The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

UNIFIED SPACE DATA LINK PROTOCOL

RECOMMENDED STANDARD

CCSDS 732.1-B-3

BLUE BOOK
June 2024

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

UNIFIED SPACE DATA LINK PROTOCOL

RECOMMENDED STANDARD

CCSDS 732.1-B-3

BLUE BOOK
June 2024

AUTHORITY

Issue:	Recommended Standard, Issue 3
Date:	June 2024
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 email address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
Email: 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 Recommended Standard for use in developing flight and ground systems for space missions and has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The Unified Space Data Link Protocol described herein is intended for missions that are cross-supported between Agencies of the CCSDS.

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 email address indicated on page i.

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

Member Agencies

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

Observer Agencies

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

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 732.1-B-1	Unified Space Data Link Protocol, Recommended Standard, Issue 1	October 2018	Original issue, superseded
CCSDS 732.1-B-2	Unified Space Data Link Protocol, Recommended Standard, Issue 2	October 2021	Second issue, superseded
CCSDS 732.1-B-3	Unified Space Data Link Protocol, Recommended Standard, Issue 3	June 2024	Current issue: – adds VC Packet and VC Access services; – changes the maximum number of frames per CLTU from one to greater than one to match the Telecommand specification; and – adds to the security section a table, a diagram, and text concerning the order of COP to SDLS, similar to the Telecommand specification.
EC 1	Editorial Change 1	October 2024	Addresses errors and inconsistencies in – page layout; – paragraph formatting and numbering; – capitalization in figure D-2.

NOTE – Change bars in the inside margin indicate changes from the second issue.

CONTENTS

CONTENTS (continued)

Section Page

Figure

CONTENTS (continued)

Figure Page

CONTENTS (continued)

Figure Page

Table

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

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

1.2 SCOPE

This Recommended Standard defines the USLP in terms of:

a) the services provided to the users of this protocol;

b) the Protocol Data Units (PDUs) employed by the protocol; and

c) the procedures performed by the protocol.

It does not specify:

a) individual implementations or products;

b) the implementation of service interfaces within real systems;

c) the protocol procedures specified in both the COP-1 (reference [9]) and the COP-P (reference [10]);

d) the security services specified in the SDLS protocol (reference [15]);

e) the flow control;

f) the methods or technologies required to perform the procedures; or

g) the management activities required to configure and control the protocol.

1.3 APPLICABILITY

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

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

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

1.4 RATIONALE

The CCSDS believes it is important to document the rationale underlying the recommendations chosen so that future evaluations of proposed changes or improvements will not lose sight of previous decisions. The USLP Green Book (reference [F11]) contains further details including the applicability to various space links and data rates.

1.5 DOCUMENT STRUCTURE

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

a) section 1 presents the purpose, scope, applicability, and rationale of this Recommended Standard and lists the conventions, definitions, and references used throughout the Recommended Standard;

b) section 2 provides an overview of USLP;

c) section 3 defines the services provided by the protocol entity;

d) section 4 specifies the PDUs and procedures employed by the protocol entity;

e) section 5 specifies the managed parameters used by the protocol entity;

f) section 6 specifies the protocol entity with support for the Space Data Link Security (SDLS) protocol;

g) annex A provides the Protocol Implementation Conformance Statement (PICS) proforma;

h) annex B specifies Frame Error Control Field (FECF) procedures;

i) annex C specifies the relationship of Proximity-1 (version-3) to USLP (version-4) Transfer Frames;

j) annex D specifies the truncated Transfer Frame;

k) annex E provides the security, Space Assigned Numbers Authority (SANA), and patent considerations;

l) annex F provides a list of informative references; and

m) annex G lists all acronyms used within this document

n) annex H contains informative implementation examples of how to randomize the Transfer Frame Data Zone of an Only Idle Data (OID) Transfer Frame.

1.6 CONVENTIONS AND DEFINITIONS

1.6.1 DEFINITIONS

1.6.1.1 Definitions from the Open Systems Interconnection 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, that is, in the sense that those terms are generally applicable to any of a variety of technologies that provide for the exchange of information between real systems. Those terms are:

a) blocking;

b) connection;

c) Data Link Layer;

d) entity;

e) flow control;

f) Network Layer;

g) peer entities;

h) Physical Layer;

i) protocol control information;

j) PDU;

k) real system;

l) segmenting;

m) service;

n) Service Access Point (SAP);

o) SAP address; and

p) Service Data Unit (SDU).

1.6.1.2 Definitions from OSI Service Definition Conventions

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

a) confirmation;

b) indication;

c) primitive;

d) request;

e) response;

f) service provider; and

g) service user.

1.6.1.3 Terms Defined in *Proximity-1 Space Link Protocol—Data Link Layer*

This Recommended Standard makes use of the following term defined in reference [10]:

Port ID: Identifier of the logical or physical port that is the destination for a user's SDU.

1.6.1.4 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.

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

commutation: 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 PDU sharing the same identifier.

decommutation: A function of decapsulating multiple data units, according to the formatting rule specified by the protocol definition, into separate data units, each from a different service.

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

idle data: A fixed-length, project-specified 'idle' pattern of binary digits, whose assignment is a project design choice.

isochronous: Characterized by occurring at equal intervals of time.

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

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

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

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

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

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

NOTE – The USLP Transfer Frame may be truncated or non-truncated. Within this document, depending on context, any of the terms ‘Transfer Frame’, ‘USLP Frame’, and ‘Version-4 (Transfer) Frame’ and ‘non-truncated frame’ may be used to refer to the ‘USLP Transfer Frame’. However the unqualified term ‘Transfer Frame’ refers to the ‘non-truncated Transfer Frame’.

1.6.1.5 Terms Adapted from *Communications Operation Procedure-1* and *Proximity-1 Space Link Protocol—Data Link Layer*

The term **Frame Operation Procedure (FOP)** is used in this Recommended Standard to refer to both FOP-1 (see reference [9]) and FOP-P (see reference [10]). Similarly, the term **Frame Acceptance and Reporting Mechanism (FARM)** is used in this Recommended Standard to refer to both FARM-1 (see reference [9]) and FARM-P (see reference [10]). In both cases, interpretation of which term (-1 or -P) applies depends on which **Communications Operation Procedure (COP),** COP-1 (see reference [9]) or COP-P (see reference [10]), governs the context.

The term **Directive** in the context of **FOP** refers to one of the FOP-1 Directives defined in table 4-1, ‘Directive Types’, of reference [9]. The term **Directive** in the context of **COP** refers to one of the Supervisory Protocol Data Unit (SPDU) types defined in annex B of reference [10], with the exception of the ‘Set V(R)’ Proximity-1 directive.

A **COP Control Command** is either one of the COP-1 Control Commands, ‘Unlock’ and ‘Set V(R)’, carried by Type-BC Transfer Frames, or the COP-P ‘Set V(R)’ Directive carried by a SPDU.

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

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

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

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

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

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

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

1.6.2.2 Informative Text

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

- Overview;
- Background;
- Rationale;
- Discussion.

1.6.3 CONVENTIONS

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be ‘Bit 0’; the following bit is defined to be ‘Bit 1’ and so on up to ‘Bit *N*–1’. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, that is, ‘Bit 0’ (see figure 1-1).

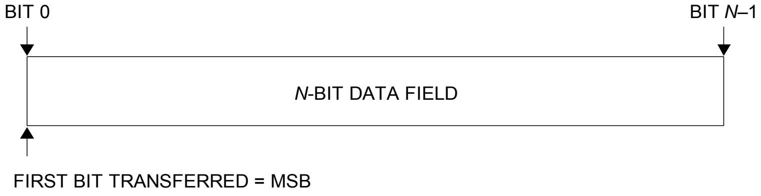


Figure 1-1: Bit Numbering Convention

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

1.7 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this document are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

[1] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. 2nd ed. International Standard, ISO/IEC 7498-1:1994. Geneva: ISO, 1994.

[2] *Information Technology—Open Systems Interconnection—Basic Reference Model—Conventions for the Definition of OSI Services*. International Standard, ISO/IEC 10731:1994. Geneva: ISO, 1994.

