

Recommendation for Space Data System Standards

|  |
| --- |
| USLP Space Data Link Protocol |

Recommended Standard

CCSDS 732.1-W-0

White Book

Feb 26 2016-Post Telecon #3

AUTHORITY

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Issue: | Recommended Standard, Issue 3 |  |
|  | Date: | September 2015 |  |
|  | Location: | Washington, DC, USA |  |
|  |  |  |  |

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

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 Recommendation 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 Advanced Orbiting Systems (USLP) Space Data Link Protocol described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

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

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *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 (BFSPO)/Belgium.
* Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
* China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
* Chinese Academy of Sciences (CAS)/China.
* 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.

DOCUMENT CONTROL

| **Document** | **Title** | **Date** | **Status** |
| --- | --- | --- | --- |
| CCSDS 732.0-B-1 | USLP Space Data Link Protocol, Issue 1 | September 2003 | Original issue, superseded |
| CCSDS 732.0-B-2 | USLP Space Data Link Protocol, Recommended Standard, Issue 2 | July 2006 | Issue 2, superseded |
| CCSDS 732.0-B-3 | USLP Space Data Link Protocol, Recommended Standard, Issue 3 | September 2015 | Current issue:   * adds specifications to support the Space Data Link Security Protocol; * updates Frame Error Control Field Encoding Procedure to be consistent with other CCSDS Space Data Link Protocol specifications; * changes all occurrences of ‘Packet Service’ and ‘Packet Transfer Service’ to ‘Virtual Channel Packet Service’; * corrects/clarifies Service Specification ‘.indication’ text; * updates/clarifies text relating to Idle Packet generation; * replaces term ‘Idle Frame’ with ‘Only Idle Data (OID) Frame’; |
|  |  |  | Current issue (continued):   * removes obsolete informative annex detailing changes from Historical Recommendation CCSDS 701.0-B-3-S (1989–2005). |

NOTE – Substantive changes from the previous issue are marked by change bars in the inside margin. For terminology changes affecting the entire document, only the first instances are marked.

CONTENTS

Section Page

[1 Introduction 1-1](#_Toc426123963)

[1.1 Purpose 1-1](#_Toc426123964)

[1.2 Scope 1-1](#_Toc426123965)

[1.3 Applicability 1-1](#_Toc426123966)

[1.4 Rationale 1-2](#_Toc426123967)

[1.5 Document Structure 1-2](#_Toc426123968)

[1.6 conventions and Definitions 1-2](#_Toc426123969)

[1.7 References 1-6](#_Toc426123970)

[2 OVERVIEW 2-1](#_Toc426123971)

[2.1 CONCEPT OF USLP Space Data Link Protocol 2-1](#_Toc426123972)

[2.2 OVERVIEW OF SERVICES 2-4](#_Toc426123973)

[2.3 OVERVIEW OF FUNCTIONS 2-10](#_Toc426123974)

[2.4 SERVICES ASSUMED FROM LOWER LAYERS 2-13](#_Toc426123975)

[3 Service DEFINITION 3-1](#_Toc426123976)

[3.1 overview 3-1](#_Toc426123977)

[3.2 SOURCE DATA 3-1](#_Toc426123978)

[3.3 Virtual Channel Packet (VCP) Service 3-3](#_Toc426123979)

[3.4 Octet Stream Service 3-7](#_Toc426123980)

[3.5 Virtual Channel Access (VCA) Service 3-11](#_Toc426123981)

[3.6 Virtual Channel Operational Control Field (VC\_OCF) Service 3-15](#_Toc426123982)

[3.7 Virtual Channel FRAME (VCF) Service 3-18](#_Toc426123983)

[3.8 MASTER Channel FRAME (MCF) Service 3-21](#_Toc426123984)

[3.9 INSERT Service 3-24](#_Toc426123985)

[4 Protocol specification without SDLS Option 4-1](#_Toc426123986)

[4.1 PROTOCOL DATA UNIT (USLP FRAME) 4-1](#_Toc426123987)

[4.2 PROTOCOL PROCEDURES AT THE SENDING END 4-18](#_Toc426123988)

[4.3 PROTOCOL PROCEDURES AT THE RECEIVING END 4-25](#_Toc426123989)

[5 Managed Parameters without SDLS Option 5-1](#_Toc426123990)

[5.1 OVERVIEW OF MANAGED PARAMETERS 5-1](#_Toc426123991)

[5.2 Managed Parameters for a Physical Channel 5-1](#_Toc426123992)

[5.3 Managed Parameters for a MASTER Channel 5-2](#_Toc426123993)

CONTENTS (continued)

Section Page

[5.4 Managed Parameters for a Virtual Channel 5-2](#_Toc426123994)

[5.5 Managed Parameters for a MAP Channel 5-2](#_Toc426123994)

[5.6 Managed Parameters for PACKET TRANSFER 5-3](#_Toc426123995)

[6 Protocol Specification with SDLS OPTION 6-1](#_Toc426123996)

[6.1 Overview 6-1](#_Toc426123997)

[6.2 Use of SDLS PROTOCOL 6-1](#_Toc426123998)

[6.3 USLP TRANSFER FRAME WITH SDLS 6-1](#_Toc426123999)

[6.4 SENDING END PROTOCOL PROCEDURES WITH SDLS 6-4](#_Toc426124000)

[6.5 RECEIVING END PROTOCOL PROCEDURES WITH SDLS 6-7](#_Toc426124001)

[6.6 MANAGED PARAMETERS WITH SDLS 6-9](#_Toc426124002)

Note to CCSDS Editor - Please order the Annexes such that the Normative come first followed by the Informative ones.

**ANNEX A IMPLEMENTATION CONFORMANCE STATEMENT (ICS) PROFORMA (NORMATIVE)………………………………………………………..…A-1**

**ANNEX\_B\_INFORMATIVE\_REFERENCES (INFORMATIVE)………………………………………………………………………...B-1**

**ANNEX C SECURITY, SANA, AND PATENT CONSIDERATIONS (INFORMATIVE)………………………………………………………………………...C-1**

**ANNEX D VARIABLE-LENGTH SUPERVISORY PROTOCOL DATA FIELD FORMATS D (INFORMATIVE)……………………………………………………..…D-1**

**ANNEX E FRAME ERROR CONTROL FIELD CODING PROCEDURES……....E-1**

**ANNEX F PDU CONTROL WORDS/DIRECTIVES (INFORMATIVE).………….F-1**

**ANNEX G ABBREVIATIONS AND ACRONYMS (INFORMATIVE)…………….G-1**

Figure (Tom to fix)

[1-1Bit Numbering Convention 1-5](#_Toc426124005)

[2-1Relationship with OSI Layers 2-1](#_Toc426124006)

[2-2Relationships between Channels 2-3](#_Toc426124007)

[2-3Asynchronous Service Model 2-5](#_Toc426124008)

[2-4Synchronous Service Model 2-6](#_Toc426124009)

[2-5Internal Organization of Protocol Entity (Sending End) 2-11](#_Toc426124010)

[2-6Internal Organization of Protocol Entity (Receiving End) 2-11](#_Toc426124011)

[2-7USLP Space Data Link Protocol Channel Tree 2-12](#_Toc426124012)

[4-1USLP Transfer Frame Structural Components 4-2](#_Toc426124013)

[4-2Transfer Frame Primary Header 4-2](#_Toc426124014)

[4-3Multiplexing Protocol Data Unit (M\_PDU) 4-11](#_Toc426124015)

[4-4Octet Stream Protocol Data Unit (B\_PDU) 4-13](#_Toc426124016)

[4-5Logic Diagram of the Encoder 4-17](#_Toc426124017)

[4-6Logic Diagram of the Decoder 4-18](#_Toc426124018)

[4-7Internal Organization of Protocol Entity (Sending End) 4-19](#_Toc426124019)

[4-8Abstract Model of Packet Processing Function 4-20](#_Toc426124020)

[4-9Abstract Model of Octet Stream Processing Function 4-21](#_Toc426124021)

[4-10Abstract Model of Virtual Channel Generation Function 4-22](#_Toc426124022)

[4-11Abstract Model of Virtual Channel Multiplexing Function 4-23](#_Toc426124023)

[4-12Abstract Model of Master Channel Multiplexing Function 4-24](#_Toc426124024)

[4-13Abstract Model of All Frames Generation Function 4-25](#_Toc426124025)

[4-14Internal Organization of Protocol Entity (Receiving End) 4-26](#_Toc426124026)

[4-15Abstract Model of Packet Extraction Function 4-27](#_Toc426124027)

CONTENTS (continued)

Section Page

[4-16Abstract Model of Octet Stream Reception Function 4-28](#_Toc426124028)

[4-17Abstract Model of Virtual Channel Reception Function 4-29](#_Toc426124029)

[4-18Abstract Model of Virtual Channel Demultiplexing Function 4-30](#_Toc426124030)

[4-19Abstract Model of Master Channel Demultiplexing Function 4-31](#_Toc426124031)

[4-20Abstract Model of All Frames Reception Function 4-32](#_Toc426124032)

[6-1Frame without SDLS Compared to Frame with SDLS 6-2](#_Toc426124033)

Table

[2-1Summary of Services Provided by USLP Space Data Link Protocol 2-7](#_Toc426124034)

[5-1Managed Parameters for a Physical Channel 5-1](#_Toc426124035)

[5-2Managed Parameters for a Master Channel 5-2](#_Toc426124036)

[5-3Managed Parameters for a Virtual Channel 5-2](#_Toc426124037)

[5-4Managed Parameters for Packet Transfer 5-3](#_Toc426124038)

[6-1Additional Managed Parameters for a Virtual Channel when USLP Space   
Data Link Protocol Supports SDLS 6-10](#_Toc426124039)

# Introduction

## 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.

## Scope

This Recommended Standard defines the USLP Space Data Link Protocol in terms of:

1. the services provided to the users of this protocol;
2. the protocol data units employed by the protocol; and
3. the procedures performed by the protocol.

It does not specify:

1. individual implementations or products;
2. the implementation of service interfaces within real systems;
3. the protocol procedures specified in both the COP-1 reference [9] and the COP-P reference [10];
4. the security services specified in the SDLS protocol reference [14];
5. the flow control
6. the methods or technologies required to perform the procedures; or
7. the management activities required to configure and control the protocol.

## 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.

## 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 concepts and rationale for the USLP is documented in the USLP Green Book reference [B11].

## Document Structure

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

1. 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;
2. section 2 provides an overview of the USLP Space Data Link Protocol;
3. section 3 defines the services provided by the protocol entity;
4. section 4 specifies the protocol data units and procedures employed by the protocol entity;
5. section 5 specifies the managed parameters used by the protocol entity;
6. section 6 specifies the protocol entity with support for the Space Data Link Security Protocol;
7. annex A provides the implementation conformance statement proforma (ICS);
8. annex B provides a list of informative references;
9. annex C provides the security, SANA, and patent considerations;
10. annex D lists the variable length Supervisory Protocol Data Unit (SPDU) formats;
11. annex E lists all acronyms used within this document.

## conventions and Definitions

### definitions

#### 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:

1. blocking;
2. connection;
3. Data Link Layer;
4. entity;
5. flow control;
6. Network Layer;
7. peer entities;
8. Physical Layer;
9. protocol control information;
10. protocol data unit;
11. real system;
12. segmenting;
13. service;
14. Service Access Point (SAP);
15. SAP address;
16. service data unit.

#### 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:

1. confirmation;
2. indication;
3. primitive;
4. request;
5. response;
6. service provider;
7. service user.

#### 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 below).

**asynchronous:** not *synchronous* (see below).

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

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

Isochronous: occupying equal time. Examples are simultaneous voice and video transmission.

**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.

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

**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.

### NOMENCLATURE

#### Normative Text

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

1. the words ‘shall’ and ‘must’ imply a binding and verifiable specification;
2. the word ‘should’ implies an optional, but desirable, specification;
3. the word ‘may’ implies an optional specification;
4. 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.

#### 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.

### 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’.

## 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] *CCSDS Proximity-1 Synchronization and Channel Coding.* Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.2-B-2 December 2013.

[8] “Registries.” Space Assigned Number 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.

NOTE – Informative references are listed in annex B.

# OVERVIEW

## CONCEPT OF USLP Space Data Link Protocol

### ARCHITECTURE

The USLP 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 [B2]. The USLP Space Data Link Protocol 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 the USLP Space Data Link Protocol with constraints listed later in this document. How the USLP Space Data Link Protocol is used in overall space data systems is shown in references [B2] through [B4] and [B11].

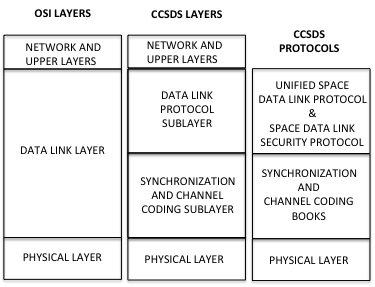


Figure 2‑1 : Relationship with OSI Layers

### Protocol Features

#### Transfer Frames, Virtual Channels, and Multiplexer Access Points

The USLP Space Data Link Protocol 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 the USLP Space Data Link Protocol 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 both packets as well as PDUs of CCSDS recognized protocols. See SANA for the set of recognized protocol IDs [8].

#### 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.

#### Efficient Data Transfer

The USLP Space Data Link Protocol 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 is often 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. The USLP Space Data Link Protocol 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).

USLP 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. This function is provided by an automatic retransmission control mechanism called the Communications Operations Procedure (COP) as described in 2.1.2.4 below.

In addition, a systematic 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).

#### Optional Communications Operation Procedure (COP)

Within this document the term, COP refers to both the Communications Operation Procedure 1 (COP-1) reference [9] as well as the Communications Operation Procedure Proximity (COP-P) within the Proximity-1 Space Data Link Protocol reference [10]. The use of either the COP-1, or COP-P procedures are optional and both are compatible with the USLP. The Protocol Data Units (CLCW for COP-1 and PLCW for COP-P) and operational procedures for both COP-1 and COP-P 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) will be 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 the USLP Space Data Link Protocol. 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 error-protection encoding applied by the Channel Coding 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 in the protocol control report. Expedited Transfer Frames are also used to send Control Commands from the FOP to the FARM.

#### Optional Space Data Link Security Protocol

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 the USLP Space Data Link Protocol.

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.

### 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

MCID = TFVN + SCID;

GVCID = MCID + VCID = TFVN + SCID + 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,

GMAP ID = GVCID + MAP ID

= MCID + VCID + MAP ID

= TFVN + SCID + VCID + MAP ID.

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.

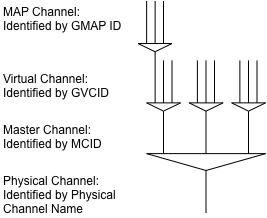


Figure 2‑2 : Relationships between Channels

### Protocol Description

The USLP Space Data Link Protocol is described in terms of:

1. the services provided to the users;
2. the protocol data units; and
3. 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 Channel Coding Sublayer.

## OVERVIEW OF SERVICES

### COMMON FEATURES OF SERVICES

The USLP Space Data Link Protocol 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 an 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:

1. 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;
2. 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 at the time of transfer.
3. unconfirmed services: the sending user does not receive confirmation from the receiving end that data has been received;
4. 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;
5. 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 [B5]).

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

1. 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.
2. 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.

### Service TYPES

#### Overview

The USLP Space Data Link Protocol in association with the COP-1 [9], COP-P [10] and SDLS [14], provide the following service types (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.

Both the sequence-controlled and the expedited QoS are provided by the COP-1 reference [9] and 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 types are provided at any Service Access Point except for the Virtual Channel Frame, 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 perform necessary procedures to provide Sequence-Controlled and/or Expedited Service QoS for its service data units.

#### 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 Request (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:

1. no service data unit is lost;
2. no service data unit is duplicated;
3. no service data unit is delivered out of sequence.

#### Expedited Service – Best Effort Delivery

The Best Effort QoS (Expedited Service) is used when ARQ is not required, 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.

NOTES

1. 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.

#### 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 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.

#### 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

#### 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

#### 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 are three cases in which a synchronous service is periodic:

1. If 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, then the service is periodic.
2. If the service is associated with a Master Channel and there is only one Master Channel in the Physical Channel, then the service is periodic.

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.

