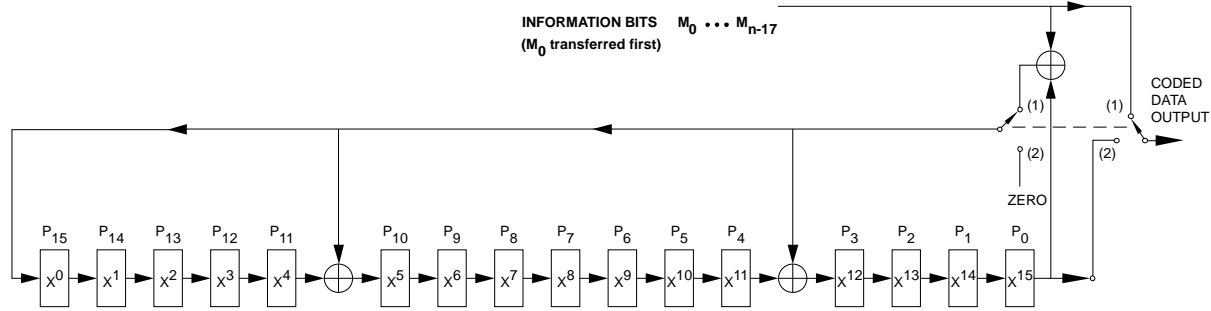


Figure B-1: Logic Diagram of the CRC-16 Encoder

B1.2 CRC-16 FECF DECODING PROCEDURE

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

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

where

- $C^*(X)$ is the received block, including the Frame Error Control Field, in polynomial form, with the first bit transferred being the most significant bit C_0^* taken as the coefficient of the highest power of X ; and
- $S(X)$ is the syndrome polynomial, which will be zero if no error is detected and non-zero if an error is detected, with the most significant bit S_0 taken as the coefficient of the highest power of X .

The received block $C^*(X)$ equals the transmitted codeword $C(X)$ plus (modulo two) the n -bit error block $E(X)$, $C^*(X) = C(X) + E(X)$, where both are expressed as polynomials of the same form, i.e., with the most significant bit C_0 or E_0 taken as the binary coefficient of the highest power of X .

NOTE – A possible syndrome polynomial generator implementation is shown in figure B-2. For each frame, the shift register cells are initialized to ‘1’. The frame includes n bits, i.e., $(n-16)$ information message bits plus the 16 bits of the FECF. All the n bits of the frame are clocked into the input and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be ‘zero’. A non-zero content indicates an erroneous block.

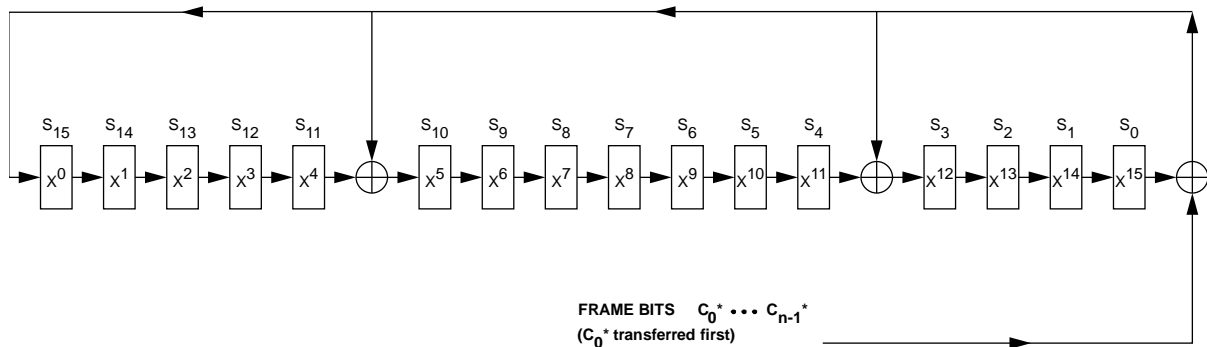


Figure B-2: Logic Diagram of the CRC-16 Decoder

B2 CRC-32 FRAME ERROR CONTROL FIELD CODING PROCEDURES

B2.1 CRC-32 ENCODING PROCEDURE

B2.1.1 For the encoding procedure, the $(n-32)$ -bit Transfer Frame shall be the information message.

B2.1.2 The encoding procedure shall accept an $(n-32)$ -bit Transfer Frame and generate a systematic binary $(n, n-32)$ block code by appending a 32-bit Cyclic Redundancy Check (CRC-32) as the final 32 bits of the FECF.

NOTES

- 1 The Bit Numbering Convention as specified in 1.6.3 is applicable below.
- 2 The Attached Sync Marker (ASM) is NOT used for computing the CRC-32.

B2.1.3 If $M(X)$ is the $(n-32)$ -bit information message to be encoded expressed as a polynomial with binary coefficients, with the first bit transferred being the most significant bit M_0 taken as the coefficient of the highest power of X , then the equation for the 32-bit Cyclic Redundancy Check, expressed as a polynomial $R(X)$ with binary coefficients, shall be:

$$R(X) = [X^{32} \cdot M(X)] \text{ modulo } G(X)$$

where $G(X)$ is the generating polynomial given by:

$$G(X) = X^{32} + X^{23} + X^{21} + X^{11} + X^2 + 1$$

and where the first transferred bit of the Cyclic Redundancy Check is the most significant bit R_0 taken as the coefficient of the highest power of X .

B2.1.4 The n -bit CRC-32-encoded block, expressed as a polynomial $C(X)$ with binary coefficients, shall be:

$$C(X) = X^{32} \cdot M(X) + R(X)$$

The $(n-32)$ bits of the message are input in the order M_0, \dots, M_{n-33} , and the n bits of the codeword are output in the order $C_0, \dots, C_{n-1} = M_0, \dots, M_{n-33}, R_0, \dots, R_{31}$.

NOTE – A possible implementation of an encoder is described in figure B-3. For each frame, the shift register is preset to the ‘all zero’ state prior to encoding. This initialization differs from that performed for the 16-bit CRC described in other CCSDS books, for which the cells are initialized to all ‘ones’. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the 32 Cyclic Redundancy Check bits.

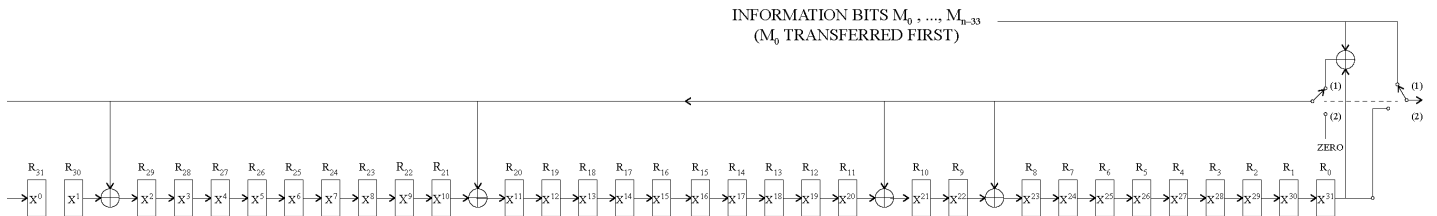


Figure B-3: A Possible Implementation of the CRC-32 Encoder

B2.2 CRC-32 FECF DECODING PROCEDURE

B2.2.1 The decoding procedure shall accept an n -bit received codeword, including the 32-bit Cyclic Redundancy Check, and generate a 32-bit syndrome.

B2.2.2 An error shall be detected if and only if at least one of the syndrome bits is non-‘zero’.

B2.2.3 The received block $C^*(X)$ shall equal the transmitted codeword $C(X)$ plus (modulo two) the n -bit error block $E(X)$, $C^*(X) = C(X) + E(X)$, where both are expressed as polynomials of the same form, i.e., with the most significant bit C_0 or E_0 taken as the binary coefficient of the highest power of X .

B2.2.4 With $C^*(X)$ being the n -bit received codeword with the first transferred bit being the most significant bit C_0^* taken as the coefficient of the highest power of X , then the equation for the 32-bit syndrome, expressed as a polynomial $S(X)$ with binary coefficients, shall be:

$$S(X) = [X^{32} \cdot C^*(X)] \text{ modulo } G(X)$$

The syndrome polynomial will be ‘zero’ if no error is detected, and non-‘zero’ if an error is detected, with the most significant bit S_0 taken as the coefficient of the highest power of X .

NOTE – A possible implementation of the syndrome polynomial generator is described in figure B-4. For each frame, the shift register cells are initialized to ‘zero’. This initialization differs from that performed for the 16-bit CRC described in other CCSDS books, for which the cells are initialized to ‘all ones’. The codeword includes n bits, i.e., $(n-32)$ information message bits plus the 32 bits of the Cyclic Redundancy Check. All the n bits of the codeword are clocked into the input and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be zero. A non-zero content indicates an erroneous block.