[3] *TM Synchronization and Channel Coding*. Issue 5. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.0-B-5. Washington, D.C.: CCSDS, September 2023.

[4] *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.2-B-2. Washington, D.C.: CCSDS, February 2023.

[5] *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.3-B-2. Washington, D.C.: CCSDS, April 2022.

[6] *TC Synchronization and Channel Coding*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-4. Washington, D.C.: CCSDS, July 2021.

[7] *Proximity-1 Space Link Protocol—Coding and Synchronization Sublayer*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.2-B-3. Washington, D.C.: CCSDS, October 2019.

[8] "Packet Version Number." Space Assigned Numbers Authority. https://sanaregistry.org/r/packet_version_number.

[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 6. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.0-B-6. Washington, D.C.: CCSDS, July 2020.

[11] *CCSDS Spacecraft Identification Field Code Assignment Control Procedures*. Issue 7. Recommendation for Space Data System Practices (Magenta Book), CCSDS 320.0-M-7. Washington, D.C.: CCSDS, November 2017.

[12] *Space Packet Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-2. Washington, D.C.: CCSDS, June 2020.

[13] *Encapsulation Packet Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.1-B-3. Washington, D.C.: CCSDS, May 2020.

[14] "USLP Protocol Identifier (UPID)." Space Assigned Numbers Authority (SANA). https://sanaregistry.org/r/uslp_protocol_id.

[15] *Space Data Link Security Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.0-B-2. Washington, D.C.: CCSDS, July 2022.

[16] *Space Data Link Security Protocol—Extended Procedures*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.1-B-1. Washington, D.C.: CCSDS, February 2020.

NOTE – Informative references are listed in annex F.

2 OVERVIEW

2.1 CONCEPT OF UNIFIED SPACE DATA LINK PROTOCOL

2.1.1 ARCHITECTURE

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

Figure 2-1 illustrates the relationship of this protocol to the reference model of Open Systems Interconnection (OSI) (reference [1]). Two sublayers of the Data Link Layer are defined for CCSDS Space Link Protocols (SLPs) as shown in reference [F2]. USLP corresponds to the Data Link Protocol Sublayer and provides functions for transferring various data using a variable-length (or fixed-length as a subset) PDU called the Transfer Frame. The Synchronization and Channel Coding Sublayer provides 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 set of Synchronization and Channel Coding Recommended Standards (references [3] through [7]) are to be used with USLP with constraints listed later in this document. How USLP is used in overall space data systems is shown in references [F2] through [F4] and [F11].

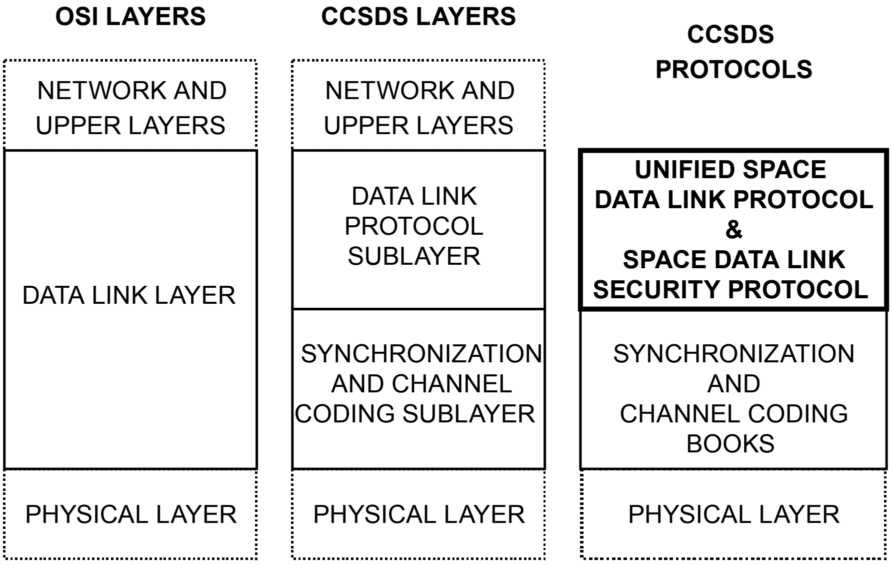

OSI LAYERS
CCSDS LAYERS
CCSDS PROTOCOLS
NETWORK AND UPPER LAYERS
NETWORK AND UPPER LAYERS
DATA LINK LAYER
DATA LINK PROTOCOL SUBLAYER
UNIFIED SPACE DATA LINK PROTOCOL & SPACE DATA LINK SECURITY PROTOCOL
SYNCHRONIZATION AND CHANNEL CODING SUBLAYER
SYNCHRONIZATION AND CHANNEL CODING BOOKS
PHYSICAL LAYER
PHYSICAL LAYER
PHYSICAL LAYER

Figure 2-1: Relationship with OSI Layers

2.1.2 PROTOCOL FEATURES

2.1.2.1 Transfer Frames, Virtual Channels, and Multiplexer Access Points

USLP provides the users with several services to transfer SDUs over a space link. These PDUs used by USLP to provide the services are known as USLP Transfer Frames (unless otherwise stated, the terms 'Transfer Frame', 'USLP Frame', and 'Version-4 (Transfer) Frame' in this document refer to the USLP Transfer Frame). Each Transfer Frame contains a primary header that provides protocol control information, identifying the length of the USLP Frame and signaling the inclusion of selected fields. The Transfer Frames carry upper-layer SDUs within the Transfer Frame Data Field (TFDF). The TFDF contains a TFDF header that identifies both how the data field is organized and the protocol to which the SDU(s) are associated.

A key feature of space data link protocols, including USLP, is the concept of Virtual Channels (VCs). The assignment of VCs allows the Physical Channel to be shared amongst 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 VC. Each Transfer Frame transferred over a Physical Channel belongs to one of the VCs of the Physical Channel.

Moreover, this protocol enables SDUs from different sources to be multiplexed together in one VC using Multiplexer Access Points (MAPs). MAP ID assignments allow SDUs arriving at a Service Access Point (SAP) at the sending end to be transferred to a SAP with the corresponding MAP ID at the receiving end. Each SAP can be associated with a specific protocol. USLP enables the transfer of CCSDS Packets, PDUs associated with protocols registered with SANA, or user defined data (Multiplexer Access Point Access Service Data Units [MAPA_SDUs], Vitual Channel Access Service Data Units [VCA_SDUs], or Octet Stream data). Reference [14] is the SANA registry of CCSDS-recognized protocol IDs.

2.1.2.2 Additional USLP Features

USLP has a larger maximum Transfer Frame size than previous CCSDS space data link protocols 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 than previous CCSDS space data link protocols. In order to accommodate a much larger frame-size range than previous space data link protocols, USLP provides a configurable-sized sequence counter in the Frame Primary Header.

2.1.2.3 Efficient Data Transfer

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

Because the underlying space link inherently includes a noisy signal path, there is a finite probability that it will introduce an error. It may be desirable to break large SDUs into relatively small pieces when the frame error rate for large USLP 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 SDU 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 SDUs are very small. For efficient transfer of SDUs, it is desirable to group these small units into larger pieces. USLP provides the capability to break large SDUs into relatively small pieces (i.e., segmentation) and to group small SDUs into larger pieces (i.e., blocking).

The COP, as described in 2.1.2.4, below, controls the transmission of SDUs through the space link, performing retransmissions needed to ensure delivery of SDUs in sequence and without gaps or duplication.

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

2.1.2.4 Communications Operation Procedure (Optional)

Within this document, the term COP refers to both the Communications Operation Procedure-1 (COP-1) (reference [9]), used in near-Earth and deep-space applications, and also the Communication Operations Procedure for Proximity links (COP-P) within the Proximity-1 Space Data Link Protocol (reference [10]). The Proximity-1 Green Book (reference [F14]) provides additional details.

The use of either the COP-1 or COP-P procedures is optional; both are compatible with USLP. The PDUs (Communications Link Control Word [CLCW] for COP-1 and Proximity Link Control Word [PLCW] for COP-P) and operational procedures for COP-1 and COP-P are not identical, but similar, and are transparent to USLP. For this reason, the generic terms FARM and FOP are used.