### summary OF SERVICES

#### Overview

Nine services are provided by the USLP Space Data Link Protocol. Three of them (MAP Packet, MAP PDU, and MAP Access) are provided for a MAP Channel. Four of them (Octet Stream, Virtual Channel Operational Control Field,Virtual Channel Frame, and On-demand Insert) are provided for a Virtual Channel. One of them (Master Channel Frame) is provided for a Master Channel. One of them (Isochronous 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 PDU  (“For Future Discussion“) | Sequence-Controlled and Expedited, Asynchronous | CCSDS Recognized PDUs – See SANA protocol ID registry | PID | All |
| MAP Access (MAPA) | Sequence-Controlled and Expedited, Asynchronous or Periodic | MAP\_SDU | GMAP ID | All |
| Virtual Channel Octet Stream | Asynchronous or Periodic | Octet Stream Data | GVCID | All |
| Virtual Channel Operational Control Field (VC\_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 |
| Isochronous Insert | Periodic | IN\_SDU | Physical Channel Name | None |
| On-demand Insert (“For Future Discussion”) | Asynchronous | IN\_SDU | GVCID | None |

#### 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.

When using a reliable ARQ protocol with USLP, 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.

#### MAP PDU Service

The MAP PDU Service transfers a sequence of variable-length, delimited, octet-aligned recognized by CCSDS external protocol data units across a space link on a specified MAP Channel. The PDUs transferred by this service must have a Protocol ID (PID) recognized by CCSDS. USLP PIDs 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.

When using a reliable ARQ protocol with USLP, then both Sequence-Controlled and Expedited service types are provided for the MAP PDU Service. The use of a reliable ARQ protocol allows multiple users (i.e., Packets with different PIDs) to share a single GMAP ID. In this case the service provider multiplexes PDUs of different PIDs to form a single stream of PDUs 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 PID and a GMAP ID.

#### 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, only one user, 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 MAP Channel if they contain the same PVN.

#### Virtual Channel Octet Stream Service

The VC 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.

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

#### Virtual Channel Operational Control Field (VC\_OCF) Service

The Virtual Channel Operational Control Field (VC\_OCF) Service provides synchronous 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, 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.

#### 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.

#### 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.

#### Insert Service

USLP provides two complementary types of Insert Service: a) Isochronous and b) On-demand. Only one Insert Service type is allowed on a Physical Channel at any given time.

The Isochronous 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 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.

The On-demand Insert Service provides the optional transfer of private, variable length, octet-aligned service data units across a space link in an episodic manner. The service is unidirectional, asynchronous, and non-sequence preserving. The Insert Zone length may vary between 1 and 255 octets and is specified in the first octet of the Insert Zone. The presence of the on-demand Insert Zone is signaled by the On-demand IZ flag in the TF header.

Multiple users may use this on-demand Insert Service on a Master Channel, identified with the VCID of the frame which contains the On-demand IZ flag set to ‘1’.

### Restrictions on Services

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

1. On one Physical Channel, the Isochronous Insert Service shall not exist simultaneously with the On-demand Insert Service.
2. On one Physical Channel, the Insert service shall not exist simultaneously with the Virtual Channel Frame or Master Channel Frame service.
3. If the Master Channel Frame Service exists on a Master Channel, other services shall not exist simultaneously on that Master Channel.
4. If the Virtual Channel Frame Service exists on a Virtual Channel, other services shall not exist simultaneously on that Virtual Channel.
5. If the Virtual Channel Octet Stream Service exists on a Virtual Channel, other services shall not exist on that Virtual Channel.
6. On one MAP Channel, the MAP Access Service shall not exist simultaneously with the MAP Packet Service.

## OVERVIEW OF FUNCTIONS

### GENERAL FUNCTIONS

The USLP Space Data Link Protocol 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, have either variable lengths and must are 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:

1. generation and processing of protocol control information (i.e., headers and trailers) to perform data identification, loss detection, and error detection;
2. segmenting and blocking of service data units to transfer variable-length service data units in fixed-length protocol data units;
3. 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.

The protocol entity does not perform the following protocol functions:

1. connection establishment and release;
2. flow control;
3. retransmission of protocol data units;
4. management or configuration of the SDLS protocol.

### 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.

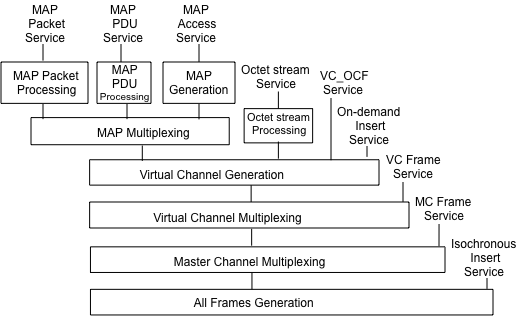


Figure 2‑5 : Internal Organization of Protocol Entity (Sending End)

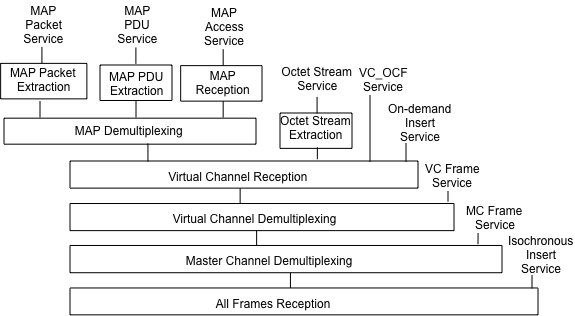


Figure 2‑6 : Internal Organization of Protocol Entity (Receiving End)

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 the USLP Space Data Link Protocol.

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.

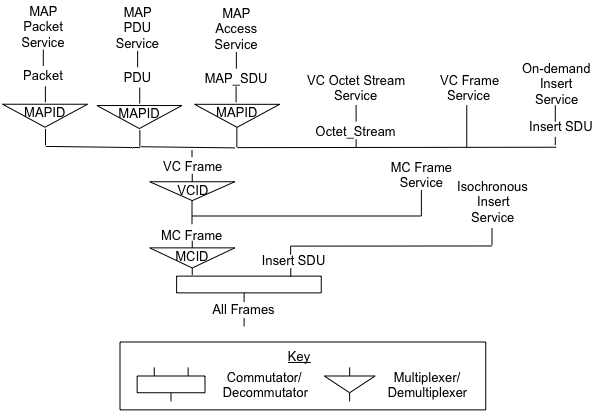


Figure 2‑7 : USLP Space Data Link Protocol Channel Tree

## SERVICES ASSUMED FROM LOWER LAYERS

### 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 the USLP Space Data Link Protocol with the constraints listed in this documents as the Synchronization and Channel Coding Sublayer specification. The functions provided by the Synchronizaton and Channel Coding Recommended Standard are:

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

When any of the coding schemes defined in references [3], [4], [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.

### SYSTEMATIC RETRANSMISSIONS

In addition, the USLP Space Data Link Protocol, 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 error rates are high or link connectivity is marginal.

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.

The USLP Space Data Link Protocol 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, the USLP Space Data Link Protocol does not limit the value of the Repetitions parameter.

### 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:

1. The probability of loss of a transfer frame shall be less than a mission specific value;
2. The probability of not identifying a frame that is in error shall be less than a mission specific value;
3. The probability of misidentifying the MCID and VCID shall be less than a mission-specified value;
4. The probability of not correctly extracting Packets from Transfer Frames using the First Header Pointer and the Packet Length Field 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.

# Service DEFINITION

## 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).

## SOURCE DATA

### SOURCE DATA OVERVIEW

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

The service data units transferred by the USLP Space Data Link Protocol shall be:

1. Packet;
2. CCSDS Recognized PDUs (MAP\_PDU);
3. MAP Access Service Data Unit (MAP\_SDU);
4. Virtual Channel Octet Stream Data;
5. Operational Control Field Service Data Unit (OCF\_SDU);
6. USLP Transfer Frame;
7. Insert Service Data Unit (IN\_SDU).

### Packet

* + 1. Packets shall be transferred over a space link with the MAP Packet Service.
    2. The Packets transferred by this service must be self-delimiting, and have a Packet Version Number (PVN) registered by SANA reference [8].
    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 clause applies also when blocking of Packets is performed 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.
   * 1. Packets shall be either contained within a single frame or within multiple sequential frames of the same GMAP\_ID using the TFDZ Construction Rules in 4.1.4.2.1.4.4.

### MAP PDU

* + 1. MAP\_PDUs shall be transferred over a space link via the MAP PDU Service.
    2. The PDUs transferred by this service must be self-delimiting, and have a Protocol ID (PID) registered by SANA reference [8].
    3. The position and length of the PDU Length Field of the PDU must be known to the service provider in order to extract PDUs from Transfer Frames at the receiving end. This clause applies also when blocking of PDUs is performed by the service provider.

NOTES

1 PDUs are variable-length, delimited, octet-aligned data units.

2 Examples of PDUs are: CFDP PDUs, IPv4 and IPv6 datagrams.

### MAP Access Service Data Unit (MAP\_SDU)

* + 1. MAP\_SDUs shall be transferred over a space link via the MAP Channel Access Service.
    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 are delimited within the TFDZ using the TFDZ construction rules.

### Virtual Channel OCTET STREAM Data

Octet Stream Data shall be transferred over a space link with the Virtual Channel Octet Stream Service. The length of the Octet Stream Data supplied in each Octet Stream service request may not be preserved through transfer, since the Octet Stream Data supplied in a service request may be segmented into multiple Transfer Frames or blocked in a Transfer Frame with the data supplied in other service requests.

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

### Operational Control Field Service Data Unit (OCF\_SDU)

* + 1. Operational Control Field Service Data Units (OCF\_SDUs) shall be transferred over a space link with the VC\_OCF Service. The OCF Flag in the transfer frame primary header signals the presence or absence of this service.
    2. Although the transfer of OCF\_SDUs is carried within the Virtual Channel frame that by management can provide the OCF service, the creation of OCF\_SDUs by the sending user may or may not be synchronized with the Virtual Channel. Such synchronization, if required for timing or other purposes, is a mission-design issue.

NOTES

1. 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.
2. As defined in 4.1.5, CCSDS specifies the use of the first bit of this field to indicate the type of data carried.

### USLP Transfer Frame

NOTE – The USLP Transfer Frame is the variable-length protocol data unit of the USLP Space Data Link Protocol, but also can be used as the service data units of the Virtual Channel Frame and Master Channel Frame Services. Its format is defined in 4.1 and 6.3 of this Recommended Standard.

### Insert Service Data Unit (IN\_SDU)

Isochronous Insert Service Data Units (IN\_SDUs) are periodic, octet-aligned data units of fixed length and their presence within the frame is controlled by Managed Parameters. Their length may be of any constant value which is an integral number of octets, between 1 octet and the maximum length of the data-carrying space of the Transfer Frames, and is established by management. The length of IN\_SDUs at the sending interface is always equal to the length at the receiving interface.

On-demand Insert Service Data Units (IN\_SDUs) are asynchronous, octet-aligned data units of variable length and their presence within the frame is signaled by the On-demand Insert Zone Flag. Their length may be of any value which is an integral number of octets, between 1 octet and 255 octets and is specified in the first octet of the Insert Zone. The length of IN\_SDUs at the sending interface is always equal to the length at the receiving interface.

## MAP Packet Service

### 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.

### MAP Packet SERVICE PARAMETERS

#### General

The parameters used by the MAPP Service primitives shall conform to the specifications contained in subsections 3.3.2.2 through 3.3.2.6.

#### 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.

#### 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.

#### 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.

* + 1. SDU ID
    2. The SDU IDparameter shall contain a user-supplied sequence number to be used to identify the associated Packet in subsequent MAPP\_Notify.indication primitives.

#### Service Type

* + - * 1. The Service Type parameter shall indicate whether the Packet should be transferred with the Sequence-Controlled Service type or the Expedited Service type.
        2. At the receiving end, the Service Type parameter is not used.
    1. **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].

#### 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.

#### Verification Status Code

* + - * 1. The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.
        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. 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.

## MAP PDU Service

### OVERVIEW OF MAP PDU SERVICE

The MAP PDU Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as PDUs across a space link. The PDUs transferred by this service must have a Protocol ID (PID) recognized by CCSDS. USLP Protocol IDs 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 PDUs with a single PID and identified with the PID and a GMAP ID. Different users (i.e., PDUs with different PIDs) may share a single MAP Channel, and if there are multiple users on a MAP Channel, the service provider multiplexes PDUs of different PIDs to form a single stream of PDUs to be transferred on that MAP Channel.

### MAP Packet SERVICE PARAMETERS

#### General

The parameters used by the MAPP Service primitives shall conform to the specifications contained in subsections 3.4.2.2 through 3.4.2.10.

#### PDU

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

NOTE – The PDU parameter is the service data unit transferred by the MAP PDU Service. Restrictions on the PDUs transferred by the MAP PDU Service are stated in 3.4.2.

#### GMAP ID

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

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

#### USLP Protocol ID

The USLP Protocol ID (PID) shall identify the CCSDS protocol entity of the upper layer that uses the MAP PDU Service.

NOTE – The PID is part of the SAP address of the MAP PDU Service.

* + 1. SDU ID
    2. The SDU IDparameter shall contain a user-supplied sequence number to be used to identify the associated PDU in subsequent MAPPDU\_Notify.indication primitives.

#### Service Type

* + - * 1. The Service Type parameter shall indicate whether the PDU should be transferred with the Sequence-Controlled Service type or the Expedited Service type.
        2. At the receiving end, the Service Type parameter is not used.
    1. **Notification Type**

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

#### PDU Quality Indicator

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

#### Verification Status Code

* + - * 1. The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.
        2. The Verification Status Code parameter shall be used to notify the user at the receiving end of the MAP PDU Service of a verification failure in a transfer frame addressed to the MAP Channel.
        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 PDU. Processing of frames failing verification is implementation specific and depends also on the processing capabilities of the service user for eventual forensic investigation.

### MAP PDU SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. MAP\_PDU.request;
2. MAP\_PDU.indication.

#### MAP\_PDU.request

##### Function

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

NOTE – The MAP\_PDU.request primitive is the service request primitive for the MAP PDU Service.

##### Semantics

The MAP\_PDU.request primitive shall provide parameters as follows:

MAP\_PDU.request (MAP PDU,  
GMAP\_ID,  
USLP Protocol ID)

##### When Generated

The MAP\_PDU.request primitive shall be passed to the service provider to request it to send the MAP PDU.

##### Effect On Receipt

Receipt of the MAP\_PDU.request primitive shall cause the service provider to transfer the MAP PDU.

##### Additional Comments

The MAP\_PDU.request primitive shall be used to transfer MAP PDUs across the space link on the specified MAP Channel.

#### MAP\_PDU.indication

##### Function

At the receiving end, the service provider shall pass a MAP\_PDU.indication to the MAP PDU Service user to deliver a MAP PDU.

NOTE – The MAP\_PDU.indication primitive is the service indication primitive for the MAP PDU Service.

##### Semantics

The MAP\_PDU.indication primitive shall provide parameters as follows:

MAP\_PDU.indication (MAP PDU,  
GMAP\_ID,  
USLP Protocol ID,

MAP\_PDU Quality Indicator (optional),  
Verification Status Code (optional))

##### When Generated

The MAP\_PDU.indication primitive shall be passed from the service provider to the MAP PDU Serviceuser at the receiving end to deliver a MAP\_PDU.

##### Effect On Receipt

The effect of receipt of the MAP\_PDU.indication primitive by the MAP\_PDU Serviceuser is undefined.

##### Additional Comments

The MAP\_PDU.indication primitive shall be used to deliver MAP PDUsto the MAP PDU Serviceuser identi­fied by the GMAP\_ID, USLP Protocol ID. Incomplete MAP PDUs may be delivered (optional).

## Virtual Channel Octet Stream Service

### OVERVIEW OF Octet Stream SERVICE

The 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.

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

### Octet Stream SERVICE PARAMETERS

#### General

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

#### 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.

#### GVCID

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

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

#### 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.

#### Verification Status Code

* + - * 1. The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.
        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. 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.

### Octet Stream SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. OCTET STREAM.request;
2. OCTET STREAM.indication.

#### OCTET STREAM.request

##### 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.

##### Semantics

The OCTET STREAM.request primitive shall provide parameters as follows:

OCTET STREAM.request (Octet Stream Data,  
GVCID)

##### When Generated

The OCTET STREAM.request primitive shall be passed to the service provider to request it to send the Octet Stream Data.

##### Effect On Receipt

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

##### Additional Comments

The OCTET STREAM.request primitive shall be used to transfer Octet Stream Data across the space link on the specified Virtual Channel.