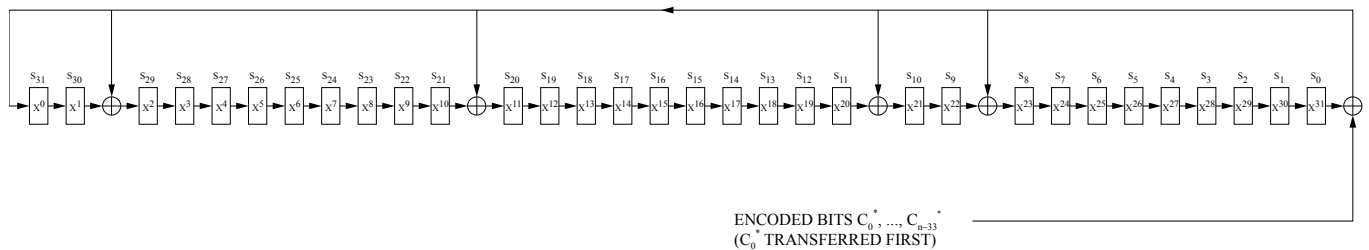


Figure B-4: A Possible Implementation of the CRC-32 Decoder

ANNEX C

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

C1 SECURITY CONSIDERATIONS

[To be supplied.]

C2 SANA CONSIDERATIONS

C2.1 GENERAL

The SLS-SLP WG requests that SANA implement new registries and modifications to existing registries as described below.

C2.2 NEW REGISTRY REQUESTS

C2.2.1 USLP Protocol Identifier (UPID)

Rational: This new registry will have the same type of structure as other Protocol IDs in CCSDS. It is a unique ID to USLP and therefore it cannot share the name space of an existing protocol ID within the existing SANA registries.

Description: It identifies the CCSDS data link layer recognized Protocol or data contained within the USLP Transfer Frame Data Zone. It is a 5 bit mandatory field in the USLP Transfer Frame Data Field Header.

Registration Policy

[CCSDSBlue](#)

Review authority

[SLS-SLP](#)

Reference

CCSDS 732.1-R-1

Data Type - Integer

Protocol Identifier	Description	Status	Reference
00000	COP-1 directives		CCSDS 232.1-B-2
00001	COP-P directives		CCSDS 211.0-B-5
00010	SDLS directives		SDLS Extended Procedures - TBD
00011	User-defined Stream Data	Unassigned	
00100	Mission Unique directives	Unassigned	
00101	Idle Octets	Unassigned	
11111	Extended Protocol ID for USLP		SANA registry USLP_Ext – TBD

C2.2.2 USLP Extended Protocol ID (UPID_Ext)

Rational: This new registry will have the same type of structure as other Protocol IDs in CCSDS e.g., Encapsulation Service. It is a unique ID to USLP and therefore it cannot share the name space of an existing protocol ID within the existing SANA registries.

Description: It identifies the CCSDS data link layer recognized Protocol or data contained within the USLP Transfer Frame Data Zone. It is an 8 bit optional field in the USLP Transfer Frame Data Field Header.

Registration Policy

[CCSDSBlue](#)

Review authority

[SLS-SLP](#)

Reference

CCSDS 732.1-R-1

Data Type - Integer

Protocol Identifier	Status	Reference
00000000-11111111	Unassigned	

C2.3 MODIFIED REGISTRIES

C2.3.1 CCSDS Abbreviations

Add:

Abbreviation: USLP – Unified Space Data Link Protocol

Reference: CCSDS 732.1-R-1

C2.3.2 CCSDS Terms

Add:

Isochronous: characterized by occurring at equal intervals of time.

C2.3.3 MAP ID

For TFCN = 4, the MAP ID is constrained to values ‘0000’ – ‘1111’

C2.3.4 Spacecraft Identifier

NOTE – Version 4 (VN = 1100); SCID = 16 bits; GSCID = VN.SCID = 20 bits;

Version 4 Range: SCID = [0x0000 – 0xFFFF]; GSCID = [C0000 - CFFFF]

C2.3.5 Transfer Frame Version Number

Version Number = ‘1100’; Transfer Frame = USLP Version 4; Reference = CCSDS 732.1-R-1

C2.3.6 Space Link Identifiers Registry

Add new registries under Space Link Identifiers for a) UPID and b) UPID_Ext

NOTE – There exists two mistakes in the Space Link Identifiers Registry agnostic to the USLP. Please fix the following two names to be consistent with the existing registries. They are:

- From “Extended Protocol Identifier” to “Extended Protocol Identifier for Encapsulation Service”
- From “Protocol Identifier” to “Protocol Identifier for Encapsulation Service”

If you make those two changes, then the existing registries and the Space Link Identifiers Registry will be consistent.

C2.3.7 VCID

Add USLP Protocol to this registry:

VCID = 000000 – 111111

Status: Unmanaged

Reference: 732.1-R-1

C3 PATENT CONSIDERATIONS

[To be supplied.]

ANNEX D

INFORMATIVE REFERENCES

(INFORMATIVE)

- [D1] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 4. CCSDS Record (Yellow Book), CCSDS A02.1-Y-4. Washington, D.C.: CCSDS, April 2014.
- [D2] *Overview of Space Communications Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-3. Washington, D.C.: CCSDS, July 2014.
- [D3] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.
- [D4] *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.
- [D5] *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.
- [D6] *TC Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-3. Washington, D.C.: CCSDS, September 2015.
- [D7] *The Application of CCSDS Protocols to Secure Systems*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-2. Washington, D.C.: CCSDS, January 2006.
- [D8] *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services*. ETSI EN 300 421 V1.1.2 (1997-08). Sophia-Antipolis: ETSI, 1997.
- [D9] *A 48/56/64 kbit/s Data Circuit-Terminating Equipment Standardized for Use on Digital Point-to-Point Leased Circuits*. ITU-T Recommendation V.38. Geneva: ITU, 1996.
- [D10] *Performance Characteristics for Intermediate Data Rate Digital Carriers Using Convolutional Encoding/Viterbi Encoding*. Rev. 10. IESS 308. Washington, DC: INTELSAT, 2000.

- [D11] *Unified Space Link Protocol*. Report Concerning Space Data System Standards (Green Book), CCSDS 700.1-G. Forthcoming.
- [D12] *Advanced Orbiting Systems, Networks and Data Links: Summary of Concept, Rationale and Performance*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 700.0-G-3. Washington, D.C.: CCSDS, November 1992.
- [D13] *TC Synchronization and Channel Coding—Summary of Concept and Rationale*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 230.1-G-2. Washington, D.C.: CCSDS, November 2012.
- [D14] *Proximity-1 Space Link Protocol—Rationale, Architecture, and Scenarios*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 210.0-G-2. Washington, D.C.: CCSDS, December 2013.
- [D15] *Telecommand Summary of Concept and Rationale*. Issue 6. Report Concerning Space Data System Standards (Green Book), CCSDS 200.0-G-6. Washington, D.C.: CCSDS, January 1987.
- [D16] *TM Synchronization and Channel Coding—Summary of Concept and Rationale*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 130.1-G-2. Washington, D.C.: CCSDS, November 2012.
- [D17] *Next Generation Uplink*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 230.2-G-1. Washington, D.C.: CCSDS, July 2014.
- [D18] *Space Data Link Protocols—Summary of Concept and Rationale*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.2-G-3. Washington, D.C.: CCSDS, September 2015.
- [D19] *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015.
- [D20] *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015.

NOTE – Normative references are listed in 1.7.

ANNEX E

PROXIMITY-1 VARIABLE-LENGTH SUPERVISORY PROTOCOL DATA FIELD FORMATS

(INFORMATIVE)

NOTES

- 1 Reference [10] contains the normative specification for Proximity-1 variable-length Supervisory Protocol Data Field formats.
- 2 Table F-2 should be consulted for a complete overview of the variable-length SPDU structure including the SPDU header and SPDU data field. This annex specifies the format of the data field only.
- 3 The Directive Type field is defined from bits 13 through 15, inclusive, in order to maintain backward compatibility with the NASA Mars Surveyor Project 2001 Odyssey orbiter.

E1 SPDU TYPE 1: DIRECTIVE/REPORT/PLCW SPDU DATA FIELD

E1.1 GENERAL

E1.1.1 The Directive/Report/PLCW SPDU is used for space link supervisory configuration and control of the transceiver and its operation.

E1.1.2 The SPDU data field is a container that can hold up to seven sixteen-bit discrete self-delimiting and self-identifying directives:

- a) each directive has a specific functionality;
- b) each directive is sixteen bits in length and is self-identified by the value in the Directive Type field (contained in bits 13, 14, and 15 of the directive);
- c) the directives are concatenated without intervening bits within the data field.

NOTE – Figure E-1 shows the Type 1 SPDU Data Field Contents.