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

The COP fully specifies the closed-loop procedures executed by the sending and receiving ends of USLP. The COP consists of a pair of synchronized procedures for each VC: 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 QoS, that is, the delivery of SDUs 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.

For COP-1 over Telecommand Channel Coding in reference [6], correctness of the delivered SDUs may be guaranteed (within known error probabilities) by the optional FECF applied by the Data Link Protocol Sublayer, and by the Frame Validation Checks performed in USLP. For COP-P over Proximity-1 Channel Coding in reference [7], Frame Validation is carried out by the Proximity-1 Coding and Synchronization (C&S) Sublayer, since the FECF is defined outside of the Version-3 Transfer Frame.

Validation of the completeness, sequentiality, and non-duplication of the delivered SDUs on a particular VC requires that an accounting (i.e., numbering) scheme for Transfer Frames be implemented by the COP.

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 COP-1, Type BC Transfer Frames are also used to send Control Commands from the FOP to the FARM. In COP-P, a Type-1 SPDU is used to send Control Commands from FOP-P to the FARM-P. Only the user of the COPs Management Service (see 2.2.4.11) is allowed to send Control Commands.

2.1.2.5 Space Data Link Security Protocol (optional)

The SDLS protocol is specified in reference [15] as well as the SDLS Extended Procedures in reference [16]. The SDLS protocol may provide security, such as authentication and encryption, for USLP Transfer Frames. Support for the SDLS protocol is an optional feature of USLP.

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

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

2.1.3 ADDRESSING

There are four identifier fields in the header of Transfer Frames: Transfer Frame Version Number (TFVN), Spacecraft Identifier (SCID), VC 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 VC 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.

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

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

All Transfer Frames with the same GVCID and MAP ID constitute a MAP Channel. Thus a VC consists of one or multiple MAP Channels.

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

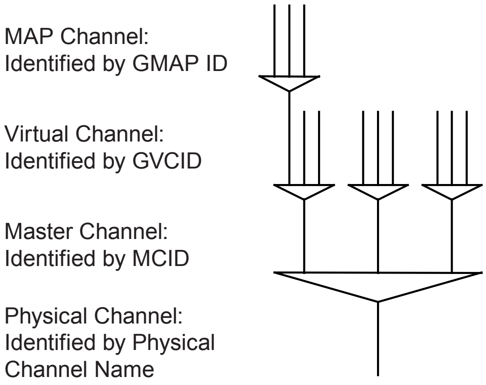


Figure 2-2: Relationships between Channels

2.1.4 PROTOCOL DESCRIPTION

USLP is described in terms of:

a) the services provided to the users;

b) the PDUs; and

c) the procedures performed by the protocol.

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

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

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

2.2 OVERVIEW OF SERVICES

2.2.1 INTRODUCTION

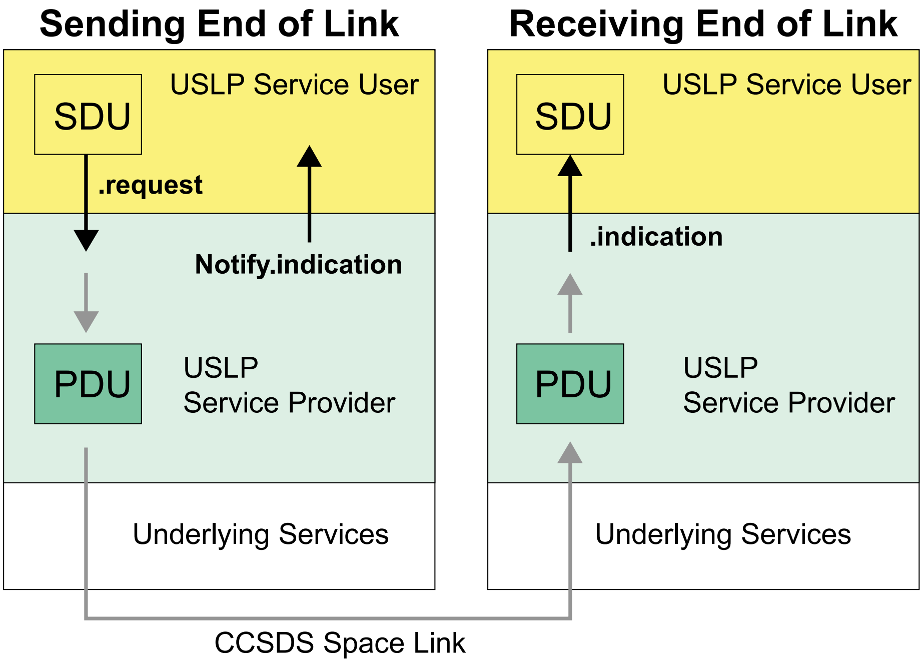


Figure 2-3: USLP Services Context Diagram

Figure 2-3 describes the interface between the service user, which provides the SDU, and the service provider, which performs the service of transferring the PDU across the space link. The user creates the .request primitive that contains an SDU ID to uniquely identify the SDU and supplies the data structure (e.g., packet, MAPA_SDU, or portion of an Octet Stream for transfer).

When Reliable Delivery is selected, in response to actually transferring that specific SDU, the Service Provider sends back a Notify.indication primitive containing that same exact SDU ID, so that the Service User knows for sure that the specific SDU (Identified by the SDU ID) was indeed transferred by the Service Provider. When Reliable Delivery is selected, the Notify.indication primitive may also report failed transfer or other events.

It should be noted that the SDU ID is not transferred across the space data link; that is, it does not get sent with the actual SDU. The SDU ID is for accounting purposes between the Service User and Service Provider on the sending end. It should also be noted that on the receive side there is only an .indication primitive transferred from the Service Provider to the Service User. Therefore on the receiving end there is no SDU ID exchanged.

2.2.2 COMMON FEATURES OF SERVICES

USLP provides users with data-transfer services. The point at which a service is provided by a protocol entity to a user is called a service access point, or SAP (see reference [1]). Each service user is identified by a SAP address that is associated with a specific ID for that specific service, for example, for MAP services a GMAP ID is required to identify that specific SAP.

SDUs submitted to a SAP are processed in the order of submission. No processing order is maintained for SDUs submitted to different SAPs.

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

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

a) unidirectional (one-way) services: One end of a connection can send, but not receive, data through the space link, while the other end can receive, but not send.

b) asynchronous services: There are no predefined timing rules for the transfer of SDUs supplied by the service user or for the transmission of Transfer Frames generated by the service provider. The user may request data transfer at any time, but there may be restrictions imposed by the service provider on the data generation rate. The timing of data transfer is determined by the provider in accordance with mission-specific rules and may depend on the traffic of the data exchange by service provider/lower layers at the time of transfer.

c) unconfirmed services: The sending user does not receive confirmation from the receiving end that data has been received.

d) incomplete services: The services do not guarantee completeness, but some services may signal gaps in the sequence of SDUs delivered to the receiving user.

e) sequence-preserving services: The sequence of SDUs 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 SDUs 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 can be extended or enhanced with Space Link Extension Services (reference [F5]).

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

a) The use of the COP procedures adds the capability for in-order delivery of SDUs to the layer above without error, omission, or duplication at the receiving end.

b) The use of the SDLS protocol adds the capability to authenticate the sender and/or to encrypt the data to provide security for specific VCs defined to use the SDLS security services.

2.2.3 SERVICE ATTRIBUTES

2.2.3.1 Overview

USLP in association with the COP-1 (reference [9]), COP-P (reference [10]), and SDLS (reference [15]), provide the following service attributes (sequence-controlled, expedited, authenticated, and/or encrypted) that determine the QoS associated with how reliably SDUs supplied by the sending user are delivered to the receiving user.

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

The SDLS protocol provides the user with the capability of authenticating the Frame originator and/or encrypting the Frame contents. SDLS security services are established by the use of the managed parameters in section 6.

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

These service attributes are provided at any SAP except for the Virtual Channel Frame (VCF) and Master Channel Frame (MCF).