NOTE – 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.

#### OCTET STREAM.indication

##### Function

At the receiving end, the service provider shall pass a 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.

##### Semantics

The OCTET STREAM.indication primitive shall provide parameters as follows:

OCTET STREAM.indication (Octet Stream Data,  
GVCID,  
Octet Stream Data Loss Flag (optional),  
Verification Status Code (optional))

##### When Generated

The OCTET STREAM.indication primitive shall be passed from the service provider to the Octet Stream Serviceuser at the receiving end to deliver Octet Stream Data.

##### Effect On Receipt

The effect of receipt of the OCTET STREAM.indication primitive by the Octet Stream Serviceuser is undefined.

##### Additional Comments

The OCTET STREAM.indication primitive shall be used to deliver Octet Stream Data to the Octet Stream Serviceuser identified by the GVCID.

NOTE – 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.

## MAP Access Service

### 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 CCSDS PID and a GMAP ID. Different users (i.e., MAP\_SDU identified with a different PID) 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 4.1.4.2.1.4.4.) on the same GMAP ID.

### MCA SERVICE PARAMETERS

#### General

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

#### 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 3.2.4.

#### 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.

#### MAP\_SDU Loss Flag

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

#### Verification Status Code

* + - * 1. The Verification Status Code is an optional parameter that may be used if the service provider supports the optional SDLS protocol.
        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. 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.

### MAPA SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. MAPA.request;
2. MAPA\_Notify.indication;
3. MAPA.indication.

#### MAPA.request

##### 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.

##### 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.

##### When Generated

The sending-end service user shall generate a MAPA.request primitive when a MAP\_SDU is ready to be transferred.

##### Effect On Receipt

Receipt of the MAPA.request primitive shall cause the service provider to transfer the MAP\_SDU.

#### MAPA\_Notify.indication

##### 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.

##### 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 Exedited Services, the Service Type parameter is not used in this primitive.

##### When Generated

The sending-end service provider shall generate a MAPA\_Notify.indication primitive in response to an event associated with the transfer of a MAP\_SDU.

##### Effect On Receipt

The effect of receipt of the MAPA\_Notify.indication primitive by the MAPA Serviceuser is undefined.

#### MAPA.indication

##### 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.

##### Semantics

The MAPA.indication primitive shall provide parameters as follows:

MAPA.indication (MAP\_SDU,  
GVCID,  
MAP ID,  
Service Type (optional),  
Verification Status Code (optional))

##### When Generated

The receiving-end service provider shall generate a MAPA.indication primitive when a MAP\_SDU is ready to be delivered.

##### Effect On Receipt

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

## Virtual Channel Operational Control Field (VC\_OCF) Service

### OVERVIEW OF VC\_OCF SERVICE

The Virtual Channel Operational Control Field (VC\_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 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 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.

### VC\_OCF SERVICE PARAMETERS

#### General

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

#### OCF\_SDU

The parameter OCF\_SDU shall be the service data unit transferred by the VC\_OCF Service in the Operational Control Field of Transfer Frames of a Virtual Channel.

NOTE – Restrictions on the OCF\_SDU transferred by the VC\_OCF Service are stated in 3.2.5.

#### 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 VC\_OCF Service.

#### OCF\_SDU Loss Flag

The OCF\_SDU Loss Flag is an optional parameter that may be used to notify the user at the receiving end of the VC\_OCF Service that a sequence discontinuity has been detected and that one or more OCF\_SDUs may have been lost. If implemented, the flag shall be derived by examining the Virtual Channel Frame Count in the Transfer Frames.

### VC\_OCF SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. VC\_OCF.request;
2. VC\_OCF.indication.

#### VC\_OCF.request

##### Function

At the sending end, the VC\_OCF Service user shall pass a VC\_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 VC\_OCF.request primitive is the service request primitive for the VC\_OCF Service.

##### Semantics

The VC\_OCF.request primitive shall provide parameters as follows:

VC\_OCF.request (OCF\_SDU,   
GVCID)

##### When Generated

The VC\_OCF.request primitive shall be passed to the service provider to request it to send the OCF\_SDU.

##### Effect On Receipt

Receipt of the VC\_OCF.request primitive shall cause the service provider to transfer the OCF\_SDU.

##### Additional Comments

The VC\_OCF.request primitive shall be used to transfer OCF\_SDUs across the space link on the specified Virtual Channel.

#### VC\_OCF.indication

##### Function

At the receiving end, the service provider shall pass a VC\_OCF.indication to the VC\_OCF Service user to deliver an OCF\_SDU.

NOTE – The VC\_OCF.indication primitive is the service indication primitive for the VC\_OCF Service.

##### Semantics

The VC\_OCF.indication primitive shall provide parameters as follows:

VC\_OCF.indication (OCF\_SDU,  
GVCID,  
OCF\_SDU Loss Flag (optional))

##### When Generated

The VC\_OCF.indication primitive shall be passed from the service provider to the VC\_OCF Serviceuser at the receiving end to deliver an OCF\_SDU.

##### Effect On Receipt

The effect of receipt of the VC\_OCF.indication primitive by the VC\_OCF Serviceuser is undefined.

##### Additional Comments

The VC\_OCF.indication primitive shall be used to deliver OCF\_SDUsto the VC\_OCF Serviceuser identified by the GVCID.

## Virtual Channel FRAME (VCF) Service

### 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.

### VCF SERVICE PARAMETERS

#### 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.

#### Frame

The Frame parameter shall be an 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.

#### 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.

#### Frame Loss Flag

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

### VCF SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. VCF.request;
2. VCF.indication.

#### VCF.request

##### 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.

##### Semantics

The VCF.request primitive shall provide parameters as follows:

VCF.request (Frame,  
GVCID)

##### When Generated

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

##### Effect On Receipt

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

##### Additional Comments

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

#### VCF.indication

##### 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.

##### Semantics

The VCF.indication primitive shall provide parameters as follows:

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

##### When Generated

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

##### Effect On Receipt

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

##### Additional Comments

The VCF.indication primitive is used to deliver Transfer Framesof a Virtual Channelto the VCF Serviceuser identified by the GVCID.

## MASTER Channel FRAME (MCF) Service

### 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.

### MCF SERVICE PARAMETERS

#### 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.

#### Frame

The Frame parameter shall be an 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.

#### 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.

#### Frame Loss Flag

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

### MCF SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. MCF.request;
2. MCF.indication.

#### MCF.request

##### 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.

##### Semantics

The MCF.request primitive shall provide parameters as follows:

MCF.request (Frame,  
MCID)

##### When Generated

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

##### Effect On Receipt

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

##### Additional Comments

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

#### MCF.indication

##### 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.

##### Semantics

The MCF.indication primitive shall provide parameters as follows:

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

##### When Generated

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

##### Effect On Receipt

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

##### Additional Comments

The MCF.indication primitive shall be used to deliver Transfer Framesof a Master Channelto the VCF Serviceuser identified by the MCID.

## ISOCHRONOUS INSERT Service

### OVERVIEW OF ISOCHRONOUS 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 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 Isochronous 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.

### INSERT SERVICE PARAMETERS

#### General

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

#### 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.

#### 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.

#### IN\_SDU Loss Flag

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

### INSERT SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. ISOCHRONOUS\_INSERT.request;
2. ISOCHRONOUS\_INSERT.indication.

#### ISOCHRONOUS\_INSERT.request

##### Function

At the sending end, the Insert Service user shall pass an ISOCHRONOUS\_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 ISOCHRONOUS\_INSERT.request primitive is the service request primitive for the Insert Service.

##### Semantics

The ISOCHRONOUS\_INSERT.request primitive shall provide parameters as follows:

ISOCHRONOUS\_INSERT.request (IN\_SDU,  
Physical Channel Name)

##### When Generated

The ISOCHRONOUS\_INSERT.request primitive is passed to the service provider to request it to send the IN\_SDU.

##### Effect On Receipt

Receipt of the ISOCHRONOUS\_INSERT.request primitive causes the service provider to transfer the IN\_SDU.

##### Additional Comments

The ISOCHRONOUS\_INSERT.request primitive is used to transfer IN\_SDUs across the space link on the specified Physical Channel.

#### ISOCHRONOUS\_INSERT.indication

##### Function

At the receiving end, the service provider shall pass an ISOCHRONOUS\_INSERT.indication to the Isochronous Insert Service user to deliver an IN\_SDU.

NOTE – The ISOCHRONOUS\_INSERT.indication primitive is the service indication primitive for the Insert Service.

##### Semantics

The ISOCHRONOUS\_INSERT.indication primitive shall provide parameters as follows:

ISOCHRONOUS\_INSERT.indication (IN\_SDU,  
Physical Channel Name,  
IN\_SDU Loss Flag (optional))

##### When Generated

The ISOCHRONOUS\_INSERT.indication primitive shall be passed from the service provider to the Isochronous Insert Serviceuser at the receiving end to deliver an IN\_SDU.

##### Effect On Receipt

The effect of receipt of the ISOCHRONOUS\_INSERT.indication primitive by the Isochronous Insert Serviceuser is undefined.

##### Additional Comments

The ISOCHRONOUS\_INSERT.indication primitive shall be used to deliver IN\_SDUsto the Isochronous Insert Serviceuser identified by the Physical Channel Name.

#### Directive.request

##### Function

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

##### Semantics

The Directive.request primitive shall provide parameters as follows:

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

##### When Generated

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

##### Effect On Receipt

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

#### Directive\_Notify.indication

##### 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.

##### Semantics

The Directive\_Notify.indication primitive shall provide parameters as follows:

Directive\_Notify.indication (GVCID,  
Directive ID,  
Notification Type)

##### 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.

#### Async\_Notify.indication

##### 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.

##### Semantics

The Async\_Notify.indication primitive shall provide parameters as follows:

Async\_Notify.indication (GVCID,   
Notification Type,  
Notification Qualifier)

##### When Generated

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

## On-DEMAND INSERT Service

### OVERVIEW OF On-Demand INSERT SERVICE

The On-demand Insert Service provides the optional transfer of private, variable length, octet-aligned service data units across a space link in an episodic manner. The service is unidirectional, asynchronous, and non-sequence preserving. The Insert Zone length may vary between 1 and 255 octets and is specified in the first octet of the Insert Zone. The presence of the on-demand Insert Zone is signaled by the On-demand IZ flag in the TF header.

Multiple users may use this on-demand Insert Service on a Master Channel, identified with the VCID of the frame which contains the On-demand IZ flag set to ‘1’.

### On-demand INSERT SERVICE PARAMETERS

#### General

The parameters used by the On-demand Insert Service primitives shall conform to the specifications contained in subsections 3.10.2.2 through 3.10.2.4.

#### IN\_SDU

The parameter IN\_SDU shall be the service data unit transferred by the On-demand Insert Service.

NOTE – Restrictions on the IN\_SDUs transferred by the On-demand Insert Service are stated in 3.2.7.

#### Global Virtual Channel ID (GVCID)

The GVCID shall indicate the Global Virtual Channel through which the IN\_SDU is to be transferred.

NOTE – The GVCID is the SAP address of the On-demand Insert Service.

#### ~~IN\_SDU Loss Flag~~

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

### On-Demand INSERT SERVICE PRIMITIVES

#### General

The service primitives associated with this service are:

1. ON-DEMAND\_INSERT.request;
2. ON-DEMAND\_INSERT.indication.

#### ON-DEMAND\_INSERT.request

##### Function

At the sending end, the On-demand Insert Service user shall pass an ON-DEMAND\_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 ON-DEMAND\_INSERT.request primitive is the service request primitive for the On-demand Insert Service.

##### Semantics

The ON-DEMAND\_INSERT.request primitive shall provide parameters as follows:

ON-DEMAND\_INSERT.request (IN\_SDU,  
GVCID)

##### When Generated

The ON-DEMAND\_INSERT.request primitive is passed to the service provider to request it to send the IN\_SDU.

##### Effect On Receipt

Receipt of the ON-DEMAND\_INSERT.request primitive causes the service provider to transfer the IN\_SDU.

##### Additional Comments

The ON-DEMAND\_INSERT.request primitive is used to transfer IN\_SDUs across the space link on the specified Virtual Channel.

#### ON-DEMAND\_INSERT.indication

##### Function

At the receiving end, the service provider shall pass an ON-DEMAND\_INSERT.indication to the On-demand Insert Service user to deliver an IN\_SDU.

NOTE – The ON-DEMAND\_INSERT.indication primitive is the service indication primitive for the On-Demand Insert Service.

##### Semantics

The ON-DEMAND\_INSERT.indication primitive shall provide parameters as follows:

ON-DEMAND\_INSERT.indication (IN\_SDU,  
VCID))  
~~IN\_SDU Loss Flag (optional))~~

##### When Generated

The ON-DEMAND\_INSERT.indication primitive shall be passed from the service provider to the On-demand Insert Serviceuser at the receiving end to deliver an IN\_SDU.

##### Effect On Receipt

The effect of receipt of the ON-DEMAND\_INSERT.indication primitive by the On-demand Insert Serviceuser is undefined.

##### Additional Comments

The ON-DEMAND\_INSERT.indication primitive shall be used to deliver IN\_SDUsto the On-demand Insert Serviceuser identified by the GVCID.

#### Directive.request

##### Function

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

##### Semantics

The Directive.request primitive shall provide parameters as follows:

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

##### When Generated

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

##### Effect On Receipt

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

#### Directive\_Notify.indication

##### 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.

##### Semantics

The Directive\_Notify.indication primitive shall provide parameters as follows:

Directive\_Notify.indication (GVCID,  
Directive ID,  
Notification Type)

##### 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.

#### Async\_Notify.indication

##### 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.

##### Semantics

The Async\_Notify.indication primitive shall provide parameters as follows:

Async\_Notify.indication (GVCID,   
Notification Type,  
Notification Qualifier)

##### When Generated

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

# Protocol specification without SDLS Option

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

## PROTOCOL DATA UNIT

### USLP Transfer Frame

* + 1. A USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

1. Transfer Frame Primary Header (4 to 14 octets; mandatory);
2. Transfer Frame Insert Zone (integral number of octets; optional);
3. Transfer Frame Data Field (integral number of octets; mandatory);
4. Operational Control Field (4 octets; optional);
5. Frame Error Control Field (2 octets, optional).

NOTE – 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 [B11] for the motivation of this exceptional case.

* + 1. The USLP Transfer Frame length shall be consistent with the specifications contained in references [3], [4], [5], [6], and [7]. The structural components of the USLP Transfer Frame are shown in figure 4‑1.

NOTES

1. The protocol data unit of the USLP Space Data Link Protocol 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 start of the Transfer Frame is signaled by the underlying Channel Coding Sublayer.
3. The Frame Error Control Field is controlled by Managed Parameters, see Section 5.

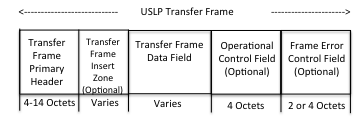


Figure 4‑1 : USLP Transfer Frame Structural Components

### TRANSFER FRAME PRIMARY HEADER

#### General

The Transfer Frame Primary Header is mandatory and shall consist of up to 13 fields, positioned contiguously, in the following sequence:

1. Transfer Frame Version Number (4 bits, mandatory);
2. Spacecraft Identifier (16 bits, mandatory);
3. Source or Destination Identifier (1 bit, mandatory);
4. Virtual Channel Identifier (6 bits, mandatory);
5. Multiplexer Access Point (MAP) Identifier (4 bits, mandatory);
6. End of Transfer Frame Primary Header Flag (1 bit, mandatory);
7. Frame Length (16 bits, mandatory);
8. Reserve Spares (2 bits, mandatory);
9. Bypass Flag (1 bit, mandatory);
10. On-demand Insert Zone Flag (1 bit, mandatory);
11. Operational Control Field Flag (1 bit, mandatory);
12. Virtual Channel Frame Count Length (3 bits, mandatory);
13. Virtual Channel Frame Count (0 to 7 octets; optional).

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

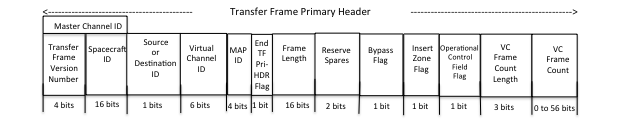


Figure 4‑2 : Transfer Frame Primary Header

#### Master Channel Identifier

##### General

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

The Master Channel Identifier shall consist of:

1. Transfer Frame Version Number (4 bits, mandatory);
2. Spacecraft Identifier (16 bits, mandatory).

##### Transfer Frame Version Number

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

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.