										Directive Type 3 bits (13,14,15)
Mode (0,1,2)		Data Rate (3,4,5,6)		Modulation (7)		Data Encoding (8,9)		Frequency (10,11,12)		'000' =SET TRANSMITTER PARAMETERS
Time Sample (0,1,2,3,4,5)		Duplex (6,7,8)	Reserved (9,10)	Remote No More Data (11)			Token (12)		'001' = SET CONTROL PARAMETERS	
Mode (0,1,2)		Data Rate (3,4,5,6)		Modulation (7)		Data Decoding (8,9)		Frequency (10,11,12)		'010' = SET RECEIVER PARAMETERS
Receiver Frame Sequence Number (SEQ_CTRL_FSN)(0,1,2,3,4,5,6,7)						Reserved (8,9,10,11,12)			'011' = Set V(R)	
Reserved (0,1,2)		Status Report Request (3,4,5,6,7)		Time-Tag Request (8,9,10)		PCID 0 PLCW Request (11)		PCID 1 PLCW Request (12)		'100' = Report Request
										'101' = Reserved
Dirac tion (0)	Freq Table (1)	Rate Table (2)	Carrier Mod (3,4)	Data Mod (5,6)	Mode Select (7,8)	scrambler (9,10)	Diff. Enco ding (11)	R-S Code (12)	'110'=SET PL EXTENSIONS	
Source Spacecraft ID (0,1,2,3,4,5,6,7,8,9)					Reserved (10,11,12)				'111' = Report Source SCID	

Figure E-1: Type 1 SPDU Data Field Contents**E1.2 SET TRANSMITTER PARAMETERS DIRECTIVE****E1.2.1 General**

The SET TRANSMITTER PARAMETERS directive consists of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- Directive Type (3 bits);
- Transmitter Frequency (3 bits);
- Transmitter Data Encoding (2 bits);
- Transmitter Modulation (1 bit);
- Transmitter Data Rate (4 bits);
- Transmitter (TX) Mode (3 bits).

NOTE – The structural components of the SET TRANSMITTER PARAMETERS directive are shown in figure E-2.

Bit 0			Bit 15		
TX Mode 3 bits	TX Data Rate 4 bits	TX Modulation 1 bit	TX Data Encoding 2 bits	TX Frequency 3 bits	Directive Type 3 bits
0,1,2	3,4,5,6	7	8,9	10,11,12	13,14,15

Figure E-2: SET TRANSMITTER PARAMETERS Directive**E1.2.2 Directive Type**

E1.2.2.1 Bits 13–15 of the SET TRANSMITTER PARAMETERS directive contains the Directive Type.

E1.2.2.2 The 3-bit Directive Type field identifies the type of protocol control directive and contains the binary value ‘000’ for the SET TRANSMITTER PARAMETERS directive.

E1.2.3 Transmitter Frequency**E1.2.3.1 General**

Bits 10–12 of the SET TRANSMITTER PARAMETERS directive are used to set the transmitter frequency of the partnered transceiver to the desired value.

E1.2.3.2 Return Transmitter Frequency (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field defines the transmit frequency of the responder. Actual frequency assignments are given in the Physical Layer (see reference [15]).

‘000’	‘001’	‘010’	‘011’	‘100’	‘101’	‘110’	‘111’
Ch0R	Ch1R	Ch2R	Ch3R	Ch4R	Ch5R	Ch6R	Ch7R

E1.2.4 Transmitter Data Encoding

Bits 8–9 of the SET TRANSMITTER PARAMETERS directive contain the following coding options:

- a) ‘00’ = LDPC(2048,1024) rate 1/2 code (see reference [3]);
- b) ‘01’ = Convolutional Code(7,1/2) (G2 vector inverted) with attached CRC-32 (see reference [7]).

- c) '10' = Bypass all codes;
- d) '11' = Concatenated (R-S(204,188), CC(7,1/2)) Codes.

NOTE – R-S(204,188) with CC(7,1/2) code is an ETSI standard. This option is not required for cross support. (See reference [D8] for more details.)

E1.2.5 Transmitter Modulation

Bit 7 of the SET TRANSMITTER PARAMETERS directive contains the transmission modulation options:

- a) '0' = Coherent frequency PSK;
- b) '1' = Non-coherent frequency PSK.

E1.2.6 Transmitter Data Rate

E1.2.6.1 Bits 3–6 of the SET TRANSMITTER PARAMETERS directive contain one of the following transmission data rates (rates in kb/s, e.g., 4 = 4000 b/s) prior to encoding.

NOTE – Because of the NASA Mars Surveyor Project 2001 Odyssey implementation, there is an added constraint on the use of the values in the Data Rate field for 8, 32, 128, 256 kb/s. Data rate selection is linked to the modulation field value as shown in the tables below. NC indicates non-coherent PSK, and C indicates coherent PSK. R1 through R4 indicate the field is reserved for future definition by the CCSDS. 1, 512, 1024, and 2048 kb/s data rates can only be selected using the SET PL EXTENSIONS directive (see E1.7).

E1.2.6.2 Ordered by Data Rate:

'1000'	'1001'	'0000'	'0001'	'1100'	'0010'	'0011'	'1101'	'0100'	'0101'	'0110'	'0111'	'1010'	'1011'	'1110'	'1111'
2	4	8 NC	8 C	16	32 NC	32 C	64	128 NC	128 C	256 NC	256 C	R1	R2	R3	R4

E1.2.6.3 Ordered by Bit pattern:

'0000'	'0001'	'0010'	'0011'	'0100'	'0101'	'0110'	'0111'	'1000'	'1001'	'1010'	'1011'	'1100'	'1101'	'1110'	'1111'
8 NC	8 C	32 NC	32 C	128 NC	128 C	256 NC	256 C	2	4	R1	R2	16	64	R3	R4

E1.2.6.4 Proximity-1 coded symbol (R_{cs}) and data rate (R_d) table:

Prox-1 Coded Symbol Rates (R_{cs})	Prox-1 Uncoded Data Rates (R_d) $R_d = R_{cs}$	Prox-1 Convolutionally Coded Data Rates (R_d) $R_d = .5 * R_{cs}$	Prox-1 LDPC computed data rates (R_d) $R_d = .48484 * R_{cs}$
1000	1000	N/A	N/A
2000	2000	1000	969.6969697
4000	4000	2000	1939.393939
8000	8000	4000	3878.787879
16000	16000	8000	7757.575758
32000	32000	16000	15515.15152
64000	64000	32000	31030.30303
128000	128000	64000	62060.60606
256000	256000	128000	124121.2121
512000	512000	256000	248242.4242
1024000	1024000	512000	496484.8485
2048000	2048000	1024000	992969.697
4096000	N/A	2048000	1985939.394

E1.2.6.5 1, 512, 1024, and 2048 kb/s data rates can only be selected using the SET PL EXTENSIONS directive (see E1.7).

E1.2.7 Transmitter Mode

Bits 0–2 of the SET TRANSMITTER PARAMETERS directive contain the Transmitter Mode options. This field identifies the operating mode of the transmitter:

- a) '000' = Mission Specific;
- b) '001' = Proximity-1 Protocol;
- c) '010' = Mission Specific;
- d) '011' = Mission Specific;
- e) '100' = Mission Specific;
- f) '101' = Mission Specific;
- g) '110' = Reserved by CCSDS;
- h) '111' = Reserved by CCSDS.

NOTE – Reference [10] annex F should be consulted for NASA Mars Survey Project 2001 Odyssey Orbiter Transmitter Mode mission-specific values.

E1.3 SET CONTROL PARAMETERS

E1.3.1 General

E1.3.1.1 The SET CONTROL PARAMETERS directive consists of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) Token (1 bit);
- c) Remote No More Data (1 bit);
- d) Reserved (2 bits);
- e) Duplex (3 bits);
- f) Time Sample (6 bits).

E1.3.1.2 This directive is used to set from zero to four control parameters at a time: 1) setting the token for half-duplex operations; 2) setting the Remote No More Data condition for session termination in full or half duplex; 3) setting the Duplex parameter; 4) setting the number of time samples to be taken during timing services.

NOTE – The structural components of the SET CONTROL PARAMETERS directive are shown in figure E-3.

Bit 0			Bit 15		
Time Sample 6 bits	Duplex 3 bits	Reserved 2 bits	Remote No More Data 1 bit	Token 1 bit	Directive Type 3 bits
0,1,2,3,4,5	6,7,8	9,10	11	12	13,14,15

Figure E-3: SET CONTROL PARAMETERS Directive

E1.3.2 Directive Type

E1.3.2.1 Bits 13–15 of the SET CONTROL PARAMETERS directive contain the Directive Type.

E1.3.2.2 The 3-bit Directive Type field identifies the type of protocol control directive and contains the binary value ‘001’ to identify the SET CONTROL PARAMETERS directive.

E1.3.3 Token

Bit 12 of the SET CONTROL PARAMETERS directive contains the value of the Token field as follows:

- a) '0' = No Change;
- b) '1' = Transmit.