For the VCF and MCF Services, the service provider does not make any distinction between Sequence-Controlled and Expedited service QoS applicable to SDUs supplied by the user. The user should request necessary procedures to provide Sequence-Controlled and/or Expedited Service QoS for its SDUs.

2.2.3.2 Sequence-Controlled Service—Reliable Delivery

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

For Sequence-Controlled Service, SDUs 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 VC in the order in which they are presented at the SAP. The retransmission mechanism ensures with a high probability of success that:

a) no SDU is lost;

b) no SDU is duplicated; and

c) no SDU is delivered out of sequence.

2.2.3.3 Expedited Service—Best Effort Delivery

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

For Expedited Service, SDUs supplied by the sending user are transmitted one or more times (see reference [6]). There is no guarantee that all Expedited SDUs are delivered to the receiving user.

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

2.2.3.4 Security Service

The optional use of the SDLS protocol (references [15] and [16]) provides all its functions (authentication, encryption, authenticated encryption) for the data in the Transfer Frame Data Zone (TFDZ) of a USLP Frame. It provides full protection for the service data of the MAP Packet (MAPP) service, the MAP Octet Stream Service, the MAP Access (MAPA) Service, the Virtual Channel Packet (VCP) Service, and the Virtual Channel Access (VCA) 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 Master Channel Operational Control Field (USLP_MC_OCF) Service or for the Insert Service. It also provides no protection for the frames supplied to USLP by external sources such as by the VCF and the MCF services.

2.2.3.5 Asynchronous Service

In asynchronous service, there are no timing relationships between the transfer of SDUs 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 2-4), each SDU 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 SDUs from the sending user are transferred, and transferred only once.

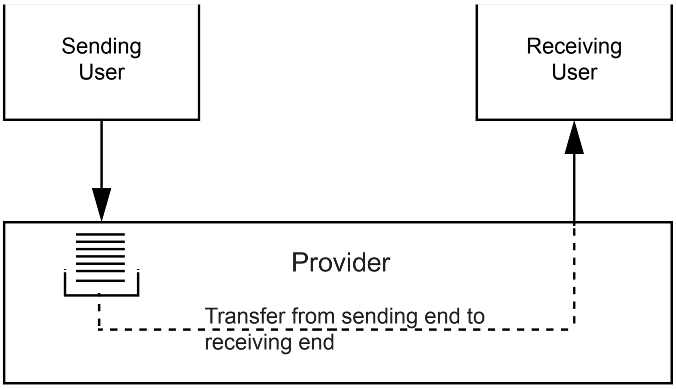


Figure 2-4: Asynchronous Service Model

2.2.3.6 Synchronous Service

In Synchronous Service (figure 2-5), each SDU from a sending user is placed in a buffer that can hold only one SDU; 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 transmission data rate and the fixed size of the USLP Frame. 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 SDU 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).

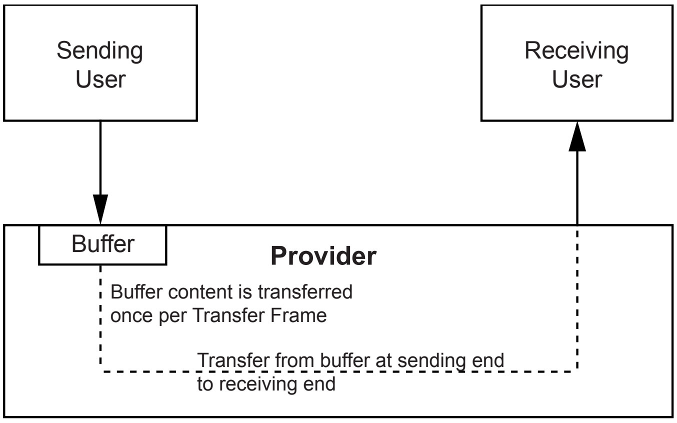

Sending User
Receiving User
Buffer
Provider
Buffer content is transferred once per Transfer Frame
Transfer from buffer at sending end to receiving end

Figure 2-5: Synchronous Service Model

2.2.3.7 Periodic Service

Periodic transfer from service interface to service interface is provided with a specified maximum delay and a specified maximum jitter at the service interface. There is one case in which periodic service is synchronous: when the service is associated with an MC, and that MC produces Transfer Frames that are transmitted back-to-back.

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

2.2.4 SUMMARY OF SERVICES

2.2.4.1 Introduction

The following data transfer services are provided by USLP:

- MAPP, MAPA, and MAP Octet Stream are provided for a MAP Channel;
- VCP, VCA, and VCF is provided for a VC;
- USLP_MC_OCF and MCF are provided for an MC;
- Insert is provided for all Transfer Frames on a Physical Channel.

In addition, the protocol provides the COPs Management Service, which is used at the sending end to control the COPs automatic retransmission procedure of a VC.

Table 2-1 summarizes these services and shows their characteristics, the 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; and
- authentication, to confirm the source and integrity of the data.

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

Service	Service Attribute	Service Data Unit	SAP Address	SDLS Security Features
MAPP	Sequence-Controlled and Expedited, Asynchronous	Packet	GMAP ID + PVN	All
MAPA	Sequence-Controlled and Expedited, Asynchronous	MAPA_SDU	GMAP ID	All
MAP Octet Stream	Asynchronous	Octet Stream Data	GMAP ID	All
VCP	Sequence-Controlled and Expedited, Asynchronous	Packet	GVCID	All
VCA	Sequence-Controlled and Expedited, Asynchronous	VCA_SDU	GVCID	All
USLP_MC_OCF	Synchronous or Asynchronous	OCF_SDU	GVCID	None
VCF	Asynchronous or Synchronous	Transfer Frame	GVCID	None
MCF	Asynchronous or Synchronous	Transfer Frame	MCID	None
Insert	Periodic	IN_SDU	Physical Channel Name	None
COPs Management	N/A	N/A	GVCID	N/A

2.2.4.2 MAP Packet Service

The MAPP Service provides transfer of a sequence of variable-length, delimited, octet-aligned SDUs 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 a COP is used, then both Sequence-Controlled and Expedited services are provided for the MAPP Service. When the related managed parameter states that there is no COP in Effect, only Best Effort Delivery is provided for a MAP Channel.

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.

For a given service instance, multiple users, each identified with the GMAP ID of the MAP Channel and a PVN, can use this service on a MAP Channel. Packets containing MAP IDs from different users may be multiplexed together within one VC as long as these packets are multiplexed into the VC within the series of sequentially numbered USLP Frames that contain each complete MAP Packet. Therefore a new MAP ID cannot appear within a VCID until the packets from the previous MAP ID are completed.

2.2.4.3 Virtual Channel Packet Service

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

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the VCP Service. The user requests with a parameter of the service request primitive whether Type-A or Type-B should be applied for each Packet.

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

2.2.4.4 MAP Access Service

The MAPA Service provides transfer of a sequence of privately formatted SDUs of variable length, called MAPA_SDUs, across a space link. The length of the SDUs transferred by this service is not constrained by the length of the Data Field of the Transfer Frame. The MAPA Service delivers MAPA_SDUs whose length is not included within the data unit itself.

For a given service instance, multiple users, each identified with the GMAP ID of the MAP Channel, can use this service on a MAP Channel. MAPA_SDUs from different users may be multiplexed together within one VC as long as these SDUs are multiplexed into the VC within the series of sequentially numbered USLP Frames that contain each complete MAPA_SDU.

2.2.4.5 Virtual Channel Access Service

The VCA Service provides transfer of a sequence of privately formatted service data units of variable length across a space link.

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the VCA Service. The user requests with a parameter of the service request primitive whether Type-A or Type-B should be applied for each service data unit.

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.

2.2.4.6 MAP Octet Stream Service

The MAP Octet Stream Service provides transfer of a string of aligned octets, whose internal structure and boundaries are unknown to the service provider, across a space link. The service is unidirectional, asynchronous, and sequence-preserving. Based upon the QoS parameter selected by the user, either Sequence-Controlled or Expedited service can be provided. The octet stream is transferred using variable-length Transfer Frames exclusively (fixed-length USLP Frames are prohibited), since no idle data is inserted for variable-length USLP Frames, making for a more efficient transfer mechanism.