##### Spacecraft Identifier

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

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.

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].

#### Source-or-Destination Identifier

* + - * 1. Bit 20 of the Transfer Frame Primary Header shall contain the Source or Destination ID.
        2. The Source-or-Destination Identifier shall be used to identify the association of the data contained in the Transfer Frame.

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

1. ‘0’ = SCID refers to the *source* of the Transfer Frame;
2. ‘1’ = SCID refers to the *destination* of the Transfer Frame.

#### Virtual Channel Identifier

* + - * 1. Bits 21–26 of the Transfer Frame Primary Header shall contain the Virtual Channel Identifier (VCID).
        2. The Virtual Channel Identifier shall be used to identify the Virtual Channel.

NOTE – VCID 63 shall be used for transmission of Only Idle Data (OID) Transfer Frames. See 4.1.4.1.9.

NOTES

1. 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.
2. 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 (if signaled or managed) and/or an OCF if signaled within the Frame Primary Header.

#### Multiplexer Access Point Identifier

* + - * 1. Bits 27–30 of the Transfer Frame Primary Header shall contain the Multiplexer Access Point Identifier (MAP ID).
        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 one 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.

#### End of Transfer Frame Primary Header Flag

* + - * 1. Bit 31 of the Transfer Frame Primary Header shall contain the End of Transfer Frame Primary Header Flag.
        2. The End of Transfer Frame Primary Header Flag shall signal the use of a truncated TF 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 TF 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.

#### Frame Length

* + - * 1. Bits 32–47 of the Transfer Frame Primary Header shall contain the Frame Length.
        2. This 16-bit field shall contain a length count C which equals one fewer than the total octets in the Transfer Frame.
        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* = (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.

#### Reserve Spares

* + - * 1. Bits 48-49 of the Transfer Frame Primary Header shall contain the reserved spares.
        2. This 2-bit field is reserved for future definition by CCSDS and shall be set to ‘00’.

#### Bypass Flag

* + - * 1. Bit 50 of the Transfer Frame Primary Header shall contain the Bypass Flag.

The single-bit Bypass 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:

1. setting the Bypass 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;
2. setting the Bypass Flag to value ‘1’ shall specify that this Transfer Frame is a 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 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 Flag. See 4.4.8.5.
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.

#### On-demand Insert Zone Flag

* + - * 1. Bit 51 of the Transfer Frame Primary Header shall contain the On-demand Insert Zone Flag.
        2. The On-demand Insert Zone Flag shall indicate the presence or absence of the On-demand Insert Zone. It shall be ‘1’ if the On-demand Insert Zone is present; it shall be ‘0’ if the On-demand Insert Zone is not present.

NOTE – The presence or absence of the Isochronous Insert Zone is controlled by Managed Parameters and is not affected by this flag.

#### Operational Control Field Flag

* + - * 1. Bit 52 of the Transfer Frame Primary Header shall contain the Operational Control Field Flag.
        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.

#### Virtual Channel Frame Count Length

* + - * 1. Bits 53-55 of the Transfer Frame Primary Header shall contain the length of the Virtual Channel Frame Count field.
        2. This 3-bit field shall define the length of the Virtual Channel Frame Count field.
        3. The value of this field shall be interpreted as shown in Table 4-2.

Table 4‑2 : 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. The COP does not use this field of Expedited Transfer Frames; in this case the contents of the Frame Count field (i.e., Sequence Number) is set to ‘all zeroes’ by COP.

#### NOTE – A VC Frame Count Length of ‘000’ indicates that the VC Frame Count field is absent from the Transfer Frame Header.

#### Virtual Channel Frame Count

* + - * 1. If present, the Virtual Channel Frame Count shall be the final field in the Transfer Frame Header.
        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.
        3. The VC Frame Count shall increment monotonically and independently for the set of Sequence-Controlled Frames i.e., the Bypass Flag are set to ‘0’. In this case, the Frame Count is called the Sequence-Controlled Counter.
        4. The VC Frame Count shall increment monotonically and independently for the set of Expedited frames i.e., the Bypass Flag are set to ‘1’. In this case, the Frame Count is called the FARM-B (Expedited Frame) Counter.
        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.

1. 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].

### Transfer Frame INSERT ZONE

* + 1. If included, the Transfer Frame Insert Zone shall follow, without gap, the Transfer Frame Primary Header.
    2. If the Physical Channel supports the Isochronous Insert Service for transfer of periodic data, then the Insert Zone shall exist in every Transfer Frame transmitted within the same Physical Channel, including OID Transfer Frames.
       - 1. For Isochronous 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.
         2. Once set by management, the length of the Insert Zone for Isochronous Service shall be static throughout a Mission Phase.

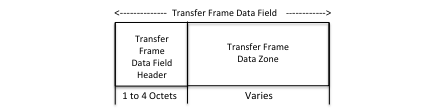
NOTE – The presence or absence of the Isochronous Insert Zone is controlled by Managed Parameters.

* + 1. If a given Virtual Channel supports the On-demand Insert Service for transfer of asynchronous data, then the presence of the Insert Zone shall be signaled by the On-demand Insert Zone flag.
       - 1. For On-demand Insert Service, the length of the Insert Zone per given Virtual Channel may vary by frame and shall be contained in the first octet of the Insert Service Data Unit (IN\_SDU). The Insert Zone length may vary between 1 and 255 integer octets. The Insert Zone may contain one or more self-delimiting octet-aligned IN\_SDUs.

### TRANSFER FRAME DATA FIELD

#### General

* + - * 1. The Transfer Frame Data Field (TFDF) shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone, if present.
        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).
        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**

* + - * 1. The Transfer Frame Data Field, which shall contain an integral number of octets, may vary in length up to a maximum of 65528 octets.
        2. 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 Control Command information.
        3. MAP\_SDUs shall not be mixed with Packets on the same Virtual Channel. Idle Data shall be transferred on a Virtual Channel that transfers Packets. Whether a particular Virtual Channel transfers Packets (and possibly Idle Data) or MAP\_SDUs shall be established by management and static throughout a Mission Phase.
        4. If Packets are contained in the Transfer Frame Data Field, Packets shall be inserted contiguously and in forward order into the Trans­fer Frame Data Field.

NOTE – The first and last Packets of the Transfer Frame Data Field are not necessarily complete, since the first Packet may be a continuation of a Packet begun in the previous Trans­fer Frame, and the last Packet may continue in the subsequent Trans­fer Frame of the same Virtual Channel.

* + - * 1. 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. Such a Transfer Frame is called an OID (Only Idle Data in its Data Field) Transfer Frame. 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.

NOTES

1. Transfer Frames containing 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.
2. Idle Data in the Transfer Frame Data Field of an OID Transfer Frame must not be confused with the Idle Packet specified in reference [8].
3. 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. OID Transfer Frames may be sent on Virtual Channels that also carry valid Packets, but it is preferred that a separate Virtual Channel be dedicated to carry OID Transfer Frames unless there is a need to send OID Transfer Frames on a specific Virtual Channel (e.g., to transmit data in the Operational Control Field on a specific Virtual Channel).
5. An OID Transfer Frame can be generated whenever it is necessary (even in the middle of transmission of a Packet that is split into multiple Transfer Frames).
6. 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.

#### TRANSFER FRAME DATA FIELD HEADER

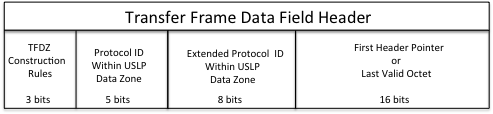
##### General

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

The TFDF Header shall contain the following fields:

1. Transfer Frame Data Zone (TFDZ) Construction Rules (3 bits, mandatory);
2. Protocol Identifier Within USLP Data Zone (5 bits, mandatory);
3. Protocol Identifier Extension Within USLP Data Zone (8 bits, optional);
4. 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.



**Figure 4-4: Transfer Frame Data Field Header**

###### TFDZ Construction Rules

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

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

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

The TFDZ Construction Rules shall be interpreted as shown in table 4‑3.

Table 4‑3 : Summary of the TFDZ Construction Rules

|  |  |
| --- | --- |
| TFDZ Construction Rules | Interpretation |
| 000 | Spanning Data Units (A fixed length TFDZ that may contain one more Packets or PDUs of CCSDS recognized protocols that span one or more frames. TFDF Header contains the First Header Pointer.) |
| 001 | A partially filled fixed length TFDZ containing the last segment of a MAPA\_SDU (TFDF Header contains the Last Valid Octet Pointer). |
| 010 | Octet Stream (User defined octet aligned) Data. |
| 011 | * + - * 1. Start of a MAPA\_SDU contained within a single fixed length TFDZ. |
| 100 | Starting Segment (i.e., SDU is segmented and starts in this TFDZ but does not end in this TFDZ). |
| 101 | Continuation Segment (i.e., continuing portion of a SDU contained in this TFDZ). |
| 110 | Ending Segment (i.e., SDU is segmented and ends but does not start in this TFDZ). |
| 111 | No Segmentation (one complete SDU or multiple Packets contained in this TFDZ). |

TFDZ Construction Rule “000” defines a fixed length TFDZ and the contents are CCSDS Packets or PDUs of CCSDS recognized protocols that span transfer frame boundaries. The First Header Point is required and its function is to enable the packet or PDU extraction process to restart whenever a transfer frame is lost on that VC.

TFDZ Construction Rule “001” defines a fixed length TFDZ and contains the final segment of a MAPA\_SDU. The Last Valid Octet Pointer is required.

NOTE - MAP\_SDU must begin in the first octet of a TFDZ and segments of that MAP\_SDU must be contained in the following transfer frames of that VC with the last frame containing the MAPA\_SDU having the TFDZ Construction Rule “001”.

The TFDZ Construction Rule “010” defines an octet aligned stream that is intended to be continuous without beginning or end.

NOTE – Example is a Video Stream.

The TFDZ Construction Rule “011” defines the start of a MAPA\_SDU contained within a single fixed length TFDZ. The Last Valid Octet Pointer is required to delimit the end of the MAPA\_SDU.

The TFDZ Construction Rule “100” defines a starting segment of a Service Data Unit. The SDU is segmented and starts in this TFDZ but does not end in this TFDZ.

The TFDZ Construction Rule “101” defines a continuation segment (i.e., continuing portion of a SDU contained in the previous frame in that VC).

The TFDZ Construction Rule “110” defines the ending segment of the Service Data Unit being transferred within this VC. The TFDZ is required to be of variable length.

The TFDZ Construction Rule “111” defines that no segmentation is used in the TFDZ. The TFDZ contains one MAPA\_SDU, or one or more complete Packets, or one or more PDUs of CCSDS recognized protocols. The contents of the TFDZ are defined by the Protocol Identifier Field.

###### PROTOCOL IDENTIFIER WITHIN USLP DATA ZONE

Bits 3–7 of the Transfer Frame Data Field Header shall contain the Protocol Identifier.

The Protocol Identifier shall identify the CCSDS recognized Protocol of the PDU contained within the TFDZ.

NOTES

1. The value ‘0000’ in the Protocol ID field signals that COP-1 directives are contained within the TFDZ.
2. The value ‘00001’ in the Protocol ID field signals that COP-P directives are contained within the TFDZ.
3. 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.
4. The Protocol IDs are registered and maintained within the SANA. See SANA registry, Protocol ID Within USLP Data Zone.

###### PROTOCOL IDENTIFIER EXTENSION WITHIN USLP DATA ZONE

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

The Protocol Identifier Extension shall identify the CCSDS recognized Protocol of the PDU contained within the TFDZ.

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

NOTE - The extended Protocol IDs are registered and maintained within SANA. See SANA registry, Protocol Identifier Extension within USLP Data Zone.

###### POINTER FIELD (FIRST HEADER POINTER or LAST VALID OCTET POINTER)

General

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.

NOTE - The first octet in the TFDZ shall be an offset of “0”.

When the value in the TFDF construction rule is “000” binary, the Pointer field contains the offset within the TFDZ to the first octet of the first packet or PDU that starts within the TFDZ. In this case, the Pointer is designated the First Header Pointer.

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

If the TFDZ contains only Idle Data, the First Header Pointer shall be set to ‘all ones minus one’.

NOTES

1. The purpose of the First Header Pointer is to facilitate delimiting of variable-length PDUs contained within the TFDZ, by pointing directly to the location of the first PDU from which its length may be determined.
2. The locations of any subsequent PDUs within the same TFDZ will be determined by calculating the locations using the length field of these PDUs.
3. If the last PDU 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* and points to the start of the PDU within Frame *N*.
4. If the last PDU 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 PDU and points to the start of the next PDU that begins in that Frame.

When the value in the TFDF Construction Rules is ‘001’ binary then the Pointer field contains the offset to the last octet of the MAP\_SDU being transferred, with the remaining octets being fill octets. In this case the Pointer field is designated as the Last Valid Octet Pointer.

NOTES

1. 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).
2. If the MAP\_SDU fills the entire TFDZ then the value in the Pointer Field will be “all ones”. Alternatively the TFDF Construction Rule “100” could be used for the same purpose.

If there are no idle data in the TFDZ (i.e., the TFDZ contains only valid user data), then the TFDZ Construction Rules shall be set to ‘100’ and the Last Valid Octet shall be set to the value ‘all ones’.

If there are no valid user data in the TFDZ (i.e., the TFDZ contains only idle data), the Last Valid Octet shall be set to the value ‘all ones minus one’.

#### TRANSFER FRAME DATA ZONE (TFDZ)

* + - * 1. The Transfer Frame Data Zone shall follow, without gap, the TFDF Header.
        2. The TFDZ shall contain either Packets (PDUs), MAP\_SDUs, Octet Stream, or Idle Data (a project-specified ‘idle’ pattern).

NOTE – 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.

NOTE - Idle data is used only with fixed length TFDZs.

* + - * 1. PDUs shall be inserted contiguously and in forward order into the TFDZ.

NOTE – The first and last PDUs of the TFDZ are not necessarily complete, since the first PDU may be a continuation of a PDU begun in the previous TFDZ, and the last PDU may continue in the subsequent TFDZ of the same Virtual Channel.

* + - * 1. When fixed length and continuous frames are required and insufficient PDUs (including Idle Packets) are available at release time of a Transfer Frame of a Virtual Channel carrying TFDFs, a TFDZ that contains only Idle Data shall be generated.
        2. When fixed length and continuous frames are required and no user octets are available at release time of a Transfer Frame of a Virtual Channel carrying MAP\_SDUs, a Transfer Frame Data Zone that contains only Idle Data shall be generated.

NOTES

1. TFDZs that contain only Idle Data are sent to maintain synchronous transmission of Transfer Frames, and also to transmit data in the Operational Control Field on a specific Virtual Channel when there are no user data (PDUs or MAP\_SDUs) to send.
2. A TFDZ that contains only Idle Data must not be confused with the OID Transfer Frame defined in 4.1.4.1.9.
3. A TFDZ that contains only Idle Data may be generated whenever it is necessary (even in the middle of transmission of a PDU that is split into multiple TFDZs).
4. Idle Data in the TFDZ should not be confused with the Idle Packet specified in reference [12].

### Operational Control Field

* + 1. If present, the Operational Control Field shall occupy the four octets following, without gap, the Transfer Frame Data Field.
    2. The Operational Control Field is optional; its presence or absence is signaled by the Operational Control Field Flag for each Virtual Channel.
    3. Bit 0 of the Operational Control Field shall contain a Type Flag with the following meanings:

1. the Type Flag shall be ‘0’, if the Operational Control Field holds a Type-1-Report which shall contain either a Communications Link Control Word (CLCW), see 4.2.1 or a Proximity Link Control Word (PLCW), see 4.2.2.3.2;
2. the Type Flag shall be ‘1’, if the Operational Control Field holds a Type-2-Report.

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

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

1. if this bit is ‘0’, the contents of the report are project-specific;
2. if this bit is ‘1’, the contents of the report shall contain an SDLS Common Link Security Report (CLSR) defined in reference [SDLS Extended Procedures - TBD].

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 [B6].

### Frame Error Control Field

#### General

* + - * 1. If present, the Frame Error Control Field shall occupy two or four octets following, without gap, the Operational Control Field if this is present, or the Transfer Frame Data Field if an Operational Control Field is not present.
        2. The Frame Error Control Field is optional; its presence or absence shall be established by management.
        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 which 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 Channel Coding Sublayer. This field may be mandatory depending on the selected options for the Channel Coding Sublayer.
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 E for their definition.