NOTE – This field either notifies the remote node that there is no change in who has permission to transmit (i.e., ignore this field) or commands the remote node to the transmit state.

E1.3.4 Remote No More Data

Bit 11 of the SET CONTROL PARAMETERS directive contains the Remote No More Data field as follows:

- a) '0' = No Change;
- b) '1' = Remote Node has No More Data to Send (RNMD).

NOTE – This field either notifies the recipient node that there is no change in the remote node's data state (i.e., ignore this field) or notifies the recipient node that the remote node has no more data to send, in which case the session may be terminated when the recipient node locally has no more data to send.

E1.3.5 Reserved

Bits 9-10 of the SET CONTROL PARAMETERS directive contain spares and are set to 'all zero'.

E1.3.6 Duplex

Bits 6-8 of the SET CONTROL PARAMETERS directive contain the Duplex field as follows:

- a) '000' = No Change;
- b) '001' = Full Duplex;
- c) '010' = Half Duplex;
- d) '011' = Simplex Transmit;
- e) '100' = Simplex Receive;
- f) '101' = Reserved;

- g) '110' = Reserved;
- h) '111' = Reserved.

NOTE – This field either notifies the recipient node that there is no change in the remote node's Duplex state (i.e., ignore this field) or notifies the recipient node to change the directionality of communication accordingly.

E1.3.7 Time Sample

Bits 0-5 of the SET CONTROL PARAMETERS directive contain the Time Sample field. When this field is non-zero, it notifies the recipient to capture the time and frame sequence number (associated with the Proximity timing service—see reference [10] section 5 for the next n frames received, where n is the number of Proximity Transfer Frames contained within the Time Sample Field).

E1.4 SET RECEIVER PARAMETERS DIRECTIVE

E1.4.1 General

The SET RECEIVER PARAMETERS directive consists of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) Receiver Frequency (3 bits);
- c) Receiver Data Decoding (2 bits);
- d) Receiver Modulation (1 bit);
- e) Receiver Data Rate (4 bits);
- f) Receiver (RX) Mode (3 bits).

NOTE – The structural components of the SET RECEIVER PARAMETERS directive are shown in figure E-4.

Bit 0			Bit 15		
RX Mode 3 bits	RX Rate 4 bits	RX Modulation 1 bit	RX Data Decoding 2 bits	RX Frequency 3 bits	Directive Type 3 bits
0,1,2	3,4,5,6	7	8,9	10,11,12	13,14,15

Figure E-4: SET RECEIVER PARAMETERS Directive

E1.4.2 Directive Type

E1.4.2.1 Bits 13–15 of the SET RECEIVER PARAMETERS directive contains the Directive Type.

E1.4.2.2 The 3-bit Directive Type field identifies the type of protocol control directive and contains the binary value ‘010’ for the SET RECEIVER PARAMETERS directive.

E1.4.3 Receiver Frequency

E1.4.3.1 General

Bits 10–12 of the SET RECEIVER PARAMETERS directive are used to set the receiver frequency of the partnered transceiver to the desired value.

E1.4.3.2 Forward Receive Frequency (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field defines the receive frequency of the responder. Actual frequency assignments are given in the Physical Layer (see reference [15]).

‘000’	‘001’	‘010’	‘011’	‘100’	‘101’	‘110’	‘111’
Ch0F	Ch1F	Ch2F	Ch3F	Ch4F	Ch5F	Ch6F	Ch7F

E1.4.4 Receiver Data Decoding

Bits 8–9 of the SET RECEIVER PARAMETERS directive contain the following coding options:

‘00’ = LDPC (2048,1024) rate 1/2 code (see reference [3]);

‘01’ = Convolutional Code(7,1/2) (G2 vector inverted) with attached CRC-32 (see reference [7]);

‘10’ = Bypass all codes;

‘11’ = Concatenated R-S(204,188), CC(7,1/2).

NOTE – R-S(204,188) with CC(7,1/2) code is an ETSI standard. This option is not required for cross support. (See reference [D8] for more details.)

E1.4.5 Receiver Modulation

Bit 7 of the SET RECEIVER PARAMETERS directive contains the following transmission modulation options:

- a) '0' = Coherent frequency PSK;
- b) '1' = Non-coherent frequency PSK.

E1.4.6 Receiver Data Rate

E1.4.6.1 Bits 3–6 of the SET RECEIVER PARAMETERS directive contain one of the following receiver data rates (rates in kb/s, e.g., 4 = 4000 b/s) after decoding.

NOTE – Because of the NASA Mars Surveyor Project 2001 Odyssey implementation, there is an added constraint on the use of the values in the Data Rate field for 8, 32, 128, and 256 kb/s. Data rate selection is linked to the modulation field value as shown in the tables below ('NC' indicates non-coherent, and c indicates coherent). R1 through R4 indicates the field is reserved for future definition by the CCSDS. 1, 512, 1024, and 2048 kb/s data rates can only be selected using the SET PL EXTENSIONS directive (see E1.7).

E1.4.6.2 Ordered by Data Rate:

'1000'	'1001'	'0000'	'0001'	'1100'	'0010'	'0011'	'1101'	'0100'	'0101'	'0110'	'0111'	'1010'	'1011'	'1110'	'1111'
2	4	8 NC	8 C	16	32 NC	32 C	64	128 NC	128 C	256 NC	256 C	R1	R2	R3	R4

E1.4.6.3 Ordered by Bit pattern:

'0000'	'0001'	'0010'	'0011'	'0100'	'0101'	'0110'	'0111'	'1000'	'1001'	'1010'	'1011'	'1100'	'1101'	'1110'	'1111'
8 NC	8 C	32 NC	32 C	128 NC	128 C	256 NC	256 C	2	4	R1	R2	16	64	R3	R4

E1.4.6.4 Proximity-1 coded symbol (R_{cs}) and data rate (R_d) table:

Prox-1 Coded Symbol Rates (R_{cs})	Prox-1 Uncoded Data Rates (R_d) $R_d = R_{cs}$	Prox-1 Convolutionally Coded Data Rates (R_d) $R_d = .5 * R_{cs}$	Prox-1 LDPC Computed Data Rates (R_d) $R_d = .48484 * R_{cs}$
1000	1000	N/A	N/A
2000	2000	1000	969.6969697
4000	4000	2000	1939.393939
8000	8000	4000	3878.787879
16000	16000	8000	7757.575758
32000	32000	16000	15515.15152
64000	64000	32000	31030.30303
128000	128000	64000	62060.60606
256000	256000	128000	124121.2121
512000	512000	256000	248242.4242
1024000	1024000	512000	496484.8485
2048000	2048000	1024000	992969.697
4096000	N/A	2048000	1985939.394

NOTE – 1, 512, 1024, and 2048 kb/s data rates can only be selected using the SET PL EXTENSIONS directive (see E1.7).

E1.4.7 Receiver Mode

Bits 0–2 of the SET RECEIVER PARAMETERS directive contain the receiver mode options. This field identifies the operating mode of the receiver:

- a) '000' = Mission Specific;
- b) '001' = Proximity-1 Protocol;
- c) '010' = Mission Specific;
- d) '011' = Mission Specific;
- e) '100' = Mission Specific;
- f) '101' = Mission Specific;
- g) '110' = Reserved by CCSDS;
- h) '111' = Reserved by CCSDS.

NOTE – Reference [10] annex F should be consulted for NASA Mars Survey Project 2001 Odyssey Orbiter Receiver Mode mission-specific values.

E1.5 SET V(R) DIRECTIVE

E1.5.1 General

The SET V(R) directive consists of three fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) Spare (5 bits);
- c) Receiver Frame Sequence Number (SEQ_CTRL_FSN) (8 bits).

NOTE – The structural components of the SET V(R) directive are shown in figure E-5.

Bit 0		Bit 15
Receiver Frame Sequence Number SEQ_CTRL_FSN 8 bits	Spare 5 bits	Directive Type 3 bits
0,1,2,3,4,5,6,7	8,9,10,11,12	13,14,15

Figure E-5: SET V(R) Directive

E1.5.2 Directive Type

E1.5.2.1 Bits 13–15 of the SET V(R) directive contain the Directive Type.

E1.5.2.2 The 3-bit Directive Type field identifies the type of protocol control directive and contains the binary value ‘011’ to identify the SET V(R) directive.

E1.5.3 Spare

Bits 8–12 of the SET V(R) directive contain spare bits, set to ‘all zero’.

E1.5.4 Receiver Frame Sequence Number

Bits 0–7 of the SET V(R) directive contain the value of the Frame Sequence Number (SEQ_CTRL_FSN) to which the receiving unit of the partnered transceiver is to be set.

E1.6 REPORT REQUEST DIRECTIVE

E1.6.1 General

The REPORT REQUEST directive is the mechanism by which either (1) a status report, (2) a time-tag, or (3) a PLCW per PCID can be requested of a Proximity-1 node. It consists of seven fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) PCID 1 PLCW Request (1 bit);
- c) PCID 0 PLCW Request (1 bit);
- d) Time-Tag Request (3 bits);
- e) Status Request (5 bits);
- f) Spare (3 bits).