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

2.2.4.7 USLP Master Channel Operational Control Field Service

The USLP_MC_OCF Service provides transfer of fixed-length data units, each consisting of four octets, in the Operational Control Field (OCF) of Transfer Frames of a VC. The service is unidirectional and sequence-preserving. The transfer is signaled for inclusion within Transfer Frames of a VC 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 SDUs delivered to the receiving user.

For a given service instance, one or more users identified with the GVCID of the VC can use this service on an MC. SDUs from different users can be multiplexed together within one MC.

2.2.4.8 Virtual Channel Frame Service

The VCF Service provides transfer of a sequence of fixed- or variable-length USLP Transfer Frames of a VC, created by an independent protocol entity, across a space link. The service is unidirectional, either synchronous or asynchronous, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of SDUs delivered to the receiving user. The service does not make any distinction between Sequence-Controlled and Expedited service types applicable to SDUs supplied by the user. The user should perform necessary procedures to provide Sequence-Controlled and Expedited service types.

For a given service instance, only one user, identified with the GVCID of the VC, can use this service on a VC, and each VCF Service instance on a Physical Channel must utilize a unique GVCID value. SDUs from different users are not multiplexed together within one VC.

The VCF Service provides transfer of the independently created USLP Transfer Frames through a space link, possibly together with USLP Transfer Frames identified by other GVCID values 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 PDUs do not violate the operational integrity of the space link. Moreover, Transfer Frames provided by the VCF service user are partially formatted USLP Transfer Frames as defined in 3.2.7.

2.2.4.9 Master Channel Frame Service

The MCF Service provides transfer of a sequence of fixed- or variable-length USLP Transfer Frames of an MC, created by an independent protocol entity, across a space link. The service is unidirectional, either synchronous or asynchronous, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of SDUs delivered to a receiving user. The service does not make any distinction between Sequence-Controlled and Expedited service types applicable to SDUs supplied by the user. The user should perform necessary procedures to provide Sequence-Controlled and Expedited service types.

For a given service instance, one user, identified with the MCID of the Master Channel, can use this service on an MC and each MCF Service instance on a Physical Channel must utilize a unique MCID value. SDUs from different users are not multiplexed together within one MC.

The MCF Service provides transfer of the independently created USLP Transfer Frames through the space link, possibly together with USLP Transfer Frames identified by other MCID values 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 PDUs do not violate the operational integrity of the space link. Moreover, Transfer Frames provided by the MCF service user are partially formatted USLP Transfer Frames as defined in 3.2.7.

2.2.4.10 Insert Service

The Insert Service provides transfer of privately formatted, fixed-length, octet-aligned SDUs in fixed-length Transfer Frames on the Physical Channel across a space link in a mode that efficiently utilizes the space link transmission resources. The service is unidirectional, periodic, and sequence-preserving. The service does not guarantee completeness, but may signal gaps in the sequence of SDUs 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. SDUs from different users are not multiplexed together within one Physical Channel. The presence of the Insert Zone is signaled by the Physical Channel Managed Parameters.

2.2.4.11 COPs Management Service

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

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

2.2.5 RESTRICTIONS ON SERVICES

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

a) For fixed-length Transfer Frames only on one MAP Channel, the MAPA Service cannot exist simultaneously with the MAPP Service.

b) On one VC, the COPs Management Service shall not exist simultaneously with the VCF Service.

c) For fixed-length Transfer Frames only on one Virtual Channel, the VCA Service cannot exist simultaneously with the VCP Service.

d) The COPs Management Service shall not exist simultaneously with the MCF Service.

e) If the MCF Service exists on an MC, other data transfer services shall not exist simultaneously on the MC.

f) On one MC, only one USLP_MC_OCF Service can exist.

g) The MAP Octet Stream Service cannot exist when fixed-length Transfer Frames are used.

h) If the Virtual Channel Frame Service exists on a Virtual Channel, other data transfer services shall not exist simultaneously on that Virtual Channel.

i) If the Virtual Channel Access Service exists on a Virtual Channel, other data transfer services shall not exist simultaneously on that Virtual Channel.

j) If the Virtual Channel Packet Service exists on a Virtual Channel, other data transfer services shall not exist simultaneously on that Virtual Channel.

2.3 OVERVIEW OF FUNCTIONS

2.3.1 GENERAL FUNCTIONS

USLP transfers various SDUs, supplied by sending users, encapsulated in a sequence of PDUs using services of lower layers. The PDUs, known as USLP Transfer Frames, either have variable lengths and must be transferred over a Physical Channel asynchronously or have a fixed length and are normally transferred over a Physical Channel at a constant rate but may also be transferred over a Physical Channel asynchronously.

The protocol entity performs the following protocol functions:

a) generation and processing of protocol control information (i.e., headers and trailers) to perform data identification, loss detection, and error detection;

b) segmenting and blocking of SDUs presented at the SAP in order to transfer variable-length SDUs in either fixed- or variable-length PDUs that are shorter than those SDUs; and

c) multiplexing/demultiplexing and commutation/decommutation in order for various service users to share a single Physical Channel.

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

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

This protocol entity does not perform the following protocol functions:

a) connection establishment and release;

b) flow control;

c) retransmission of PDUs; or

d) management or configuration of the SDLS protocol.

2.3.2 INTERNAL ORGANIZATION OF PROTOCOL ENTITY

Figures 2-6 and 2-7 show the internal organization of the protocol entity of the sending and receiving ends, respectively. Data flow from top to bottom in figure 2-6 and from bottom to top in figure 2-7. 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.

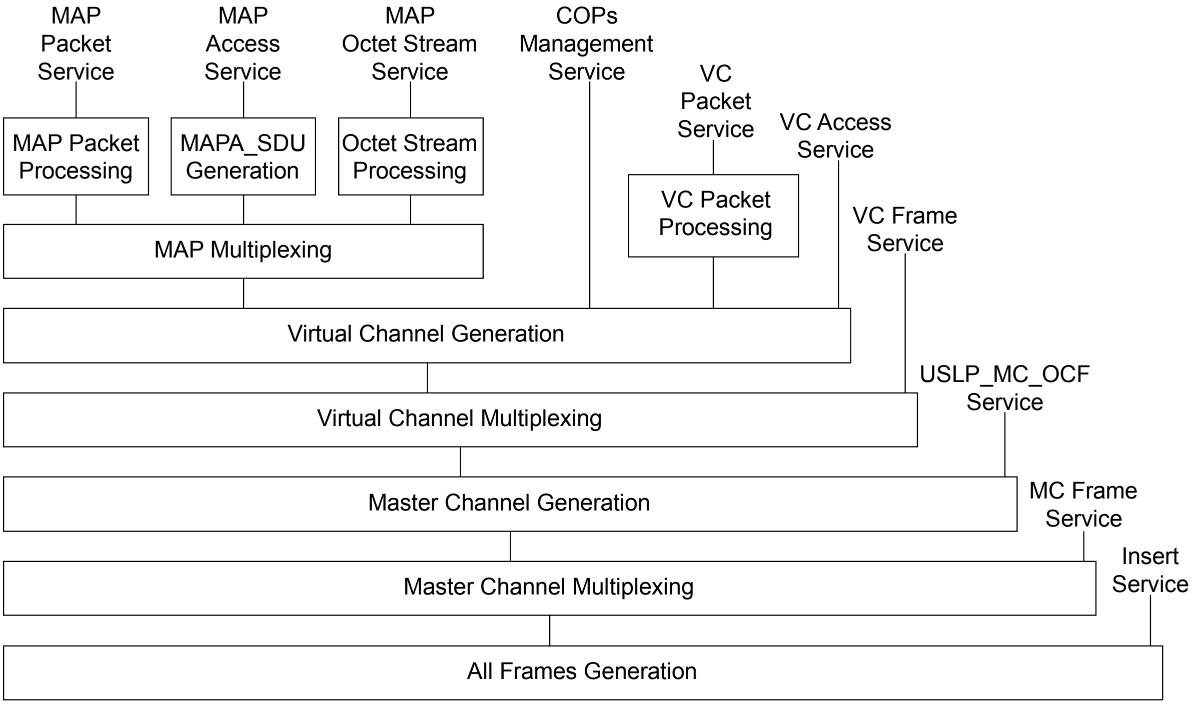


Figure 2-6: Internal Organization of Protocol Entity (Sending End)