## PROTOCOL PROCEDURES AT THE SENDING END

### overview

This subsection describes procedures at the sending end associated with each of the functions shown in figure 4‑7. 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.

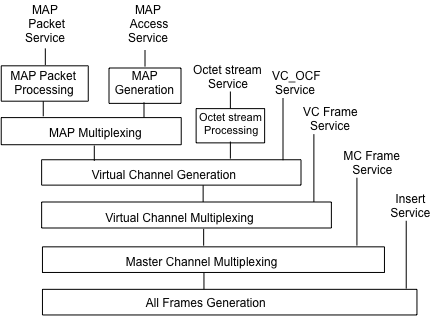


Figure 4‑12 : Internal Organization of Protocol Entity (Sending End)

### Packet Processing Function

* + 1. The Packet Processing Function shall be used to transfer variable-length Packets in the TFDZ of Transfer Frames.

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

* + 1. The TFDZ shall be constructed by concatenating Packets together until the maximum TFDZ length is exceeded. Any Packet which exceeds the maximum TFDZ length shall be split, filling the TFDZ completely, and starting a new TFDZ on the same Virtual Channel with the remainder. Construction of the next TFDZ shall continue with the concatenation of Packets until it overflows.
    2. If Packets of multiple versions are to be transferred on a Virtual Channel, Packets of these versions shall be multiplexed into a contiguous string of Packets before constructing TFDZs.
    3. 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. In the absence of sufficient Packets supplied from the users at release time, one or more Idle Packets of appropriate lengths may be created, where an Idle Packet is either
* an Idle Packet defined by reference [12], or
* an Encapsulation Idle Packet defined by reference [13].

NOTE – The shortest Idle Packet defined by reference [12] is 7-octets long (i.e., a 6-octet header plus 1 octet of idle data). If the area to be filled in a TFDZ is less than 7 octets, then the Idle Packet will spill over into the beginning of the next TFDZ. The shortest Idle Packet defined by reference [13] is one octet in length (i.e., a one-octet header).

* + 1. 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 Packet Processing Function is illustrated in figure 4-13.

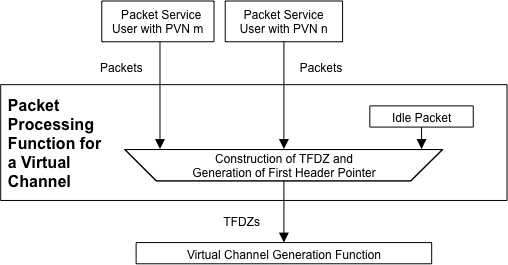


Figure 4‑13 : Abstract Model of Packet Processing Function

### Octet Stream Processing Function

* + 1. The Octet Stream Processing Function shall be used to transfer variable-length user defined octet stream in the TFDZ of Transfer Frames.

NOTE – There is an instance of the octet stream Processing Function for each MAP Channel that carries octet stream data.

* + 1. The Octet Stream Processing Function shall be used to fill the TFDZ with the Octet Stream Data supplied by the user. Each octet shall be placed sequentially, and unchanged, into the TFDZ. 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.

NOTE – If, because of the constraints of the Transfer Frame release algorithm, a TFDZ is not completely filled by Octet Stream Data at release time, the Octet Stream Processing Function will fill the remainder of the TFDZ with a locally specified idle pattern. The boundary between the end of the valid Octet Stream Data and the beginning of the idle data is indicated by setting the Last Valid Octet Field in the TFDF Header. If it is necessary, the Octet Stream Processing Function may generate an ‘idle’ TFDZ by setting the Last Valid Octet Field to ‘all ones minus one’.

NOTE – An abstract model of the Octet Stream Processing Function is illustrated in figure 4-14.

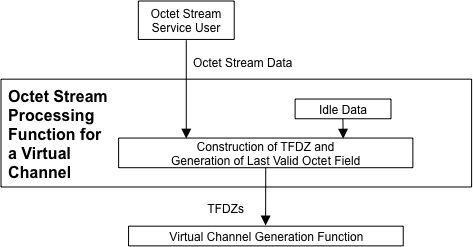


Figure 4‑14 : Abstract Model of Octet Stream Processing Function

### MAP Multiplexing Function

* + 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 is an instance of the MAP Multiplexing Function for each Virtual Channel that has multiple MAP Channels.

* + 1. The MAP Multiplexing Function shall multiplex service data units received from the instances of the Packet Processing and/or Octet Stream Processing functions and put them into a queue of Service Data Units in an appropriate order set by management.
    2. 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 4‑3.

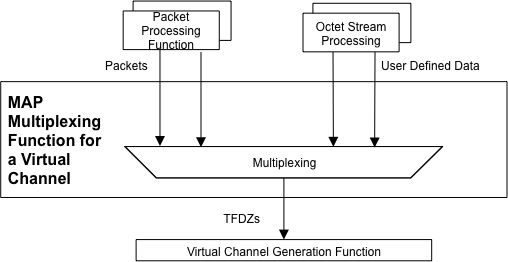


Figure 4‑3 : Abstract Model of MAP Multiplexing Function

### 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.

* + 1. The Virtual Channel Generation Function shall perform the following three procedures in the following order:

1. Maintain, increment and include the VC frame count in the TF primary header based upon the contents of the Bypass Flag field. This function also maintains the VC Frame Count Length field.
2. The Frame Operation Procedure (FOP)-P, which is a sub-procedure of the Communications Operation Procedure (COP)-P; and
3. the Frame Generation Procedure in this order.
   * 1. The FOP shall accept SDUs from the Packet Processing Function, Octet Stream Processing Function, a VCA Service User and shall control transmission and retransmission of frames by examining the report contained in the CLCWs and generating Control Commands.
     2. The Frame Generation Procedure shall generate Transfer Frames by attaching a Transfer Frame Primary Header to each TFDZ or SPDU delivered by the FOP.
     3. 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 TFDF Header fields. A Virtual Channel Frame Count shall be generated independently for each Virtual Channel and placed into the Primary Header.
     4. If there is a user of the VC\_OCF Service for a particular Virtual Channel, an OCF\_SDU supplied by the user shall be placed in the Operational Control Field.
     5. 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 4-15.

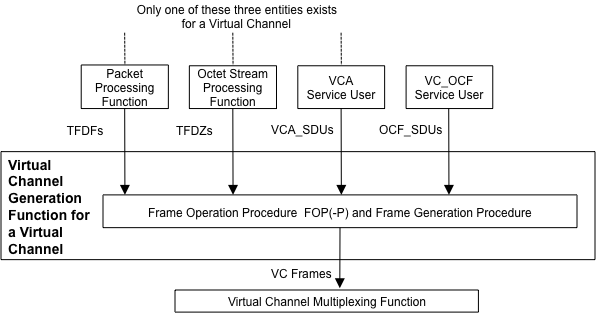


Figure 4‑15 : Abstract Model of Virtual Channel Generation Function

### Virtual Channel Multiplexing Function

* + 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.

* + 1. 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 may put the multiplexed Transfer Frames into a queue.

* + 1. 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.
    2. If there is only one Master Channel on the Physical Channel and when any of the coding schemes defined in references [3], [4], [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.

* + 1. 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 4-16.



Figure 4‑16 : Abstract Model of Virtual Channel Multiplexing Function

### Master Channel Multiplexing Function

* + 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.

* + 1. 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 may put the multiplexed Transfer Frames into a queue.

* + 1. 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.
    2. The Master Channel Multiplexing Function when any of the coding schemes defined in references [3], [4], [5] are used 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. It is not required to maintain a Virtual Channel Frame Count for OID Transfer Frames.

NOTE - 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.

NOTE – An abstract model of the Master Channel Multiplexing Function is illustrated in figure 4-17.



Figure 4‑17 : Abstract Model of Master Channel Multiplexing Function

### All FrameS Generation Function

* + 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.

* + 1. 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.
    2. If the Frame Error Control Field is present, check bits shall be generated using the encoding procedure described in 4.1.6.2 and inserted into the Transfer Frame Trailer. If this field is present, it must be present in all the Transfer Frames transmitted in a particular Physical Channel.
    3. Externally generated Transfer Frames associated with the Virtual Channel Frame and Mater Channel Frame Services shall always bypass the error control encoding functions specified above. The users of these Services must therefore 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. The All Frames Generation Function shall deliver data units to the underlying Channel Coding Sublayer.
    5. Each data unit delivered by the All Frames Generation Function shall consist of one or more Transfer Frames in accordance with parameters set by management.
    6. In accordance with parameters set by management, 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 one or more frames carrying service data on the Expedited Service.

NOTES

1. When systematic retransmissions of a data unit are requested, the improved reception probabilities from the retransmissions are increased if the data unit consists of a single Transfer Frame.
2. 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.
3. An abstract model of the All Frames Generation Function is illustrated in figure 4-18.

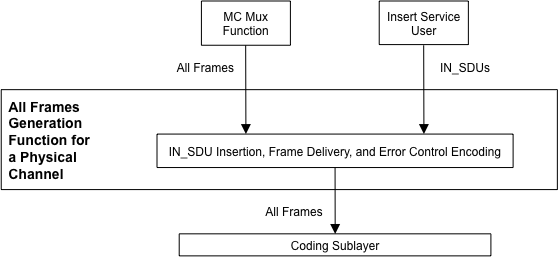


Figure 4‑18 : Abstract Model of All Frames Generation Function

## PROTOCOL PROCEDURES AT THE RECEIVING END

### overview

This subsection describes procedures at the receiving end associated with each of the functions shown in figure 4‑14. 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.

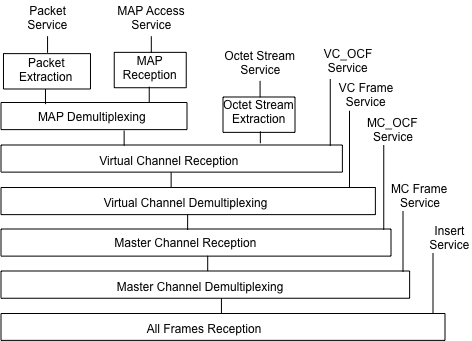


Figure 4‑14 : Internal Organization of Protocol Entity (Receiving End)

### Packet Extraction Function

* + 1. The Packet Extraction Function is used to extract variable-length Packets from the fixed-length M\_PDUs.

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

* + 1. The Packet Extraction Function shall extract Packets from M\_PDUs received from the Virtual Channel Reception Function. The First Header Pointer of each M\_PDU shall be used in conjunction with the length field of each Packet contained within the M\_PDU to provide the delimiting information needed to extract Packets.
    2. If the last Packet removed from the M\_PDU is incomplete, the Packet Extraction Function shall retrieve its remainder from the beginning of the next M\_PDU received on the same Virtual Channel. The First Header Pointer for the next M\_PDU shall be used to determine the length of the remainder and, hence, the beginning of the next Packet to be extracted.
    3. 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. Extracted Packets shall be delivered to the users on the basis of the Packet Version Number in their header.

NOTES

1. Incomplete Packets are not required to be delivered in cross support situations. Idle Packets are discarded. M\_PDUs that contain only Idle Data are also discarded.
2. An abstract model of the Packet Extraction Function is illustrated in figure 4‑15.

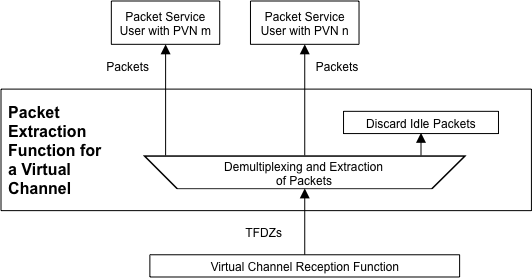


Figure 4‑15 : Abstract Model of Packet Extraction Function

### Octet stream extraction Function

* + 1. The Octet Stream Extraction Function shall be used to extract variable-length MAP\_SDUs from TFDZs on a MAP Channel.

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

* + 1. The Octet Stream Extraction Function shall extract MAP\_SDUs from TFDZs received from the MAP Demultiplexing Function.
    2. Original MAP\_SDUs shall be extracted and reconstructed from TFDZs using the Segmentation Flags and the Last Valid Octet Field within each of the Transfer Frame Data Field Headers.
    3. Extracted MAP\_SDUs shall be delivered to the user.

NOTE – An abstract model of the Octet Stream Extraction Function is illustrated in figure 4‑6.

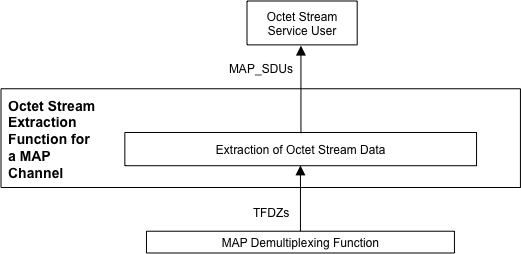


Figure 4‑6 : Abstract Model of MAP Reception Function

### MAP Demultiplexing Function

* + 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.

* + 1. 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 and/or Octet Stream Extraction Functions.
    2. 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 4‑7.

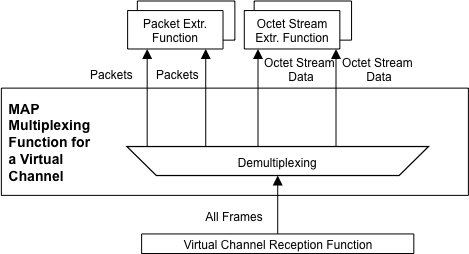


Figure 4‑7 : Abstract Model of MAP Demultiplexing Function

### Virtual Channel Reception Function

* + 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.

* + 1. 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 VCA Service user).
    2. If there is a user of the VC\_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.
    3. 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 4‑17.

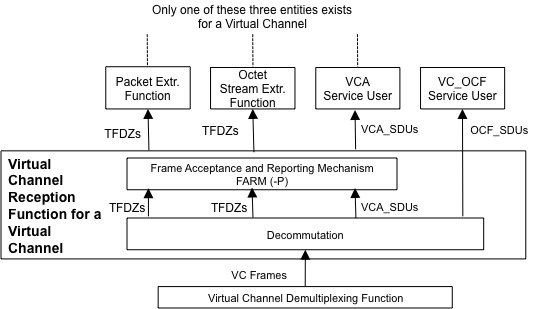


Figure 4‑17 : Abstract Model of Virtual Channel Reception Function

### Virtual Channel Demultiplexing Function

* + 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.

* + 1. 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.
    2. If a gap in the Virtual Channel Frame Count is detected, a Loss Flag may (optionally) be delivered to the users.

NOTES

1. OID Transfer Frames are discarded, when any of the coding schemes defined in references [3], [4], [5] are used . Transfer Frames with an invalid VCID are also discarded.
2. When any of the coding schemes defined in references [6] and [7] are used, then this function does not discard an OID Transfer Frame.
3. An abstract model of the Virtual Channel Demultiplexing Function is illustrated in figure 4‑18.



Figure 4‑18 : Abstract Model of Virtual Channel Demultiplexing Function

### Master Channel Demultiplexing Function

* + 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.

* + 1. 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.
    2. If frame loss is signaled by the underlying Channel Coding Sublayer, a Loss Flag may (optionally) be delivered to the users.

NOTES

1. Transfer Frames with an invalid MCID are discarded.
2. An abstract model of the Master Channel Demultiplexing Function is illustrated in figure 4‑19.



Figure 4‑19 : Abstract Model of Master Channel Demultiplexing Function

### All FrameS Reception Function

* + 1. The All Frames Reception Function shall be used to reconstitute Transfer Frames from the data stream provided by the Channel Coding Sublayer and to perform checks to determine whether the reconstituted Transfer Frames are valid or not.
       - 1. The All Frames Reception Function shall perform three procedures:

1. Frame Delimiting and Fill Removal Procedure; and
2. Frame Validation Check Procedure, in this order.
   * + - 1. The Frame Delimiting and Fill Removal Procedure shall be used to reconstitute Transfer Frames from the data stream provided by the Channel Coding Sublayer and to remove any Fill Data transferred from the Channel Coding Sublayer. See reference [6].
         2. 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.

* + 1. 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.

NOTE – A Transfer Frame which contains a detected error is not required to be delivered in cross support situations.

* + 1. 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 4‑20.