NOTE – The structural components of the REPORT REQUEST directive are shown in figure E-6.

Bit 0			Bit 15		
Spare	Status Report Request	Time-Tag Request	PCID 0 PLCW Request	PCID 1 PLCW Request	Directive Type
3 bits	5 bits	3 bit	1 bit	1 bit	3 bits
0,1,2	3,4,5,6,7	8,9,10	11	12	13,14,15

Figure E-6: Report Request

E1.6.2 Directive Type

E1.6.2.1 Bits 13–15 of the REPORT REQUEST directive contain the Directive Type.

E1.6.2.2 The 3-bit Directive Type field identifies the type of protocol control directive and contains the binary value ‘100’.

E1.6.3 Physical Channel ID 1 PLCW Report Request Field

Bit 12 of the REPORT REQUEST directive indicates whether a PLCW report for PCID 1 is required:

- a) ‘1’ = PLCW report is needed for PCID 1; transmit this report on the same PCID that the report request arrived on.

- b) '0' = PLCW report is not required.

E1.6.4 Physical Channel ID 0 PLCW Report Request Field

Bit 11 of the REPORT REQUEST directive indicates whether a PLCW report for PCID 0 is required:

- a) '1' = PLCW report is needed for PCID 0; Transmit this report on the same PCID that the report request arrived on.
- b) '0' = PLCW report is not required.

E1.6.5 Time-Tag Request Field

Bits 8–10 of the directive, if set to a value other than '000', indicates a request to the remote transceiver to initiate a Proximity-1 time tag exchange (see reference [10] section 5).

E1.6.6 Status Report Request

E1.6.6.1 The value contained in bits 3–7 of the REPORT REQUEST directive indicates the type of status report desired.

E1.6.6.2 If set to '00000', a status report is not required.

E1.6.6.3 The types of status reports are reserved for CCSDS use.

E1.6.7 Spares

Bits 0–2 of the REPORT REQUEST directive contain spare bits set to 'all zero'.

E1.7 SET PL EXTENSIONS

E1.7.1 Overview

The SET PL EXTENSIONS directive is the mechanism by which additional Physical Layer parameters can be enabled or disabled. This directive is transferred across the Proximity link from the local transceiver to the remote transceiver. This directive is provided for compatibility between transceivers with extensions in addition to those discussed in this Proximity-1 Recommended Standard.

E1.7.2 General

The SET PL EXTENSIONS directive consists of ten fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) R-S Code (1 bit);
- c) Differential Mark Encoding (1 bit);
- d) Scrambler (2 bits);
- e) Mode Select (2 bits);
- f) Data Modulation (2 bits);
- g) Carrier Modulation (2 bits);
- h) Rate Table (1 bit);
- i) Frequency Table (1 bit);
- j) Direction (1 bit).

NOTE – The structural components of the SET PL EXTENSIONS directive are shown in figure E-7.

Bit 0					Bit 15				
Direction	Freq Table	Rate Table	Carrier MOD	Data MOD	Mode Select	Scrambler	Differential Mark Encoding	R-S Code	Directive Type
1 bit	1 bit	1 bit	2 bits	2 bits	2 bits	2 bits	1 bit	1 bit	3 bits
(0)	(1)	(2)	(3,4)	(5,6)	(7,8)	(9,10)	(11)	(12)	13,14,15

Figure E-7: SET PL EXTENSIONS

E1.7.3 Directive Type

E1.7.3.1 Bits 13–15 of the SET PL EXTENSIONS directive contain the Directive Type.

E1.7.3.2 The 3-bit Directive Type field identifies the directive type and contains the binary value ‘110’.

E1.7.4 Reed-Solomon Code

Bit 12 of the SET PL EXTENSIONS directive indicates which R-S Code is used:

- a) '0' = R-S(204,188) code;
- b) '1' = R-S(255,239) code.

Neither of these R-S Codes is specified by CCSDS in other Recommended Standards and therefore they are not required for cross-support.

E1.7.5 Differential Mark Encoding

Bit 11 of the SET PL EXTENSIONS directive indicates whether Differential Mark Encoding is enabled:

- a) '0' = No differential encoding;
- b) '1' = Differential encoding enabled. The current data bit is exclusive ORed with the previously transmitted bit to determine the value of the current transmitted bit. When the current data bit is a '1', then the current encoder output bit level changes relative to the previous output value. If the data bit is a '0', then the current encoder output bit level remains constant relative to the previous output value (see table below).

Option b) is not required for cross-support except for those missions required to interoperate with NASA MRO (see reference [10] annex G).

E1.7.6 Scrambler

Bits 9-10 of the SET PL EXTENSIONS directive indicate if and what type of digital bit scrambling is used:

- a) '00' = Bypass all bit scrambling;
- b) '01' = CCITT bit scrambling enabled (see reference [D9]);
- c) '10' = Bypass all bit scrambling;
- d) '11' = IESS bit scrambling enabled (see reference [D10]);

None of these Scrambler options are specified by CCSDS in other Recommended Standards and therefore they are not required for cross-support.

E1.7.7 Mode Select

Bits 7-8 of the SET PL EXTENSIONS directive indicate the type of carrier suppression used:

- a) '00' = Suppressed Carrier (Requires transmit side utilize Modulation Index of 90° and transmit/receive sides utilize Differential Mark Encoding/Decoding);
- b) '01' = Residual Carrier;
- c) '10' = Reserved;
- d) '11' = Reserved.

Option a) is not required for cross-support except for those missions required to interoperate with NASA MRO (see reference [10] annex G).

E1.7.8 Data Modulation

Bits 5-6 of the SET PL EXTENSIONS directive indicate the type of data modulation used:

- a) '00' = NRZ-L;
- b) '01' = Bi-Phase-Level (Manchester);
- c) '10' = Reserved;
- d) '11' = Reserved.

Option a) is not required for cross-support except for those missions required to interoperate with NASA MRO (see reference [10] annex G).

E1.7.9 Carrier Modulation

Bits 3-4 of the SET PL EXTENSIONS directive indicate the type of carrier modulation to be used:

- a) '00' = No Modulation;
- b) '01' = PSK;
- c) '10' = FSK;
- d) '11' = QPSK.

Options c) and d) are not required for cross-support.

E1.7.10 Rate Table

E1.7.10.1 Bit 2 of the SET PL EXTENSIONS directive indicates which set of data rates prior to encoding is to be used.

- a) '0' = Default Set defined in the Data Rate Field of the SET TRANSMITTER PARAMETERS and SET RECEIVER PARAMETERS Directives in this annex;

b) '1' = Extended Physical Layer Data Rate Set defined below.

'0000' = 1000 b/s	'0100' = 16000 b/s	'1000' = 256000 b/s	'1100' = Reserved
'0001' = 2000 b/s	'0101' = 32000 b/s	'1001' = 512000 b/s	'1101' = Reserved
'0010' = 4000 b/s	'0110' = 64000 b/s	'1010' = 1024000 b/s	'1110' = Reserved
'0011' = 8000 b/s	'0111' = 128000 b/s	'1011' = 2048000 b/s	'1111' = Reserved

Option a) is required for cross-support. Option b) is required for cross-support for data rates less than 2000 b/s and greater than 256000 b/s.

E1.7.10.2 Proximity-1 coded symbol (R_{cs}) and data rate (R_d) table:

Prox-1 Coded Symbol Rates (R_{cs})	Prox-1 Uncoded Data Rates (R_d) $R_d = R_{cs}$	Prox-1 Convolutionally Coded Data Rates (R_d) $R_d = .5 * R_{cs}$	Prox-1 LDPC Computed Data Rates (R_d) $R_d = .48484 * R_{cs}$
1000	1000	N/A	N/A
2000	2000	1000	969.6969697
4000	4000	2000	1939.393939
8000	8000	4000	3878.787879
16000	16000	8000	7757.575758
32000	32000	16000	15515.15152
64000	64000	32000	31030.30303
128000	128000	64000	62060.60606
256000	256000	128000	124121.2121
512000	512000	256000	248242.4242
1024000	1024000	512000	496484.8485
2048000	2048000	1024000	992969.697
4096000	N/A	2048000	1985939.394

E1.7.11 Frequency Table

E1.7.11.1 General

Bit 1 of the SET PL EXTENSIONS directive indicates what set of frequencies is to be used:

- '0' = Channels 0 – 7 defined in the Frequency Field of the SET TRANSMITTER PARAMETERS and SET RECEIVER PARAMETERS Directives and specifically in the Proximity-1 Physical Layer;
- '1' = Channels 8 – 15 defined in the Extended Physical Layer Frequency Set defined below.