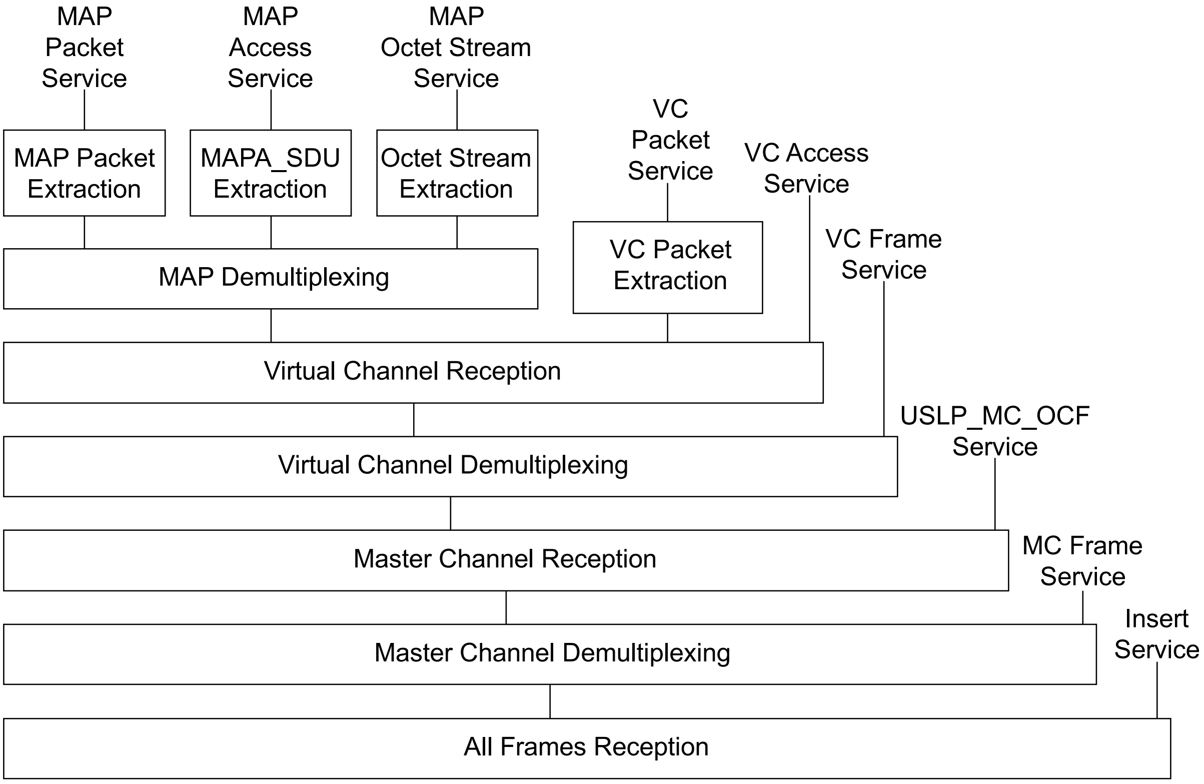


Figure 2-7: Internal Organization of Protocol Entity (Receiving End)

By extracting multiplexing/demultiplexing and commutation/decommutation functions from figures 2-6 and 2-7, the relationship among various data units can be shown as figure 2-8, which is known as the Channel Tree of USLP.

In figure 2-8, 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 PDU sharing the same identifier.

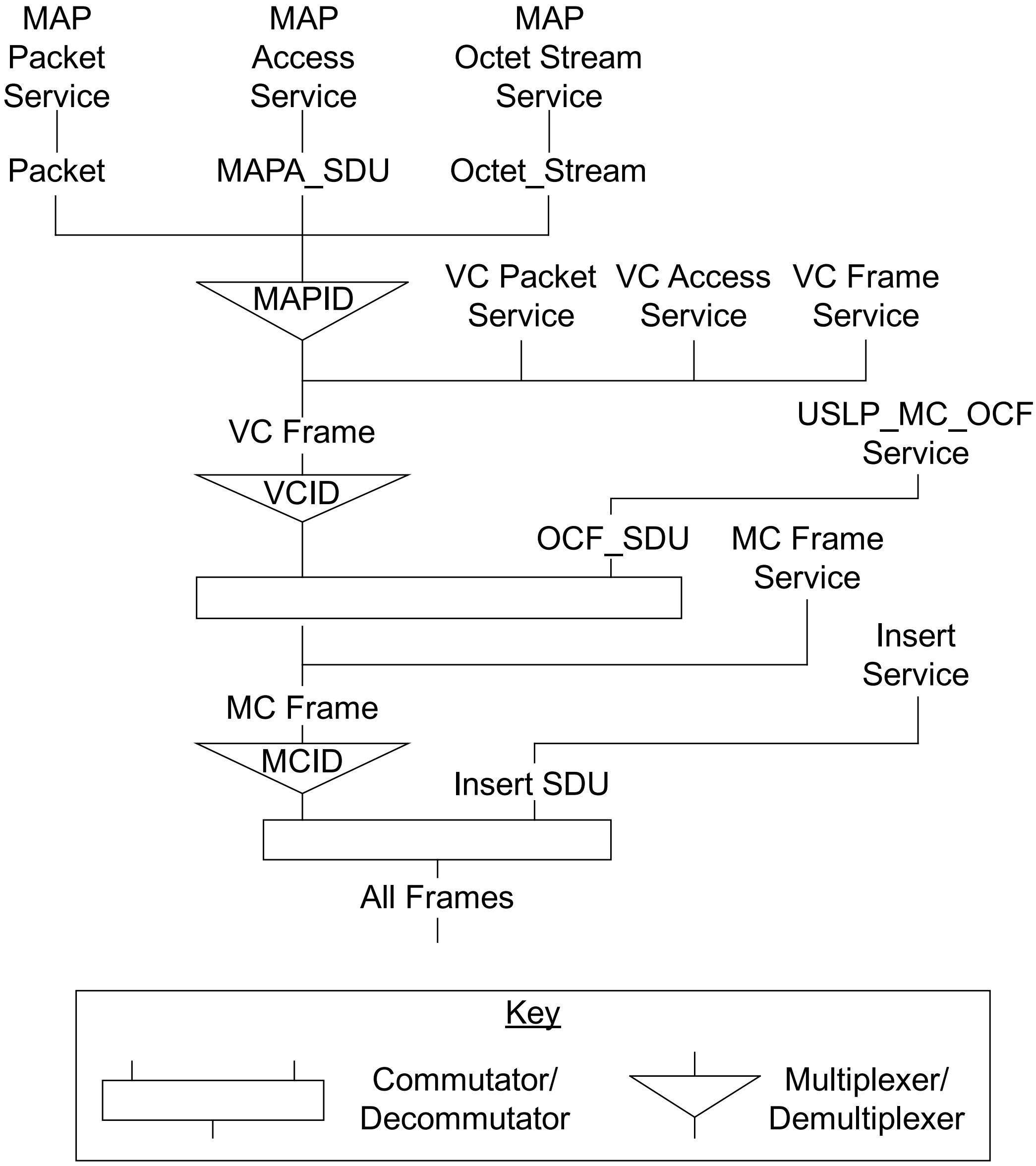


Figure 2-8: Unified Space Data Link Protocol Channel Tree

2.4 SERVICES ASSUMED FROM LOWER LAYERS

2.4.1 SERVICES ASSUMED FROM THE SYNCHRONIZATION AND CHANNEL CODING SUBLAYER

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

a) error control encoding and decoding functions (optional when the coding schemes defined in references [3] through [7] are used);

b) bit transition generation and removal functions (optional when the coding schemes defined in references [3] through [7] are used);

c) delimiting and synchronizing functions;

d) frame validation function (see below for limitation).