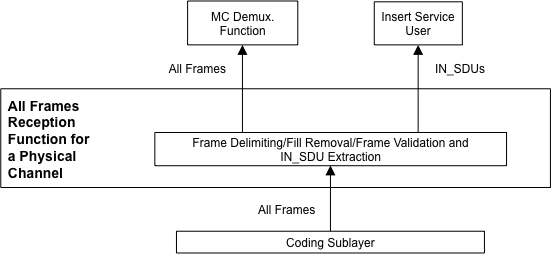


Figure 4‑20 : Abstract Model of All Frames Reception Function

#### Frame Delimiting and Fill Removal Procedure – TBD Needs input from Ken A

* + - * 1. The Channel Coding and Synchronization Recommended Standard (references [6]) shall be used as the specification for the Channel Coding Sublayer immediately below this procedure.
        2. At the sending end, the USLP Space Data Link Protocol shall pass one or more Transfer Frames to the Channel Coding Sublayer.

NOTE – The 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 Channel Coding Sublayer to ensure correct transmission of all valid data.

* + - * 1. The receiving end of this protocol shall receive as an input from the Channel Coding Sublayer a series of data octets, corresponding to the decoded Transfer Frame(s), which have been declared ‘clean’ by the Channel Coding Sublayer insofar as they contain no detected errors.

NOTE – The 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 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 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.

* + - * 1. If one or more valid Frame Length fields are detected by this Procedure and the number of octets received when the Data Stop condition occurs equals the number of octets specified by the Frame Length(s), then each Transfer Frame shall be passed on to the Frame Validation Check Procedure (see 4.4.6.5) as it is delimited.
        2. If a valid Frame Length field is detected by this 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 Channel Coding Sublayer.

* + - * 1. If a valid Frame Length field is detected by this 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

1. assume that the octets following the final expected octet of the frame are Fill Data appended by the sending end of the Channel Coding Sublayer to complete the last Codeblock (see reference [6]);
2. discard that Fill Data;
3. pass the Frame to the Frame Validation Check Procedure (see 4.4.8.5).

NOTES

1. Because the receiving end of the Channel Coding Sublayer cannot distinguish between valid data and Fill Data, the Fill Data must be stripped by this protocol.
2. The characteristics of the BCH Codeblock 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 4.4.6.5) 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 Channel Coding Sublayer, thus failing a test and causing the trailing five or six octets to be discarded.
3. TBD - Insert LDPC note here
4. 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.

#### Frame Validation Check Procedure

* + - * 1. The Frame Validation Checks shall be applied to all incoming Transfer Frames.
        2. Failure to pass any test within the Frame Validation Checks shall cause the Transfer Frame to be rejected (discarded).
        3. The Frame Validation Checks shall consist of the following tests:

1. The Transfer Frame must have an expected Transfer Frame Version Number.
2. The Transfer Frame must have one of the expected MCIDs (Transfer Frame Version Number and Spacecraft IDs).
3. The Transfer Frame Header must not contain any values which are not consistent with the implemented features for that spacecraft.
4. The value of the Frame Length must be consistent with the number of octets that are present.
5. 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.

# Managed Parameters without SDLS Option

## OVERVIEW OF MANAGED PARAMETERS

In order to conserve bandwidth on the space link, some parameters associated with the USLP Space Data Link Protocol 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 the USLP Space Data Link Protocol 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 the USLP Space Data Link Protocol without support for the SDLS protocol. Additional managed parameters for the USLP Space Data Link Protocol with the SDLS option are specified in 6.6.

## Managed Parameters for a Physical Channel

Table 5‑1 lists the managed parameters associated with a Physical Channel.

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 |
| Transfer Frame Length (octets) | Integer |
| Transfer Frame Version Number | ‘1100’ binary |
| Valid Spacecraft IDs | Set of Integers |
| MC Multiplexing Scheme | Mission Specific |
| Presence of Isochronous Insert Zone | Present, Absent |
| Isochronous 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 | Integer |
| 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 [CCSDS 131.0-B, 131.3-B]. It can be variable-length when USLP is used over [CCSDS 231.0-B] and [CCSDS 211.0-B]. 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 [CCSDS 131.0-B, 131.2-B, 131.3-B]. 4. The Maximum Number of Transfer Frames given to the Coding & Synchronizatio Sublayer applies only when USLP is used over [CCSDS 231.0-B]. |  |

## Managed Parameters for a MASTER Channel

Table 5‑2 lists the managed parameters associated with a Master Channel.

Table 5‑2 : Managed Parameters for a Master Channel

|  |  |
| --- | --- |
| Managed Parameter | Allowed Values |
| Transfer Frame Length (octets) | Integer |
| Spacecraft ID | 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 |
| NOTES  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).  2 The value of the Transfer Frame Version Number is the same for all transfer frames on a Physical Channel.  3 For VCID the binary value of ‘all ones’ (i.e. 63) is always valid as it is reserved for OID transfer frames by 4.1.4.1.9; i.e., the number of valid VCIDs always includes value 63 and the Selectable Set of Integers defined above. Generation of OID frames applies only when USLP is used over [CCSDS 131.0-B, 131.2-B, 131.3-B]. In the other cases VCID 63 may be used. | |

## Managed Parameters for a Virtual Channel

Table 5‑3 lists the managed parameters associated with a Virtual Channel.

Table 5‑3 : Managed Parameters for a Virtual Channel

|  |  |
| --- | --- |
| Managed Parameter | Allowed Values |
| Transfer Frame Length (octets) | Integer |
| Spacecraft ID | Integer |
| VCID | 0, 1, …, 62 (63 reserved) |
| Frame Sequence Control Supported | True, False |
| COP in Effect | COP-1, COP-P, None |
| CLCW Version Number | 1 |
| CLCW Reporting Rate | Real number per second |
| Valid MAP IDs | Set of integers (from 0 to 31) |
| MAP Multiplexing Scheme | Mission Specific |
| TFDF Length | Integer |
| Generate OID Frame | True, False |
| Fixed Length TFDZ | True, False |
| Inclusion of OCF Allowed | True, False |
| Inclusion of OCF Required | True, False |
| Inclusion of On-demand Insert Zone Allowed | True, False |
| Inclusion of On-demand Insert Zone 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 |
| 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 4.1.4.1.9 Generation of OID frames applies only when USLP is used over [CCSDS 131.0-B, 131.2-B, 131.3-B]. In the other cases, VCID 63 may be used. | |

## 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 | Integer |
| VCID | 0, 1, …, 63 |
| MAP ID | 0, 1, …, 31 |
|  | |

## Managed Parameters for PACKET TRANSFER

Table 5‑4 lists the managed parameters associated with a Virtual Channel used for the Virtual Channel Packet Service.

Table 5‑4 : 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 |

## Managed Parameters for a Service access point (SAP)

|  |  |
| --- | --- |
| Managed Parameter | Allowed Values |
| Sequence Controlled | True, False |
| TFDZ Length | Fixed length, Maximum length( if variable) |
| VCID | 0 - 62 |
| MAP ID | 0-31 |
| Packet Version Number | PVN |
| USLP Protocol ID | PID |

# Protocol Specification with SDLS OPTION

## Overview

This section specifies the protocol data unit and the procedures of the USLP Space Data Link Protocol with support for the Space Data Link Security Protocol (reference [14]). If the USLP Space Data Link 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.

## 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 the USLP Space Data Link Protocol to provide data authentication and data confidentiality at the Data Link Layer.

## USLP TRANSFER FRAME WITH SDLS

### OVERVIEW

To support the use of the SDLS security features, a Security Header and a Security Trailer are defined for an 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.

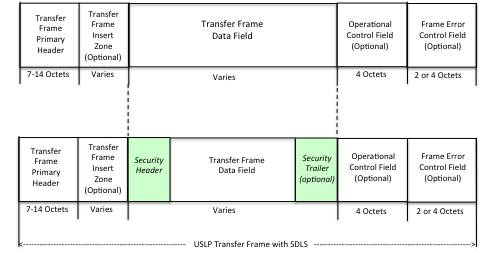


Figure 6‑1 : Frame without SDLS Compared to Frame with SDLS

### 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.

NOTES

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

### 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.

### 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.

### TRANSFER FRAME DATA FIELD IN A FRAME WITH SDLS

* + 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.
    2. In a Transfer Frame with SDLS, the Transfer Frame Data Field shall

1. 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.

1. 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.

### SECURITY TRAILER

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

NOTES

1. The Security Trailer is optional in an 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.

### OPERATIONAL CONTROL FIELD IN A FRAME WITH SDLS

* + 1. The Operational Control Field of a frame with SDLS shall conform to the specifications of 4.1.5.2 through 4.1.5.5 as modified by 6.3.7.2.
    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.

### FRAME ERROR CONTROL FIELD IN A FRAME WITH SDLS

* + 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, 4.1.6.2, 4.1.6.3, as modified by 6.3.8.2.
    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.

## SENDING END PROTOCOL PROCEDURES WITH SDLS

### OVERVIEW

When a secure USLP link is required, the USLP Space Data Link Protocol supports the use of the SDLS protocol. In this case, the USLP Space Data Link Protocol contains differences in the sending end procedures compared to the procedures described in 4.3. This subsection defines those differences.

The SDLS ApplySecurity Function may interface with the USLP Space Data Link Protocol at either the Virtual Channel Generation Function (4.3.4) or the Virtual Channel Multiplexing Function (4.3.5). 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.

### Packet Processing Function with SDLS

* + 1. The Packet Processing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.3.2 and 6.4.2.2.
    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).

### Octet Stream Processing Function with SDLS

* + 1. The Octet Stream Processing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.3.3 and 6.4.3.2.
    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).

### Virtual Channel generation Function with SDLS

* + 1. When assembling a Transfer Frame, the Virtual Channel Generation Function shall conform to the specifications of 4.3.4, 6.3, and 6.4.4.2 through 6.4.4.3.
    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).
   * 1. If the Virtual Channel Generation Function contains the interface to the SDLS protocol,
3. it shall call the SDLS ApplySecurity function for the Transfer Frames that it assembles for Virtual Channels that use SDLS;
4. the order of processing between the functions of the USLP and SDLS protocols shall occur as follows in the Virtual Channel Generation Function:
5. the frame assembly processing by theVirtual Channel Generation Function;
6. 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.

### Virtual Channel Multiplexing Function WITH SDLS

* + 1. The Virtual Channel Multiplexing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.3.5, 6.4.5.2.
    2. If the Virtual Channel Multiplexing Function contains the interface to the SDLS protocol,

1. 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;
2. the order of processing between the functions of the USLP and SDLS protocols shall occur as follows in the Virtual Channel Multiplexing Function:
3. the Virtual Channel multiplexing processing of the Virtual Channel Multiplexing Function;
4. the call by the Virtual Channel Multiplexing Function to the SDLS ApplySecurity Function.

### Master Channel Multiplexing Function WITH SDLS

The Master Channel Multiplexing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.3.6.

### All FRAMES Generation Function WITH SDLS

The All Frames Generation Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.3.7.

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.

## RECEIVING END PROTOCOL PROCEDURES WITH SDLS

### 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 4.4. 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.

### Error reporting

#### 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.

#### Requirements

* + - * 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).
        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 3.3.2.6, 3.4.2.5, and 3.5.2.5).

### Packet EXTRACTION Function with SDLS

* + 1. The Packet Extraction Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.2 and 6.5.3.2.
    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).

### Octet Stream EXTRACTION Function with SDLS

* + 1. The Octet Stream Extraction Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.3 and 6.5.4.2.
    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).

### Virtual Channel Reception Function with SDLS

* + 1. The Virtual Channel Reception Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.4 and 6.5.5.2 through 6.5.5.3.
    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.
    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.

### Virtual Channel DEMultiplexing Function WITH SDLS

* + 1. The Virtual Channel Demultiplexing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.5 and 6.5.6.2.
    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.

### Master Channel DEMultiplexing Function WITH SDLS

The Master Channel Demultiplexing Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.6.

### All FRAMES Reception Function WITH SDLS

The All Frames Reception Function of an USLP Protocol entity that supports SDLS shall conform to the specifications of 4.4.7.

## MANAGED PARAMETERS WITH SDLS

### Overview

Managed parameters for the SDLS protocol are specified in reference [14].

### ADDITIONAL MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

The managed parameters associated with a Virtual Channel for the USLP Space Data Link Protocol that supports the SDLS protocol shall conform to the definitions in table 5‑4 and the additional definitions in table 6‑1.

Table 6‑1 : Additional Managed Parameters for a Virtual Channel when USLP 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]. | |

1. Implementation Conformance   
   Statement (ICS) Proforma  
     
   (normative)
   1. INTRODUCTION
      1. OVERVIEW

This annex provides the Implementation Conformance Statement (ICS) Requirements List (RL) for an implementation of [Specification]. The ICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

* + 1. ABBREVIATIONS AND CONVENTIONS

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

Item Column

The item column contains sequential numbers for items in the table.

Feature Column

The feature column contains a brief descriptive name for a feature. It implicitly means “Is this feature supported by the implementation?”

Status Column

The status column uses the following notations:

* M mandatory;
* O optional;
* C conditional;
* X prohibited;
* I out of scope;
* N/A not applicable.

Support Column Symbols

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

Y Yes, supported by the implementation.

N No, not supported by the implementation.

N/A Not applicable.

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

* + 1. INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the Recommended Standard by completing the RL; that is, the state of compliance with all mandatory requirements and the options supported are shown. The resulting completed RL is called an ICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in 0. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

* 1. ICS PROFORMA FOR [SPECIFICATION]
     1. GENERAL INFORMATION
        1. Identification of ICS

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

* + - 1. Identification of Implementation Under Test

|  |  |
| --- | --- |
| Implementation Name |  |
| Implementation Version |  |
| Special Configuration |  |
| Other Information |  |

* + - 1. Identification of Supplier

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

* + - 1. Identification of Specification

|  |  |
| --- | --- |
| [CCSDS Document Number] | |
| Have any exceptions been required?  NOTE – A YES answer means that the implementation does not conform to the Recommended Standard. Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming. | Yes [  ]      No [  ] |

* + 1. REQUIREMENTS LIST

[See CCSDS A20.1-Y-1, *CCSDS Implementation Conformance Statements* (Yellow Book, Issue 1, April 2014).]

ANNEX\_B  
  
INFORMATIVE REFERENCES  
  
(Informative)

[B1] *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.

[B2] *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.

[B3] *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.

[B4] *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.

[B5] *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.

[B6] *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.

[B7] *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.

[B8] *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.

[B9] *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.

[B10] *Performance Characteristics for Intermediate Data Rate Digital Carriers Using Convolutional Encoding/Viterbi Encoding*. Rev. 10. IESS 308. Washington, DC: INTELSAT, 2000.

[B11] *USLP Green Book* Issue TBD. Report Concerning Space Data System Standards (Green Book), CCSDS 732.1-G-1. Washington, D.C.: CCSDS, TBD.

[B12] *Advanced Orbiting Systems, Networks and Data Links: Summary of Concept, Rationale and Performance.* Green Book. Issue 3. November 1992. CCSDS 700.0-G-3.

[B13] *TC Synchronization and Channel Coding--Summary of Concept and Rationale.* Green Book. Issue 2. November 2012. CCSDS 230.1-G-2.

[B14] *Proximity-1 Space Link Protocol--Rationale, Architecture, and Scenarios.* Green Book. Issue 2. December 2013. CCSDS 210.0-G-2.

[B15] *Telecommand Summary of Concept and Rationale.* Green Book. Issue 6. January 1987. CCSDS 200.0-G-6.

[B16] *TM Synchronization and Channel Coding--Summary of Concept and Rationale.* Green Book. Issue 2. November 2012. CCSDS 130.1-G-2.

[B17] *Next Generation Uplink.* Green Book. Issue 1. July 2014. CCSDS 230.2-G-1.

[B18] *Space Data Link Protocols--Summary of Concept and Rationale.* Green Book. Issue 3. September 2015. CCSDS 130.2-G-3.

[B19] *TM Space Data Link Protocol.* Blue Book. Issue 2. September 2015. CCSDS 132.0-B-2.

[B20] *AOS Space Data Link Protocol.* Blue Book. Issue 3. September 2015. CCSDS 732.0-B-3.

NOTE – Normative references are listed in 1.7.