E1.7.11.2 Forward Link (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field defines the receive frequency of the responder. Actual frequency assignments are given in the Physical Layer (see reference [15]).

'000'	'001'	'010'	'011'	'100'	'101'	'110'	'111'
Ch8F	Ch9F	Ch10F	Ch11F	Ch12F	Ch13F	Ch14F	Ch15F

E1.7.11.3 Return Link (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the return link, this 3-bit field defines the transmit frequency of the responder. Actual frequency assignments are given in the Physical Layer (see reference [15]).

'000'	'001'	'010'	'011'	'100'	'101'	'110'	'111'
Ch8R	Ch9R	Ch10R	Ch11R	Ch12R	Ch13R	Ch14R	Ch15R

E1.7.12 Direction

Bit 0 of the SET PL EXTENSIONS directive indicates if the fields in this directive apply to the transmit or receive side of the transceiver:

- a) '0' = transmit side;
- b) '1' = receive side.

E1.8 REPORT SOURCE SPACECRAFT ID**E1.8.1 General**

The REPORT SOURCE SPACECRAFT ID is the mechanism by which the local transceiver can provide status of its source spacecraft ID to the remote transceiver across the Proximity link. It consists of three fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

- a) Directive Type (3 bits);
- b) Reserved (3 bits);
- c) Source Spacecraft ID (10 bits).

NOTE – The structural components of the REPORT SOURCE SPACECRAFT ID are shown in figure E-8.

Bit 0			Bit 15
	Source Spacecraft ID	Reserved	Directive Type
	10 bits	3 bits	3 bits
	0,1,2,3,4,5,6,7,8,9	10,11,12	13,14,15

Figure E-8: Report Source Spacecraft ID**E1.8.2 Directive Type**

E1.8.2.1 Bits 13–15 of the REPORT SOURCE SPACECRAFT ID status report contains the Directive Type.

E1.8.2.2 The 3-bit Directive Type field identifies the type of status report and contains the binary value ‘111’.

E1.8.3 Reserved

Bits 10–12 of the REPORT SOURCE SPACECRAFT ID status report contains reserved bits, set to ‘all zero’.

E1.8.4 Source Spacecraft ID

Bits 0-9 of the REPORT SOURCE SPACECRAFT ID status report contains the SCID of the source of the Transfer Frame.

E2 SPDU TYPE 2: TIME DISTRIBUTION SPDU DATA FIELD

NOTE – Table F-2 should be consulted for a complete overview of the variable-length SPDU structure including the SPDU header and SPDU data field.

E2.1 GENERAL

E2.1.1 The TIME DISTRIBUTION SPDU data field is the container that describes both the type and value of the time entity for distribution.

E2.1.2 A single TIME DISTRIBUTION directive is contained within a TIME DISTRIBUTION SPDU.

E2.1.3 The format of the TIME DISTRIBUTION SPDU data field consists of four fields, positioned contiguously, in the following sequence:

- a) TIME DISTRIBUTION Directive Type (1 octet);

- b) Transceiver Clock (8 octets);
- c) Send Side Delay (3 octets);
- d) One-Way-Light-Time (3 octets);

All time code fields in this directive use the CCSDS Unsegmented Time Code format (reference [16]).

NOTE – The structural components of the TIME DISTRIBUTION SPDU data field are shown in figure E-9.

Octet 0		Octet 14	
Directive Type	Transceiver Clock	Send Side Delay	One-Way- Light-Time
1 Octet	8 Octets	3 Octets	3 Octets

Figure E-9: Type 2 SPDU Data Field Contents

E2.2 TIME DISTRIBUTION DIRECTIVE TYPE

E2.2.1 Octet 0 of the TIME DISTRIBUTION SPDU data field contains the TIME DISTRIBUTION Directive Type field indicating the function to be performed (if any) with the time contents.

E2.2.2 TIME DISTRIBUTION Types are:

- a) '00000000' = NULL;
- b) '00000001' = TIME TRANSFER;
- c) all others = Reserved for CCSDS use.

E2.3 TRANSCEIVER CLOCK

E2.3.1 When the Time Distribution Type equals TIME TRANSFER,

- a) octet 1 through octet 8 contain the value of the clock corresponding to when the trailing edge of the last bit of the ASM of the transmitted PLTU crosses the clock capture point within the transceiver;
- b) this time code field is divided into 5 octets of coarse time and 3 octets of fine time (see reference [16]).

E2.3.2 Otherwise, this field contains reserved bits, set to 'all zero'.

E2.4 SEND SIDE DELAY

E2.4.1 When the Time Distribution Type equals TIME TRANSFER,

- a) octet 9 through octet 11 contain the delay time between the transceiver internal clock capture point and when the trailing edge of the last bit of the ASM of the transmitted PLTU crossed the time reference point (see reference [10] section 5 ‘Proximity-1 Timing Services’);
- b) this time code field is divided into 1 octet of coarse time and 2 octets of fine time (see reference [4]).

E2.4.2 Otherwise, this field contains reserved bits, set to ‘all zero’.

E2.5 ONE WAY LIGHT TIME

E2.5.1 When the Time Distribution Type equals TIME TRANSFER, and when the mission has decided OWLT should be used,

- a) octet 12 through octet 14 contain the calculated one way light time between the instance that the trailing edge of the last bit of the ASM of the transmitted PLTU crosses the time reference point of the initiator node to the time reference point of the destination node;
- b) this time code field is divided into 1 octet of coarse time and 2 octets of fine time (see reference [4]).

E2.5.2 Otherwise, this field contains reserved bits, set to ‘all zero’.

ANNEX F

PROTOCOL DATA UNIT (CONTROL WORDS/DIRECTIVES) FORMATS (INFORMATIVE)

F1 COMMUNICATIONS LINK CONTROL WORD

F1.1 GENERAL

F1.1.1 The Communications Link Control Word (CLCW), which is the protocol data unit transmitted from the receiving end to the sending end, provides the mechanism by which the FARM at the receiving end reports the status of frame acceptance to the Frame Operation Procedure (FOP) at the sending end.

NOTES

- 1 The controlling specification for how the CLCW is used within the COP is contained in reference [9].
- 2 CLCWs are usually carried in the Operational Control Field of Transfer Frames using the MC_OCF or VC_OCF Service.
- 3 Although it is not necessary that the CLCW reporting rate (from the receiving end to the sending end) match the Transfer Frame transfer rate (from the sending end to the receiving end), some minimum CLCW sampling rate is necessary for the proper operation of the COP.

F1.1.2 The CLCW consists of ten fields, positioned contiguously, in the following sequence:

- a) Control Word Type (1 bit, mandatory);
- b) CLCW Version Number (2 bits, mandatory);
- c) Status Field (3 bits, mandatory);
- d) COP in Effect (2 bits, mandatory);
- e) Virtual Channel Identification (6 bits, mandatory);
- f) Reserved Spare (2 bits, mandatory);
- g) Flags (5 bits, mandatory);
- h) FARM-B (Expedited) Frame Counter (2 bits, mandatory);
- i) Reserved Spare (1 bit, mandatory);

j) Report Value (8 bits, mandatory).

NOTE – The structural components of the CLCW are shown in figure F-1.

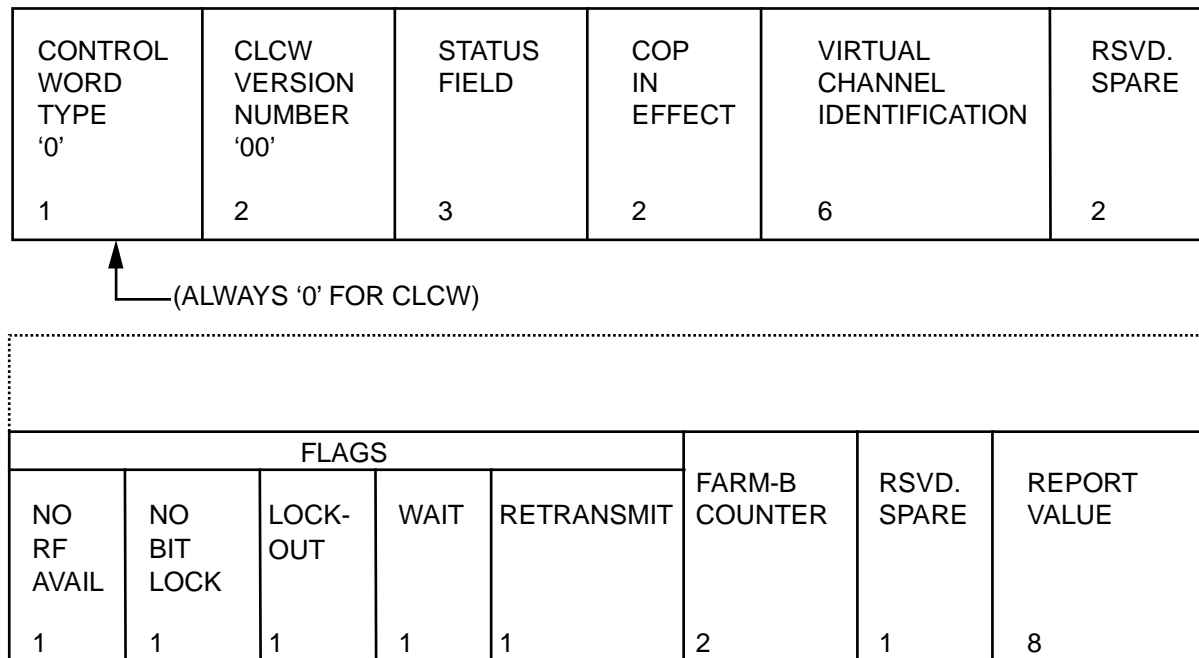


Figure F-1: Communications Link Control Word

F1.2 CONTROL WORD TYPE

F1.2.1 Bit 0 of the CLCW contains the Control Word Type.