When any of the coding schemes defined in references [3], [4], and [5] are used, the Synchronization and Channel Coding Sublayer transfers fixed-length, delimited Transfer Frames as a contiguous stream of bits over a space link using the services of the underlying Physical Layer. Subsection 5.2 defines the managed parameter ‘Physical Channel Transfer Frame Type’, which in this case can only be set to the value ‘Fixed Length’.

When any of the coding schemes defined in references [6] and [7] are used, the Synchronization and Channel Coding Sublayer transfers variable-length, delimited Transfer Frames as an intermittent stream of bits over a space link using the services of the underlying Physical Layer. Subsection 5.2 defines the managed parameter ‘Physical Channel Transfer Frame Type’, which in this case is nominally set to the value ‘Variable Length’. However, as remarked in 2.3.1, fixed-length frames may also be transferred over a Physical Channel asynchronously using any of the coding schemes defined in references [6] and [7], setting the managed parameter ‘MC Transfer Frame Type’/‘VC Transfer Frame Type’ to the value ‘Fixed Length’.

Frame validation can be performed at the receiving end in the following ways depending upon the Channel Coding Sublayer specification selected:

a) If any of the coding schemes defined in references [3], [4], and [5] are used, the TM Synchronization and Channel Coding Sublayer can deliver fully validated Frames with or without the use of the optional FECF.

b) If any of the coding schemes defined in reference [6] are used, the TC Synchronization and Channel Coding Sublayer delivers a data stream corresponding to one or more decoded Transfer Frames, possibly incomplete or containing fill data, and USLP can deliver fully validated Frames utilizing the Frame Delimiting and Fill Data Removal Procedure (4.3.11.2) and the USLP Frame Validation Check Procedure (4.3.11.3).

c) If any of the coding schemes defined in reference [7] are used, the Proximity-1 Synchronization and Channel Coding Sublayer delivers fully validated USLP Frames through the use of the mandatory CRC added to the frame by Proximity-1 coding.

2.4.2 SYSTEMATIC RETRANSMISSIONS

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

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

USLP requests the systematic retransmissions in accordance with parameters set by management. For each VC, management sets the value to be used for the Repetitions parameter when requesting the transfer of USLP Frames carrying SDUs on the Sequence-Controlled Service (i.e., Type-AD frames in COP-1). For each VC, management sets a similar parameter for USLP Frames carrying COP Control Commands (i.e., Type-BC frames in COP-1). For a Physical Channel, management sets an upper limit for the value of the Repetitions parameter specified in reference [6].

NOTE – Repetitions are not permitted for USLP Frames carrying SDUs on the Expedited Service (i.e., for Type- BD frames in COP-1).

3 SERVICE DEFINITION

3.1 OVERVIEW

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

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

3.2 SOURCE DATA

3.2.1 SOURCE DATA OVERVIEW

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

The SDUs transferred by USLP shall be:

a) Packet (MAP Packet or VC Packet);

b) MAPA_SDU;

c) VCA_SDU;

d) Octet Stream Data;

e) OCF_SDU;

f) USLP Transfer Frame; and

g) Insert SDU (IN_SDU).

3.2.2 PACKET

3.2.2.1 Packets shall be transferred over a space link with the MAPP or VCP Services.

3.2.2.2 The packets transferred by these services must be self-delimiting and have a PVN registered by SANA (reference [8]).

3.2.2.3 The position and length of the Packet Length Field of the packets must be known to the service provider in order to extract packets from Transfer Frames at the receiving end.

3.2.2.4 Packets shall be contained either within a single USLP Frame or within multiple sequential USLP Frames of the same GVCID (for VCP service) or GMAP ID (for MAPP service) using the TFDZ Construction Rules in 4.1.4.2.2.

NOTES

1 Paragraph 3.2.2.4 applies when packets span Transfer Frames using TFDZ Construction Rule ‘000’ and also when blocking of packets is performed by the service provider using TFDZ Construction Rule ‘111’. (See 4.1.4.2.2 TFDZ Construction Rules.)

2 Packets are variable-length, delimited, octet-aligned data units.

3 Examples of packets are: CCSDS Space Packets, CCSDS Encapsulation Packets.

3.2.3 MAP ACCESS SERVICE DATA UNIT

3.2.3.1 MAPA_SDUs shall be transferred over a space link via the MAPA Service.

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

NOTE – MAPA_SDUs are variable-length, octet-aligned data units, the format of which is unknown to the service provider. Their length is provided to the SAP, and they are delimited within the TFDZ using the TFDZ construction rules.

3.2.4 VIRTUAL CHANNEL ACCESS SERVICE DATA UNIT

3.2.4.1 VCA_SDUs shall be transferred over a space link via the Virtual Channel Access Service.

3.2.4.2 A single VCA_SDU may be transmitted in the Data Zone of one or multiple Transfer Frames, and therefore the length of VCA_SDUs is not constrained by the length of the TFDZ.

NOTE – VCA_SDUs are variable-length, delimited, octet-aligned data units, the format of which is unknown to the service provider.

3.2.5 MAP OCTET STREAM DATA

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

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

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

3.2.6 OPERATIONAL CONTROL FIELD SERVICE DATA UNIT

OCF_SDUs shall be transferred over a space link with the USLP_MC_OCF Service.

NOTES

1 Although the transfer of OCF_SDUs is carried within the VCF, which by management can provide the OCF service, the creation of OCF_SDUs by the sending user may or may not be synchronized with a single VC. Such synchronization, if required for timing or other purposes, is a mission-design issue.

2 OCF_SDUs are fixed-length data units, each consisting of four octets, carried in the OCF, defined in 4.1.5, from a sending end to a receiving end.

3.2.7 USLP TRANSFER FRAME

USLP Transfer Frames transferred by the VCF and MCF Services shall be partially formatted USLP Transfer Frames, and the following restrictions apply:

a) if the USLP_MC_OCF Service exists on an MC, the OCF and the OCF Flag of the Transfer Frames submitted to the VCF Service on the same MC shall be empty;

b) if the Insert Service exists on the Physical Channel, the Transfer Frame Insert Zone of the USLP Transfer Frames submitted to the MCF or VCF Service shall be empty;

c) the FECF of the USLP Transfer Frames submitted to the MCF or VCF Service shall be empty, if it is present on the Physical Channel.

NOTE – The USLP Transfer Frame is the PDU of the USLP Space Data Link Protocol, but can also be used as the SDUs of the VCF and MCF Services. The USLP Transfer Frame format is defined in 4.1 and 6.3 of this Recommended Standard. When fixed-length USLP Frames are transmitted, the length of any Transfer Frame transferred on a Physical Channel must be the same, and is established by management.

3.2.8 INSERT SERVICE DATA UNIT

3.2.8.1 Insert Service Data Units (IN_SDUs) shall be periodic, octet-aligned data units of fixed length. Their presence within the USLP Frame shall be controlled by the Physical Channel Managed Parameter ‘Presence of Insert Zone’.

3.2.8.2 When present, each Transfer Frame on the Physical Channel shall contain one IN_SDU.

3.2.8.3 IN_SDU length shall be established by management and controlled by the Physical Channel Managed Parameter 'Insert Zone Length'.

3.2.8.4 The length of the Insert Zone may be of any constant value that is an integral number of octets, between 1 octet and the maximum length of the data-carrying space of the Transfer Frame.

3.3 MAP PACKET SERVICE

3.3.1 OVERVIEW OF PACKET SERVICE

The MAPP Service provides transfer of a sequence of variable-length, delimited, octet-aligned SDUs known as packets across a space link. The packets transferred by this service must have a PVN recognized by CCSDS. PVNs are registered in reference [8]. The service is unidirectional, asynchronous, and sequence-preserving. It does not guarantee completeness (because of potential gaps), nor does it signal gaps in the sequence of SDUs delivered to a receiving user. A sequence discontinuity in Transfer Frames does not guarantee that there will be a gap in packets for a specific Application Process ID (APID). (See reference [12].)