ANNEX C  
  
Security, SANA, and Patent Considerations  
  
(Informative)

* + 1. Security Considerations
    2. security concerns with respect to the CCSDS document
       1. Data Privacy
       2. Data Integrity
       3. Authentication of Communicating Entities
       4. Control of Access to Resources
       5. Availability of Resources
       6. Auditing of Resource Usage
    3. Potential threats and attack scenarios
    4. Consequences of not applying security to the technology
  1. SANA Considerations

[See CCSDS 313.0-Y-1, *Space Assigned Numbers Authority (SANA)—Role, Responsibilities, Policies, and Procedures* (Yellow Book, Issue 1, July 2011).]

* 1. Patent Considerations

[See CCSDS A20.0-Y-4, *CCSDS Publications Manual* (Yellow Book, Issue 4, April 2014).]

ANNEX D  
VARIABLE-LENGTH SUPERVISORY  
PROTOCOL DATA FIELD FORMATS  
  
(INFORMATIVE)

NOTES

1. Table 4‑5 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.
2. 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.
   1. SPDU type 1: directive/report/plcw spdU data field
      1. General
         1. The Directive/Report/PLCW SPDU shall be used for space link supervisory configuration and control of the transceiver and its operation.
         2. The SPDU data field shall be a container that can hold up to seven sixteen-bit discrete self-delimiting and self-identifying directives:
3. each directive shall have a specific functionality;
4. each directive shall be sixteen bits in length and shall be self identified by the value in the Directive Type field (contained in bits 13, 14, and 15 of the directive);
5. the directives shall be concatenated without intervening bits within the data field.

NOTE – Figure A‑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 |
| Direction  (0) | | Freq Table (1) | | Rate Table  (2) | | Carrier Mod  (3,4) | | | | Data Mod (5,6) | | | Mode Select (7,8) | | | scrambler (9,10) | Diff.  Encoding (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 A‑1 : Type 1 SPDU Data Field Contents

* + 1. SET TRANSMITTER PARAMETERS DIRECTIVE
       1. General

The SET TRANSMITTER PARAMETERS directive shall consist of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. Transmitter Frequency (3 bits);
3. Transmitter Data Encoding (2 bits);
4. Transmitter Modulation (1 bit);
5. Transmitter Data Rate (4 bits);
6. Transmitter (TX) Mode (3 bits).

NOTE – The structural components of the SET TRANSMITTER PARAMETERS directive are shown in figure A‑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 A‑2 : SET TRANSMITTER PARAMETERS Directive

* + - 1. Directive Type
         1. Bits 13–15 of the SET TRANSMITTER PARAMETERS directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of protocol control directive and shall contain the binary value ‘000’ for the SET TRANSMITTER PARAMETERS directive.
      2. Transmitter Frequency
         1. General

Bits 10–12 of the SET TRANSMITTER PARAMETERS directive shall be used to set the transmitter frequency of the partnered transceiver to the desired value.

* + - * 1. Return Transmitter Frequency (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field shall define 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 |

* + - 1. Transmitter Data Encoding

Bits 8–9 of the SET TRANSMITTER PARAMETERS directive shall contain the following coding options:

1. ‘00’ = LDPC(2048,1024) rate 1/2 code (see reference [3]);
2. ‘01’ = Convolutional Code(7,1/2) (G2 vector inverted) with attached CRC-32 (see reference [7]).
3. ‘10’ = Bypass all codes;
4. ‘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 [B8] for more details.)

* + - 1. Transmitter Modulation

Bit 7 of the SET TRANSMITTER PARAMETERS directive shall contain the transmission modulation options:

1. ‘0’ = Coherent frequency PSK;
2. ‘1’ = Non-coherent frequency PSK.
   * + 1. Transmitter Data Rate
          1. Bits 3–6 of the SET TRANSMITTER PARAMETERS directive shall 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 A1.7).

* + - * 1. 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 |

* + - * 1. 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 |

* + - * 1. Proximity-1 coded symbol (Rcs) and data rate (Rd) table:

|  |  |  |  |
| --- | --- | --- | --- |
| Prox-1Coded Symbol Rates (Rcs) | Prox-1 Uncoded Data Rates (Rd) Rd = Rcs | Prox-1 Convolutionally Coded Data Rates (Rd) Rd = .5 \* Rcs | Prox-1 LDPC computed data rates (Rd) Rd = .48484 \* Rcs |
| 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 |

* + - * 1. 1, 512, 1024, and 2048 kb/s data rates can only be selected using the SET PL EXTENSIONS directive (see A1.7).
      1. Transmitter Mode

Bits 0–2 of the SET TRANSMITTER PARAMETERS directive shall contain the Transmitter Mode options. This field identifies the operating mode of the transmitter:

1. ‘000’ = Mission Specific;
2. ‘001’ = Proximity-1 Protocol;
3. ‘010’ = Mission Specific;
4. ‘011’ = Mission Specific;
5. ‘100’ = Mission Specific;
6. ‘101’ = Mission Specific;
7. ‘110’ = Reserved by CCSDS;
8. ‘111’ = Reserved by CCSDS.

NOTE – See reference [10] Annex F should be consulted for NASA Mars Survey Project 2001 Odyssey Orbiter Transmitter Mode mission-specific values.

* + 1. SET CONTROL PARAMETERS
       1. General
          1. The SET CONTROL PARAMETERS directive shall consist of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. Token (1 bit);
3. Remote No More Data (1 bit);
4. Reserved (2 bits);
5. Duplex (3 bits);
6. Time Sample (6 bits).
   * + - 1. 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 A‑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 A‑3 : SET CONTROL PARAMETERS Directive

* + - 1. Directive Type
         1. Bits 13–15 of the SET CONTROL PARAMETERS directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of protocol control directive and shall contain the binary value ‘001’ to identify the SET CONTROL PARAMETERS directive.
      2. Token

Bit 12 of the SET CONTROL PARAMETERS directive shall contain 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.

* + - 1. Remote No More Data

Bit 11 of the SET CONTROL PARAMETERS directive shall contain 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.

* + - 1. Reserved

Bits 9-10 of the SET CONTROL PARAMETERS directive shall contain spares and be set to ‘all zero’.

* + - 1. Duplex

Bits 6-8 of the SET CONTROL PARAMETERS directive shall 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.

* + - 1. Time Sample

Bits 0-5 of the SET CONTROL PARAMETERS directive shall 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.

* + 1. SET RECEIVER PARAMETERS directive
       1. General

The SET RECEIVER PARAMETERS directive shall consist of six fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. Receiver Frequency (3 bits);
3. Receiver Data Decoding (2 bits);
4. Receiver Modulation (1 bit);
5. Receiver Data Rate (4 bits);
6. Receiver (RX) Mode (3 bits).

NOTE – The structural components of the SET RECEIVER PARAMETERS directive are shown in figure A‑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 A‑4 : SET RECEIVER PARAMETERS Directive

* + - 1. Directive Type
         1. Bits 13–15 of the SET RECEIVER PARAMETERS directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of protocol control directive and shall contain the binary value ‘010’ for the SET RECEIVER PARAMETERS directive.
      2. Receiver Frequency
         1. General

Bits 10–12 of the SET Receiver PARAMETERS directive shall be used to set the receiver frequency of the partnered transceiver to the desired value.

* + - * 1. Forward Receive Frequency (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field shall define 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 |

* + - 1. Receiver Data Decoding

Bits 8–9 of the SET RECEIVER PARAMETERS directive shall 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 [H8] for more details.)

* + - 1. Receiver Modulation

Bit 7 of the SET RECEIVER PARAMETERS directive shall contain the following transmission modulation options:

1. ‘0’ = Coherent frequency PSK;
2. ‘1’ = Non-coherent frequency PSK.
   * + 1. Receiver Data Rate
          1. Bits 3–6 of the SET receiver PARAMETERS directive shall 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 A1.7).

* + - * 1. 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 |

* + - * 1. 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 |

* + - * 1. Proximity-1 coded symbol (Rcs) and data rate (Rd) table:

|  |  |  |  |
| --- | --- | --- | --- |
| Prox-1 Coded Symbol Rates (Rcs) | Prox-1 Uncoded Data Rates (Rd) Rd = Rcs | Prox-1 Convolutionally Coded Data Rates (Rd) Rd = .5 \* Rcs | Prox-1 LDPC Computed Data Rates (Rd) Rd = .48484 \* Rcs |
| 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 A1.7).

* + - 1. Receiver Mode

Bits 0–2 of the SET RECEIVER PARAMETERS directive shall contain the receiver mode options. This field identifies the operating mode of the receiver:

1. ‘000’ = Mission Specific;
2. ‘001’ = Proximity-1 Protocol;
3. ‘010’ = Mission Specific;
4. ‘011’ = Mission Specific;
5. ‘100’ = Mission Specific;
6. ‘101’ = Mission Specific;
7. ‘110’ = Reserved by CCSDS;
8. ‘111’ = Reserved by CCSDS.

NOTE – See reference [10] Annex F should be consulted for NASA Mars Survey Project 2001 Odyssey Orbiter Receiver Mode mission-specific values.

* + 1. SET V(R) directive
       1. General

The SET V(R) directive shall consist of three fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. Spare (5 bits);
3. Receiver Frame Sequence Number (SEQ\_CTRL\_FSN) (8 bits).

NOTE – The structural components of the SET V(R) directive are shown in figure A‑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 A‑5 : SET V(R) Directive

* + - 1. Directive Type
         1. Bits 13–15 of the SET V(R) directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of protocol control directive and shall contain the binary value ‘011’ to identify the SET V(R) directive.
      2. Spare

Bits 8–12 of the SET V(R) directive shall contain spare bits, set to ‘all zero’.

* + - 1. Receiver Frame Sequence Number

Bits 0–7 of the SET V(R) directive shall contain the value of the Frame Sequence Number (SEQ\_CTRL\_FSN) to which the receiving unit of the partnered transceiver is to be set.

* + 1. REPORT REQUEST directive
       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 shall consist of seven fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. PCID 1 PLCW Request (1 bit);
3. PCID 0 PLCW Request (1 bit);
4. Time-Tag Request (3 bits);
5. Status Request (5 bits);
6. Spare (3 bits).

NOTE – The structural components of the REPORT REQUEST directive are shown in figure A‑6.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit 0 |  | |  |  | |  | Bit 15 | |
| Spare   3 bits | Status Report Request  5 bits | Time-Tag Request  3 bit | | PCID 0 PLCW Request 1 bit | PCID 1 PLCW Request 1 bit | | | Directive Type  3 bits |
| 0,1,2 | 3,4,5,6,7 | 8,9,10 | | 11 | 12 | | | 13,14,15 |

Figure A‑6 : Report Request

* + - 1. Directive Type
         1. Bits 13–15 of the REPORT REQUEST directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of protocol control directive and shall contain the binary value ‘100’.
      2. Physical Channel ID 1 PLCW Report Request Field

Bit 12 of the REPORT REQUEST directive shall indicate 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.

* + - 1. Physical Channel ID 0 PLCW Report Request Field

Bit 11 of the REPORT REQUEST directive shall indicate 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.

* + - 1. Time-Tag Request Field

Bits 8–10 of the directive, if set to a value other than ‘000’, shall indicate a request to the remote transceiver to initiate a Proximity-1 time tag exchange (see reference [10] section 5).

* + - 1. Status Report Request
         1. The value contained in bits 3–7 of the REPORT REQUEST directive shall indicate the type of status report desired.
         2. If set to ‘00000’, a status report is not required.
         3. The types of status reports are reserved for CCSDS use.
      2. Spares

Bits 0–2 of the REPORT REQUEST directive shall contain spare bits set to ‘all zero’.

* + 1. SET PL EXTENSIONS

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**.**

* + - 1. General

The SET PL extensions directive shall consist of ten fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. R-S Code (1 bit);
3. Differential Mark Encoding (1 bit);
4. Scrambler (2 bits);
5. Mode Select (2 bits);
6. Data Modulation (2 bits);
7. Carrier Modulation (2 bits);
8. Rate Table (1 bit);
9. Frequency Table (1 bit);
10. Direction (1 bit).

NOTE – The structural components of the SET PL EXTENSIONS directive are shown in figure A‑7.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit 0 | |  | | | |  | |  | | |  | Bit 15 | |
| Direction   1 bit | Freq Table  1 bit | | Rate Table  1 bit | Carrier MOD  2 bits | Data MOD  2 bits | | Mode Select  2 bits | | Scrambler   2 bits | Differential Mark Encoding 1 bit | | R-S Code  1 bit | Directive Type  3 bits |
| (0) | (1) | | (2) | (3,4) | (5,6) | | (7,8) | | (9,10) | (11) | | (12) | 13,14,15 |

Figure A‑7 : SET PL EXTENSIONS

* + - 1. Directive Type
         1. Bits 13–15 of the SET PL EXTENSIONS directive shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the directive type and shall contain the binary value ‘110’.
      2. Reed-Solomon Code

Bit 12 of the SET PL EXTENSIONS directive shall indicate which R-S Code is used:

1. ‘0’ = R-S(204,188) code;
2. ‘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.

* + - 1. Differential Mark Encoding

Bit 11 of the SET PL EXTENSIONS directive shall indicate whether Differential Mark Encoding is enabled:

1. ‘0’ = No differential encoding;
2. ‘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).

* + - 1. Scrambler

Bits 9-10 of the SET PL EXTENSIONS directive shall indicate if and what type of digital bit scrambling is used:

1. ‘00’ = Bypass all bit scrambling;
2. ‘01’ = CCITT bit scrambling enabled (see reference [B9]);
3. ‘10’ = Bypass all bit scrambling;
4. ‘11’ = IESS bit scrambling enabled (see reference [B10]);

None of these Scrambler options are specified by CCSDS in other Recommended Standards and therefore they are not required for cross-support.

* + - 1. Mode Select

Bits 7-8 of the SET PL EXTENSIONS directive shall indicate the type of carrier suppression used:

1. ‘00’ = Suppressed Carrier (Requires transmit side utilize Modulation Index of 90°and transmit/receive sides utilize Differential Mark Encoding/Decoding);
2. ‘01’ = Residual Carrier;
3. ‘10’ = Reserved;
4. ‘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.)

* + - 1. Data Modulation

Bits 5-6 of the SET PL EXTENSIONS directive shall indicate the type of data modulation used:

1. ‘00’ = NRZ-L;
2. ‘01’ = Bi-Phase-Level (Manchester);
3. ‘10’ = Reserved;
4. ‘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).

* + - 1. Carrier Modulation

Bits 3-4 of the SET PL EXTENSIONS directive shall indicate the type of carrier modulation to be used:

1. ‘00’ = No Modulation;
2. ‘01’ = PSK;
3. ‘10’ = FSK;
4. ‘11’ = QPSK.

Options c) and d) are not required for cross-support.

* + - 1. Rate Table

Bit 2 of the SET PL EXTENSIONS directive shall indicate which set of data rates prior to encoding shall be used.

1. ‘0’ = Default Set defined in the Data Rate Field of the Set Transmitter Parameters and Set Receiver Parameters Directives in this annex;
2. ‘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.

* + - 1. Proximity-1 coded symbol (Rcs) and data rate (Rd) table:

|  |  |  |  |
| --- | --- | --- | --- |
| Prox-1 Coded Symbol Rates (Rcs) | Prox-1 Uncoded Data Rates (Rd) Rd = Rcs | Prox-1 Convolutionally Coded Data Rates (Rd) Rd = .5 \* Rcs | Prox-1 LDPC Computed Data Rates (Rd) Rd = .48484 \* Rcs |
| 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 |

* + - 1. Frequency Table
         1. General

Bit 1 of the SET PL EXTENSIONS directive shall indicate what set of frequencies is to be used:

1. ‘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;
2. ‘1’ = Channels 8 – 15 defined in the Extended Physical Layer Frequency Set defined below.
   * + - 1. Forward Link (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the forward link, this 3-bit field shall define 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 |

* + - * 1. Return Link (e.g., Orbiter as Initiator; Landed Asset as Responder)

In the context of the return link, this 3-bit field shall define 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 |

* + - 1. Direction

Bit 0 of the SET PL EXTENSIONS directive shall indicate if the fields in this directive apply to the transmit or receive side of the transceiver:

1. ‘0’ = transmit side;
2. ‘1’ = receive side.
   * 1. REPORT SOURCE SPACECRAFT ID
        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 shall consist of three fields, positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0):

1. Directive Type (3 bits);
2. Reserved (3 bits);
3. Source Spacecraft ID (10 bits).

NOTE – The structural components of the REPORT SOURCE SPACECRAFT ID are shown in figure A‑8.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Bit 0 |  |  |  | |  | Bit 15 | |
| Source Spacecraft ID  10 bits | | | | Reserved  3 bits | | | Directive Type  3 bits |
| 0,1,2,3,4,5,6,7,8,9 | | | | 10,11,12 | | | 13,14,15 |

Figure A‑8 : Report Source Spacecraft ID

* + - 1. Directive Type
         1. Bits 13–15 of the REPORT Source Spacecraft Id status report shall contain the Directive Type.
         2. The 3-bit Directive Type field identifies the type of status report and shall contain the binary value ‘111’.
      2. Reserved

Bits 10–12 of the REPORT Source Spacecraft Id status report shall contain reserved bits, set to ‘all zero’.