NOTE – This field is used to distinguish the CLCW from another type of report that may be alternatively contained in the field that carries the CLCW (e.g., the Operational Control Field of Transfer Frames).

F1.2.2 This one-bit field is set to '0'.

F1.3 CLCW VERSION NUMBER

F1.3.1 Bits 1-2 of the CLCW contain the (Binary Encoded) CLCW Version Number.

F1.3.2 This two-bit field is set to '00'.

NOTE – The CLCW Version Number is included to provide future growth flexibility. At present a single 'Version-1' CLCW, whose binary encoded Version Number is '00', is defined in this Recommended Standard.

F1.4 STATUS FIELD

Bits 3-5 of the CLCW contain the Status Field.

NOTES

- 1 Application of the Status Field is mission-specified.
- 2 The Status Field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

F1.5 COP IN EFFECT

F1.5.1 Bits 6-7 of the CLCW contain the COP in Effect parameter and are used to indicate the COP that is being used.

F1.5.2 For COP-1, this two-bit field is set to '01'.

NOTE – At present a single COP, COP-1, is defined in this Recommended Standard.

F1.6 VIRTUAL CHANNEL IDENTIFICATION

Bits 8-13 of the CLCW contain the Virtual Channel Identifier of the Virtual Channel with which this report is associated.

NOTE – Each Virtual Channel in use has its own CLCW reporting activated.

F1.7 RESERVED SPARE

F1.7.1 Bits 14-15 of the CLCW contain the Reserved Spare.

F1.7.2 These two bits are reserved by CCSDS for future application and are set to '00'.

F1.8 FLAGS

F1.8.1 General

Bits 16-20 of the CLCW contain the Flags specified in the following subsections.

F1.8.2 No RF Available Flag

F1.8.2.1 Bit 16 of the CLCW contains the No RF Available Flag.

F1.8.2.2 The No RF Available Flag provides a logical indication of the ‘ready’ status of the radio frequency (RF) elements within the space link provided by the Physical Layer.

NOTE – Precise definition of the set of physical states which must each be in the ‘ready’ condition before communication is possible is mission-specified. For example, the flag can represent a logical sum of the overall ready status of components such as the RF transponder and the demodulator.

F1.8.2.3 A setting of ‘0’ in the No RF Available Flag indicates that the Physical Layer is Available (i.e., any Transfer Frame will be received and processed by the Physical Layer and passed on to this protocol if correct).

F1.8.2.4 A setting of ‘1’ in the No RF Available Flag indicates that the Physical Layer is not available and that Transfer Frames cannot be transferred without corrective action within the Physical Layer.

F1.8.2.5 The single No RF Available Flag applies to all Virtual Channels and is updated whenever a change is signaled by the Physical Layer.

NOTE – This field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

F1.8.3 No Bit Lock Flag

F1.8.3.1 Bit 17 of the CLCW contains the No Bit Lock Flag.

NOTES

- 1 The No Bit Lock Flag is an optional, mission-specific engineering measurement that provides a performance quality indicator that indicates specifically whether the Physical Layer is working normally by having enough signal energy to achieve bit synchronization with the received data stream.
- 2 Failure to achieve bit lock may indicate that the Physical Layer is operating at a non-nominal performance level and that the Transfer Frame rejection rate may be correspondingly abnormally high.

F1.8.3.2 Use of the No Bit Lock Flag is optional; if used,

- a) ‘0’ indicates bit lock has been achieved;
- b) ‘1’ indicates bit lock has not been achieved.

F1.8.3.3 The single No Bit Lock Flag applies to all Virtual Channels and is updated whenever a change is signaled by the Physical Layer.

F1.8.3.4 If the No Bit Lock Flag is not used, it is set permanently to ‘0’.

NOTE – This field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

F1.8.4 Lockout Flag

F1.8.4.1 Bit 18 of the CLCW contains the Lockout Flag.

F1.8.4.2 The Lockout Flag is used to indicate the Lockout status of the FARM of a particular Virtual Channel.

F1.8.4.3 A setting of ‘1’ in the Lockout Flag indicates Lockout.

NOTE – Lockout occurs whenever a Sequence-Controlled Transfer Frame that violates certain Frame Acceptance Checks is received on a particular Virtual Channel. Once the FARM is in Lockout, all subsequent Sequence-Controlled Transfer Frames will be rejected by the FARM until the condition is cleared.

F1.8.4.4 A setting of ‘0’ in the Lockout Flag indicates that the FARM is not in Lockout.

F1.8.4.5 Separate Lockout Flags are maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Lockout Flag are contained in reference [9].

F1.8.5 Wait Flag

F1.8.5.1 Bit 19 of the CLCW contains the Wait Flag.

F1.8.5.2 The Wait Flag is used to indicate that the receiving end is unable to accept data for processing on a particular Virtual Channel.

NOTE – An inability to accept data could be caused by temporary lack of storage and/or processing resources in the receiving end of this protocol or higher layers.

F1.8.5.3 A setting of ‘1’ (i.e., Wait) in the Wait Flag for a particular Virtual Channel indicates that all further Sequence-Controlled Transfer Frames on that Virtual Channel will be rejected by the FARM until the condition is cleared.

F1.8.5.4 A setting of ‘0’ (i.e., Do Not Wait) in the Wait Flag indicates that the receiving end is able to accept and process incoming Sequence-Controlled Transfer Frames.

F1.8.5.5 Separate Wait Flags are maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Wait Flag are contained in reference [9].

F1.8.6 Retransmit Flag

F1.8.6.1 Bit 20 of the CLCW contains the Retransmit Flag.

NOTE – The Retransmit Flag is used to speed the operation of the COP by providing immediate indication to the FOP at the sending end that retransmission is necessary.

F1.8.6.2 A setting of ‘1’ in the Retransmit Flag indicates that one or more Sequence-Controlled Transfer Frames on a particular Virtual Channel have been rejected or found missing by the FARM and therefore retransmission is required.

F1.8.6.3 A setting of ‘0’ in the Retransmit Flag indicates that there are no outstanding Sequence-Controlled Transfer Frame rejections in the sequence received so far, and thus retransmissions are not required.

F1.8.6.4 Separate Retransmit Flags are maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Retransmit Flag are contained in reference [9].

F1.9 FARM-B (EXPEDITED FRAME) COUNTER

F1.9.1 Bits 21-22 of the CLCW contain the FARM-B (Expedited Frame) Counter.

F1.9.2 Separate FARM-B (Expedited Frame) Counters are maintained for each Virtual Channel.

NOTE – This 2-bit field contains the two least significant bits of a FARM-B Counter. This counter is maintained within the FARM and increments once each time a Expedited Transfer Frame is accepted in Bypass mode on a particular Virtual Channel. The field supports the verification that Type-B (Expedited) Transfer Frames (Control or User Data) were accepted by the receiving end.

F1.10 RESERVED SPARE

F1.10.1 Bit 23 of the CLCW contains the Reserved Spare.

F1.10.2 This bit is reserved by CCSDS for future application and is set to ‘0’.

F1.11 REPORT VALUE

F1.11.1 Bits 24-31 of the CLCW contain the Report Value.

F1.11.2 Separate Report Values are maintained for each Virtual Channel.

NOTE – This 8-bit field contains the value of the Next Expected Frame Sequence Number, $N(R)$, which is equal to the value of FARM's Receiver_Frame_Sequence_Number, $V(R)$. The FARM $V(R)$ counter increments once each time a Sequence-Controlled Transfer Frame containing user data is accepted on a particular Virtual Channel. The precise specifications for use of the Report Value are contained in reference [9].

F2 PROTOCOL DATA UNIT (SPDU)

F2.1 OVERVIEW

The protocol data units discussed below are used by the local transceiver either for local control within the transceiver, or for reporting status to and controlling the remote transceiver.

Supervisory Protocol Data Units (SPDUs) are of either fixed or variable length based upon the value of the SPDU format ID. Currently there is only one fixed-length SPDU defined, i.e., PLCW. Variable-length SPDUs provide the capability for concatenating and multiplexing protocol objects, i.e., directives and status reports. Each SPDU Type is further described in tables F-1 and F-2.