* + - 1. Source Spacecraft ID

Bits 0-9 of the REPORT Source Spacecraft Id status report shall contain the SCID of the source of the transfer frame.

* 1. SPDU TYPE 2: TIME DISTRIBUTION SPDU DATA FIELD

NOTE – Table 4‑5 should be consulted for a complete overview of the variable-length SPDU structure including the SPDU header and SPDU data field.

* + 1. General
       1. The Time Distribution SPDU data field is the container that describes both the type and value of the time entity for distribution.
       2. A single Time Distribution directive shall be contained within a Time Distribution SPDU.
       3. The format of the Time Distribution SPDU data field shall consist 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 shall comply with the CCSDS Unsegmented Time Code format (reference [16]).

NOTE – The structural components of the TIME DISTRIBUTION SPDU data field are shown in figure A‑9.

|  |  |  |  |
| --- | --- | --- | --- |
| Octet 0 |  |  | Octet 14 |
| Directive Type  1 Octet | Transceiver Clock  8 Octets | Send Side Delay  3 Octets | One-Way-Light-Time  3 Octets |

Figure A‑9 : Type 2 SPDU Data Field Contents

* + 1. TIME DISTRIBUTION directive type
       1. Octet 0 of the time distribution SPDU data field shall contain the time distribution Directive Type field indicating the function to be performed (if any) with the time contents.
       2. Time Distribution Types are:

1. ‘00000000’ = NULL;
2. ‘00000001’ = TIME TRANSFER;
3. all others = Reserved for CCSDS use.
   * 1. Transceiver Clock
        1. When the Time Distribution Type equals TIME TRANSFER,
4. octet 1 through octet 8 shall 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;
5. this time code field shall be divided into 5 octets of coarse time and 3 octets of fine time. (See reference [16].)
   * + 1. Otherwise, this field shall contain reserved bits, set to ‘all zero’.
     1. Send side delay
        1. When the Time Distribution Type equals TIME TRANSFER,
6. octet 9 through octet 11 shall 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’);
7. this time code field shall be divided into 1 octet of coarse time and 2 octets of fine time. (See reference [4].)
   * + 1. Otherwise, this field shall contain reserved bits, set to ‘all zero’.
     1. ONE WAY LIGHT TIME
        1. When the Time Distribution Type equals TIME TRANSFER, and when the mission has decided OWLT should be used,
8. octet 12 through octet 14 shall 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;
9. this time code field shall be divided into 1 octet of coarse time and 2 octets of fine time. (See reference [4].)
   * + 1. Otherwise, this field shall contain reserved bits, set to ‘all zero’.

ANNEX E  
FRAME ERROR CONTROL FIELD CODING PROCEDURES  
  
(Normative)

E1 CRC-16 Frame Error Control Field Coding Procedures

E1.1 CRC-16 FECF Encoding Procedure

The Frame Error Control Field is computed by applying Cyclic Redundancy Check (CRC) techniques. The Encoding Procedure shall accept an (*n*–16)-bit Transfer Frame, excluding the Frame Error Control Field, and generates a systematic binary (*n*,*n*–16) block code by appending a 16-bit Frame Error Control Field as the final 16 bits of the codeblock, where *n* is the length of the Transfer Frame.

NOTE – The Bit Numbering Convention as specified in 1.6.3 is applicable below.

The equation for the contents of the Frame Error Control Field is:

FECF = [(*X*16 · M(*X*)) + (*X*(*n*-16) · L(*X*))] modulo G(*X*)

= P0∙ *X*15 + P1∙ *X*14 + P2∙ *X*13 + … + P14∙ *X*1 + P15∙ *X*0

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 P0 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 M0 taken as the coefficient of the highest power of *X*;

L(*X*) is the presetting polynomial given by

L(*X*) = ;

G(*X*) is the generating polyno­mial given by

G(*X*) = *X*16 + *X*12 + *X*5 + 1.

NOTES

1. The *X*(*n*-16) ∙ 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 E-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 E-1 : Logic Diagram of the CRC-16 Encoder

##### CRC-16 FECF Decoding Procedure

The error detection syndrome, S(*X*), is given by

S(*X*) = [(*X*16 · C\*(*X*)) + (*X*n · L(*X*))] 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 C0\* 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 S0 taken as the coefficient of the highest power of *X*.

The received block C\*(*X*) equals the transmitted codeblock 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 C0 or E0 taken as the binary coefficient of the highest power of *X*.

NOTE – A possible syndrome polynomial generator implementation is shown in figure 6‑. 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 E-2 : Logic Diagram of the CRC-16 Decoder

#### CRC-32 Frame Error Control Field Coding Procedures

##### CRC-32 Encoding Procedure

For the encoding procedure, the (*n-*32)-bit transfer frame shall be the information message.

The encoding procedure shall accept an (*n-*32)-bit Version-3 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 ASM is NOT used for computing the CRC-32.

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 M0 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  M(*X*)] 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 R0 taken as the coefficient of the highest power of *X*.

The *n*-bit CRC-32–encoded block, expressed as a polynomial C(*X*) with binary coefficients, shall be:

C(*X*) = *X*32  M(*X*) + R(*X*)

The (*n*-32) bitsof the message are input in the order *M0*,, *M*n-33, and the *n* bits of the codeword are output in the order *C0*,, *Cn*-1 = *M0*,, *Mn*-33, *R*0,, *R*31.

NOTE – A possible implementation of an encoder is described in figure **E-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 E-3 : A Possible Implementation of the CRC-32 Encoder

##### CRC-32 FECF Decoding Procedure

The decoding procedure shall accept an *n*-bit received codeword, including the 32-bit Cyclic Redundancy Check, and generates a 32-bit syndrome. An error is detected if and only if at least one of the syndrome bits is non-‘zero’.

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 C0 or E0 taken as the binary coefficient of the highest power of *X*.

With C\*(*X*) being the *n*-bit received codeword with the first transferred bit being the most significant bit C0\* 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  C\*(*X*)] 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 S0 taken as the coefficient of the highest power of *X*.

NOTE – A possible implementation of the syndrome polynomial generator is described in figure **E-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 E-4 : A Possible Implementation of the CRC-32 Decoder

ANNEX F  
PROTOCOL DATA UNIT (CONTROL WORDS/DIRECTIVES)  
(Informative)

### COMMUNICATIONS LINK CONTROL WORD

#### General

* + - * 1. The Communications Link Control Word (CLCW), which is the protocol data unit transmitted from the receiving end to the sending end, shall provide 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.
   * + - 1. The CLCW shall consist of ten fields, positioned contiguously, in the following sequence:
4. Control Word Type (1 bit, mandatory);
5. CLCW Version Number (2 bits, mandatory);
6. Status Field (3 bits, mandatory);
7. COP in Effect (2 bits, mandatory);
8. Virtual Channel Identification (6 bits, mandatory);
9. Reserved Spare (2 bits, mandatory);
10. Flags (5 bits, mandatory);
11. FARM-B (Expedited) Frame Counter (2 bits, mandatory);
12. Reserved Spare (1 bit, mandatory);
13. Report Value (8 bits, mandatory).

NOTE – The structural components of the CLCW are shown in figure 6‑.



Figure 6‑10 : Communications Link Control Word

#### Control Word Type

* + - * 1. Bit 0 of the CLCW shall contain the Control Word Type.
        2. 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.
        3. This one-bit field shall be set to ‘0’.

#### CLCW Version Number

* + - * 1. Bits 1-2 of the CLCW shall contain the (Binary Encoded) CLCW Version Number.
        2. This two-bit field shall be 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.

#### Status Field

Bits 3-5 of the CLCW shall 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.

#### COP in Effect

* + - * 1. Bits 6-7 of the CLCW shall contain the COP in Effect parameter and shall be used to indicate the COP that is being used.
        2. For COP-1, this two-bit field shall be set to ‘01’.

NOTE – At present a single COP, COP-1, is defined in this Recommended Standard.

#### Virtual Channel Identification

Bits 8-13 of the CLCW shall 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.

#### Reserved Spare

* + - * 1. Bits 14-15 of the CLCW shall contain the Reserved Spare.
        2. These two bits are reserved by CCSDS for future application and shall be set to ‘00’.

#### Flags

##### General

Bits 16-20 of the CLCW shall contain the Flags specified in the following subsections.

##### No RF Available Flag

Bit 16 of the CLCW shall contain the No RF Available Flag.

The No RF Available Flag shall provide 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.

A setting of ‘0’ in the No RF Available Flag shall indicate 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).

A setting of ‘1’ in the No RF Available Flag shall indicate that the Physical Layer is not available and that Transfer Frames cannot be transferred without corrective action within the Physical Layer.

The single No RF Available Flag shall apply to all Virtual Channels and shall be 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.

##### No Bit Lock Flag

Bit 17 of the CLCW shall contain 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.

Use of the No Bit Lock Flag is optional; if used,

1. ‘0’ shall indicate bit lock has been achieved;
2. ‘1’ shall indicate bit lock has not been achieved.

The single No Bit Lock Flag shall apply to all Virtual Channels and shall be updated whenever a change is signaled by the Physical Layer.

If the No Bit Lock Flag is not used, it shall be 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.

##### Lockout Flag

Bit 18 of the CLCW shall contain the Lockout Flag.

The Lockout Flag shall be used to indicate the Lockout status of the FARM of a particular Virtual Channel.

A setting of ‘1’ in the Lockout Flag shall indicate 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.

A setting of ‘0’ in the Lockout Flag shall indicate that the FARM is not in Lockout.

Separate Lockout Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Lockout Flag are contained in reference [9].

##### Wait Flag

Bit 19 of the CLCW shall contain the Wait Flag.

The Wait Flag shall be 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.

A setting of ‘1’ (i.e., Wait) in the Wait Flag for a particular Virtual Channel shall indicate that all further Sequence-Controlled Transfer Frames on that Virtual Channel will be rejected by the FARM until the condition is cleared.

A setting of ‘0’ (i.e., Do Not Wait) in the Wait Flag shall indicate that the receiving end is able to accept and process incoming Sequence-Controlled Transfer Frames.

Separate Wait Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Wait Flag are contained in reference [9].

##### Retransmit Flag

Bit 20 of the CLCW shall contain 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.

A setting of ‘1’ in the Retransmit Flag shall indicate 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.

A setting of ‘0’ in the Retransmit Flag shall indicate that there are no outstanding Sequence-Controlled Transfer Frame rejections in the sequence received so far, and thus retransmissions are not required.

Separate Retransmit Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Retransmit Flag are contained in reference [9].

#### FARM-B (Expedited Frame) Counter

* + - * 1. Bits 21-22 of the CLCW shall contain the FARM-B (Expedited Frame) Counter.
        2. Separate FARM-B (Expedited Frame) Counters shall be 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.

#### Reserved Spare

* + - * 1. Bit 23 of the CLCW shall contain the Reserved Spare.
        2. This bit is reserved by CCSDS for future application and shall be set to ‘0’.

#### Report Value

* + - * 1. Bits 24-31 of the CLCW shall contain the Report Value.
        2. Separate Report Values shall be 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].

### Protocol Data Unit (SPDU)

#### 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 6‑2 and 6‑3.

SPDUs can be transmitted using only the Bypass 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 D for the detailed specification of variable-length SPDUs.)

#### General

* + - * 1. Fixed-length SPDUs shall consist of the following fields positioned contiguously in the following sequence:

1. SPDU Header (2 bits) consisting of:
2. SPDU Format ID (1 bit),
3. SPDU Type Identifier (1 bit);
4. 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.
   * + - 1. Variable-length SPDUs shall consist of the following fields positioned contiguously in the following sequence:
5. SPDU Header (1 octet) consisting of:
6. SPDU Format ID (1 bit),
7. SPDU Type Identifier (3 bits),
8. 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.

1. 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.

#### Fixed-Length SPDU

##### General

A ‘1’ in the SPDU Format ID field shall identify a 16-bit fixed-length SPDU. This format provides for only two fixed SPDUs (see table 6‑2), which shall be differentiated by the SPDU Type Identifier field:

1. a ‘0’ in the SPDU Type Identifier field identifies the SPDU as a PLCW;
2. a ‘1’ in the SPDU Type Identifier field is reserved for future CCSDS specification.

Table 6‑2 : 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 4.2.2.3.2) |
| Type F2 | ‘1’ | ‘1’ | | Reserved for CCSDS Use |

##### Type F1 SPDU: Proximity Link Control Word

###### General

The PLCW shall consist of seven fields positioned contiguously in the following sequence (described from least significant bit, Bit 15, to most significant bit, Bit 0—see figure 6‑):

1. Report Value (8 bits);
2. Expedited Frame Counter (3 bits);
3. Reserved Spares (2 bits);
4. Retransmit Flag (1 bit);
5. SPDU Type Identifier (1 bit);
6. SPDU Format ID (1 bit).

The PLCW shall be transmitted using the Bypass Flag set to ‘1’. (see 4.1.2.7.1)

NOTE – The structural components of the PLCW are shown in figure 6‑. (See Annex G for NASA Mars Surveyor Project 2001 Odyssey PLCW definition.)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit 0 |  |  | |  |  | |  | | Bit 15 |
| SPDU Header | | SPDU Data Field | | | | | | | |
| SPDU Format ID  1 bit | SPDU Type Identifier  1 bit | Retrans­mit Flag  1 bit | Reserved  Spares  2 bits | | | Expedited Frame Counter  3 bits | | Report Value (FSN) 8 bits | |

Figure 6‑11 : Proximity Link Control Word Fields

###### Report Value

Bits 8–15 of the PLCW shall contain the Report Value.

The Report Value field shall contain the value of V(R). See reference [10].

Separate Report Values shall be reported for each physical channel independent of the I/O port.

###### Expedited Frame Counter

Bits 5–7 of the PLCW shall contain the Expedited\_Frame\_Counter.

The Expedited\_Frame\_Counter shall provide a modulo-8 counter indicating that Expedited frames have been received.

###### Reserved Spares

Bits 3-4 of the PLCW shall contain the Reserved Spare bits.

The Reserved Spare bit field shall be set to ‘00’.

###### PLCW Retransmit Flag

Bit 2 of the PLCW shall contain the PLCW Retransmit Flag.

A setting of ‘0’ in the PLCW Retransmit Flag shall indicate that there are no outstanding frame rejections in the sequence received so far, and thus retransmissions are not required.

A setting of ‘1’ in the PLCW Retransmit Flag shall indicate that a received frame left a frame sequence number gap and that a retransmission of the expected frame is required.

#### Variable-Length SPDU

##### General

A ‘0’ in the SPDU Format ID field shall identify 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 6‑3. Currently three of these eight types are defined in the following two subsections. The remainder are reserved for future CCSDS specification.

Table 6‑3 : 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 1 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. | | | | |

##### Type 1 SPDU: Directives/Reports

An SPDU Type Identifier equal to ‘000’ shall identify 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 SPDUTypes are shown in table 6‑3. Formats of variable-length SPDU data fields are defined in annex D.

##### Type 2 SPDU: Time Distribution PDU

An SPDU Type Identifier equal to ‘001’ shall identify a Type 2 SPDU with a data field containing from 1 to 15 octets of Time Distribution supervisory data.

Octet 0 of the data field shall contain the time distribution directive type, followed by the actual time field value (1 to 14 octets).

NOTE – Variable-length SPDUTypes are shown in table 6‑3. Formats of variable-length SPDU data fields are defined in annex D.

##### Type 3 SPDU: Status Reports

An SPDU Type Identifier equal to ‘010’ shall identify 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 D.)

ANNEX G  
ABBREVIATIONS AND ACRONYMS  
(normative)

This annex lists the acronyms used in this Recommended Standard.

ARQ Automatic Repeat Request

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

USLP Unified Space Data Link Protocol

VC Virtual Channel

VCF Virtual Channel Frame

VCID Virtual Channel Identifier