SPDUs can be transmitted using only the Bypass/Sequence Control Flag = '1'.

SPDUs are all self-identifying and self-delimiting. Only variable-length SPDUs further decompose into specific types of supervisory directives or reports. (See annex E for a detailed description of variable-length SPDUs.)

F2.2 GENERAL

F2.2.1 Fixed-length SPDUs consist of the following fields positioned contiguously in the following sequence:

- a) SPDU Header (2 bits) consisting of:
 - 1) SPDU Format ID (1 bit),
 - 2) SPDU Type Identifier (1 bit);
- b) SPDU Data field (14 bits) consisting of either the data field of a fixed-length PLCW or the data field of a CCSDS-reserved SPDU.

F2.2.2 Variable-length SPDUs consist of the following fields positioned contiguously in the following sequence:

- a) SPDU Header (1 octet) consisting of:
 - 1) SPDU Format ID (1 bit),

- 2) SPDU Type Identifier (3 bits),
- 3) Data Field Length (4 bits) (this represents the actual number of octets in the data field of the SPDU);

NOTE – Data Field Length is not a ‘length minus one’ field.

- b) Supervisory Data field (variable length, i.e., 0 to 15 octets) consisting of one or more supervisory directives or status reports of the same SPDU type.

F2.3 FIXED-LENGTH SPDU

F2.3.1 General

A ‘1’ in the SPDU Format ID field identifies a 16-bit fixed-length SPDU. This format provides for only two fixed SPDUs (see table F-1), which are differentiated by the SPDU Type Identifier field:

- a) a ‘0’ in the SPDU Type Identifier field identifies the SPDU as a PLCW;
- b) a ‘1’ in the SPDU Type Identifier field is reserved for future CCSDS specification.

Table F-1: Fixed-Length Supervisory Protocol Data Unit

Fixed-Length SPDU (16 bits)	SPDU Header (2 bits)		SPDU Data (14 bits)
	SPDU Format ID (Bit 0)	SPDU Type Identifier (Bit 1)	(Contains 1 protocol object, i.e., directive or report or PLCW) (Bits 2 through 15)
Type F1	‘1’	‘0’	Fixed-Length PLCW (see F2.3.2)
Type F2	‘1’	‘1’	Reserved for CCSDS Use

F2.3.2 Type F1 SPDU: Proximity Link Control Word

F2.3.2.1 General

F2.3.2.1.1 The PLCW consists of seven fields positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0—see figure F-2):

- a) Report Value (8 bits);

- b) Expedited Frame Counter (3 bits);
- c) Reserved Spares (2 bits);
- d) Retransmit Flag (1 bit);
- e) SPDU Type Identifier (1 bit);
- f) SPDU Format ID (1 bit).

F2.3.2.1.2 The PLCW is transmitted using the Bypass/Sequence Control Flag set to ‘1’ (see 4.1.2.8).

NOTE – The structural components of the PLCW are shown in figure F-2.

Bit 0		Bit 15			
SPDU Header		SPDU Data Field			
SPDU Format ID 1 bit	SPDU Type Identifier 1 bit	Retrans- mit Flag 1 bit	Reserved Spares 2 bits	Expedited Frame Counter 3 bits	Report Value (FSN) 8 bits

Figure F-2: Proximity Link Control Word Fields

F2.3.2.2 Report Value

F2.3.2.2.1 Bits 8–15 of the PLCW contain the Report Value.

F2.3.2.2.2 The Report Value field contains the value of V(R) (see reference [10]).

F2.3.2.2.3 Separate Report Values are reported for each Physical Channel independent of the I/O port.

F2.3.2.3 Expedited Frame Counter

F2.3.2.3.1 Bits 5–7 of the PLCW contains the Expedited_Frame_Counter.

F2.3.2.3.2 The Expedited_Frame_Counter provides a modulo-8 counter indicating that Expedited frames have been received.

F2.3.2.4 Reserved Spares

F2.3.2.4.1 Bits 3-4 of the PLCW contain the Reserved Spare bits.

F2.3.2.4.2 The Reserved Spare field is set to '00'.

F2.3.2.5 PLCW Retransmit Flag

F2.3.2.5.1 Bit 2 of the PLCW contains the PLCW Retransmit Flag.

F2.3.2.5.2 A setting of '0' in the PLCW Retransmit Flag indicates that there are no outstanding frame rejections in the sequence received so far, and thus retransmissions are not required.

F2.3.2.5.3 A setting of '1' in the PLCW Retransmit Flag indicates that a received frame left a frame sequence number gap and that a retransmission of the expected frame is required.

F2.4 VARIABLE-LENGTH SPDU

F2.4.1 General

A '0' in the SPDU Format ID field identifies a variable-length SPDU data field, which may contain from 0 to 15 octets of supervisory data.

NOTE – This form of SPDU uses bits 1 through 3 of the SPDU header to identify one of eight possible SPDU types, summarized in table F-2. Currently three of these eight types are defined in the following two subsections. The remainder are reserved for future CCSDS specification.

Table F-2: Variable-Length Supervisory Protocol Data Unit

Variable-Length SPDU	SPDU Header (1 octet, fixed)			SPDU Data Field (0-15 octets)
	Format ID (Bit 0)	SPDU Type Identifier (Bits 1,2,3)	Length of SPDU Data Field (Bits 4,5,6,7)	(Contains one or more protocol objects, i.e., directives, reports)
Type 1	‘0’	‘000’	Length in Octets	Directives/Reports (see note)
Type 2	‘0’	‘001’	"	TIME DISTRIBUTION PDU
Type 3	‘0’	‘010’	"	Status Reports
Type 4	‘0’	‘011’	"	Reserved for CCSDS Use
Type 5	‘0’	‘100’	"	Reserved for CCSDS Use
Type 6	‘0’	‘101’	"	Reserved for CCSDS Use
Type 7	‘0’	‘110’	"	Reserved for CCSDS Use
Type 8	‘0’	‘111’	"	Reserved for CCSDS Use
NOTE – Directives and Reports can be multiplexed within the SPDU Data Field.				

F2.4.2 Type 1 SPDU: Directives/Reports

An SPDU Type Identifier equal to ‘000’ identifies a Type 1 SPDU with a data field containing from zero to seven 16-bit concatenated and multiplexed protocol objects, i.e., directives or reports.

NOTE – Variable-length SPDU Types are shown in table F-2. Formats of variable-length SPDU data fields are described in annex E.

F2.4.3 Type 2 SPDU: Time Distribution PDU

F2.4.3.1 An SPDU Type Identifier equal to ‘001’ identifies a Type 2 SPDU with a data field containing from 1 to 15 octets of Time Distribution supervisory data.

F2.4.3.2 Octet 0 of the data field contains the time distribution directive type, followed by the actual time field value (1 to 14 octets).

NOTE – Variable-length SPDU Types are shown in table F-2. Formats of variable-length SPDU data fields are described in annex E.

F2.4.4 Type 3 SPDU: Status Reports

An SPDU Type Identifier equal to '010' identifies a Type 3 SPDU with a data field containing from 0 to 15 octets of Status Report information.

NOTES

- 1 The format of these reports is enterprise specific and is left up to the implementation.
- 2 Provision is made in the protocol to identify when a status report is required (NEED_STATUS_REPORT) and when a status report is requested (see Type 1 SPDU Report Request, annex E).

ANNEX G

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

This annex lists the acronyms used in this Recommended Standard.

ARQ	Automatic Repeat Queuing
CCSDS	Consultative Committee for Space Data Systems
CLCW	Communications Link Control Word
CLSR	Common Link Security Report
COP	Communications Operation Procedure
FARM	Frame Acceptance and Reporting Mechanism
FOP	Frame Operation Procedure
GMAP ID	Global Multiplexer Access Point Identifier
GVCID	Global Virtual Channel Identifier
MAP ID	Multiplexer Access Point Identifier
MAP	Multiplexer Access Point
MAPA	Multiplexer Access Point Access
MAPP	Multiplexer Access Point Packet
MC	Master Channel
MCF	Master Channel Frame
MCID	Master Channel Identifier
MRO	(NASA) Mars Reconnaissance Orbiter
MSB	Most Significant Bit
OID	Only Idle Data
OSI	Open Systems Interconnection

PLCW	Proximity Link Control Word
PVN	Packet Version Number
QoS	Quality of Service
SANA	Space Assigned Numbers Authority
SAP	Service Access Point
SCID	Spacecraft Identifier
SDU	Service Data Unit
SDLS	Space Data Link Security
TC	Telecommand
TFVN	Transfer Frame Version Number
UPID	USLP Protocol ID
USLP	Unified Space Data Link Protocol
VC	Virtual Channel
VCF	Virtual Channel Frame
VCID	Virtual Channel Identifier