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

3.3.2 MAP PACKET SERVICE PARAMETERS

3.3.2.1 General

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

3.3.2.2 Packet

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

NOTE – The Packet parameter is the SDU transferred by the MAPP Service. Restrictions on the packets transferred by the MAPP Service are stated in 3.2.2.

3.3.2.3 GMAP ID

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

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

3.3.2.4 Packet Version Number

The PVN shall identify the CCSDS protocol entity of the upper layer that uses the MAPP Service.

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

3.3.2.5 SDU ID

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

3.3.2.6 QoS

At the sending end, the QoS parameter shall indicate whether the packet should be transferred with the Sequence-Controlled QoS or the Expedited QoS.

NOTE – When the related managed parameter states that there is no COP in Effect, this parameter is ignored in the MAPP.request, is omitted in the MAPP.indication, and no MAPP_Notify.indication is generated.

3.3.2.7 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with the transfer of a packet. The values taken by this parameter are defined in references [9] or [10].

3.3.2.8 Packet Quality Indicator

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

3.3.2.9 Verification Status Code

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

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

3.3.2.9.3 A non-zero value shall indicate that the SDLS protocol has detected an error; the values taken by this parameter are defined in reference [15].

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

3.3.3 MAP PACKET SERVICE PRIMITIVES

3.3.3.1 General

The service primitives associated with the MAPP service are:

a) MAPP.request;

b) MAPP_Notify.indication;

c) MAPP.indication.

3.3.3.2 MAPP.request

3.3.3.2.1 Function

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

3.3.3.2.2 Semantics

The MAPP.request primitive shall provide parameters as follows:

```
MAPP.request            (Packet,
                         GMAP ID,
                         PVN,
                         SDU ID,
                         QoS)
```

3.3.3.2.3 When Generated

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

3.3.3.2.4 Effect on Receipt

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

3.3.3.3 MAPP_Notify.indication

3.3.3.3.1 Function

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

3.3.3.3.2 Semantics

The MAPP_Notify.indication primitive shall provide parameters as follows:

```
MAPP_Notify.indication     (GMAP ID,
                            PVN,
                            SDU ID,
                            QoS,
                            Notification Type)
```

3.3.3.3.3 When Generated

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

3.3.3.3.4 Effect on Receipt

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

3.3.3.4 MAPP.indication

3.3.3.4.1 Function

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

3.3.3.4.2 Semantics

The MAPP.indication primitive shall provide parameters as follows:

MAPP.indication (Packet,
GMAP ID,
PVN,
QoS (optional),
Packet Quality Indicator (optional),
Verification Status Code (optional))

3.3.3.4.3 When Generated

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

3.3.3.4.4 Effect on Receipt

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

3.4 VIRTUAL CHANNEL PACKET SERVICE

3.4.1 OVERVIEW

The VCP Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link on a specified Virtual Channel. The Packets transferred by this service must be assigned a PVN by CCSDS. PVNs presently authorized by CCSDS have a PVN registered by SANA (reference [8]).

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the VCP Service. The user requests with a parameter of the service request primitive whether Type-A or Type-B should be applied for each Packet.

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

3.4.2 VCP SERVICE PARAMETERS

3.4.2.1 General

The parameters used by the VCP Service primitives shall conform to the specifications of the following subsections.

3.4.2.2 Packet

The Packet parameter shall contain a Packet for transfer on the Virtual Channel identified by GVCID.

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

3.4.2.3 GVCID

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

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

3.4.2.4 Packet Version Number

The Packet Version Number parameter shall contain the PVN of the Packet to be transferred.

NOTE – The PVN is part of the SAP address of the VCP Service and identifies the upper-layer protocol entity that uses the VCP Service.

3.4.2.5 SDU ID

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

3.4.2.6 Service Type

3.4.2.6.1 The Service Type parameter shall indicate whether the Packet should be transferred with the Sequence-Controlled Service type (Type-A) or the Expedited Service type (Type-B).

3.4.2.6.2 At the receiving end, the Service Type parameter is not used.

3.4.2.7 Notification Type

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

3.4.2.8 Packet Quality Indicator

3.4.2.8.1 The Packet Quality Indicator shall indicate whether the Packet delivered by the service provider to the service user at the receiving end is complete or not.

3.4.2.8.2 This parameter shall be used only when the service provider is required to deliver incomplete Packets to the service user at the receiving end.

3.4.2.9 Verification Status Code

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

3.4.2.9.2 The Verification Status Code parameter shall be used to notify the user at the receiving end of the Packet Service of a verification failure in a transfer frame addressed to the Virtual Channel.

3.4.2.9.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [15].

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

3.4.3 VCP SERVICE PRIMITIVES

3.4.3.1 General

The service primitives associated with the VCP service are:

a) VCP.request;

b) VCP_Notify.indication;

c) VCP.indication.

3.4.3.2 VCP.request

3.4.3.2.1 Function

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

3.4.3.2.2 Semantics

The VCP.request primitive shall provide parameters as follows:

```
VCP.request                (Packet,
                            GVCID,
                            Packet Version Number,
                            SDU_ID,
                            Service Type)
```

3.4.3.2.3 When Generated

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

3.4.3.2.4 Effect On Receipt

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

3.4.3.3 VCP_Notify.indication

3.4.3.3.1 Function

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

3.4.3.3.2 Semantics

The VCP_Notify.indication primitive shall provide parameters as follows:

```
VCP_Notify.indication        (GVCID,
                              Packet Version Number,
                              SDU ID,
                              Service Type,
                              Notification Type)
```

3.4.3.3.3 When Generated

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

3.4.3.3.4 Effect On Receipt

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

3.4.3.4 VCP.indication

3.4.3.4.1 Function

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

3.4.3.4.2 Semantics

The VCP.indication primitive shall provide parameters as follows:

```
VCP.indication          (Packet,
                         GVCID,
                         Packet Version Number,
                         Service Type (optional),
                         Packet Quality Indicator (optional),
                         Verification Status Code (optional))
```

3.4.3.4.3 When Generated

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

3.4.3.4.4 Effect On Receipt

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

3.5 MAP ACCESS SERVICE

3.5.1 OVERVIEW

The MAPA Service provides transfer of a sequence of privately formatted, octet-aligned, variable-length SDUs 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 SAP. The service is unidirectional, asynchronous, and sequence-preserving. The service does not guarantee completeness, but is capable of signaling gaps in the sequence of SDUs delivered to the receiving user, if the VC is limited to one and only one MAP ID assignment.

A user of this service is a protocol entity that sends or receives a MAPA_SDU within one or more USLP Frames identified with a USLP Protocol Identifier (UPID) and a GMAP ID. Different users (i.e., MAPA_SDUs identified with different UPIDs) may share a single MAP Channel but cannot share a single TFDZ. The service provider isolates each MAPA_SDU to a single USLP Frame or sequentially numbered USLP Frames using the TFDZ construction rules (see 4.1.4.2.2) on the same GMAP ID.

3.5.2 MAPA SERVICE PARAMETERS

3.5.2.1 General

The parameters used by the MAPA Service primitives shall conform to the specifications contained in subsections 3.5.2.2 through 3.5.2.8.

3.5.2.2 MAPA_SDU

The parameter MAPA_SDU shall be the SDU transferred by the MAPA Service.

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

3.5.2.3 GMAP ID

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

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

3.5.2.4 SDU ID

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

3.5.2.5 QoS

At the sending end, the QoS parameter shall indicate whether the MAPA_SDU should be transferred with the Sequence-Controlled QoS or the Expedited QoS.

3.5.2.6 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with the transfer of a MAPA_SDU. The values taken by this parameter are defined in references [9] or [10].

3.5.2.7 MAPA_SDU Loss Flag

3.5.2.7.1 The MAPA_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 MAPA_SDUs have been lost.

3.5.2.7.2 Gaps may be reported only if the VC is limited to one and only one MAP ID.

3.5.2.7.3 If implemented, the flag shall be derived by examining the VCF Count in the Transfer Frames.

3.5.2.8 Verification Status Code

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

3.5.2.8.2 The parameter shall be used to notify the user at the receiving end of the MAPA Service of a verification failure in a Transfer Frame addressed to the VC.

3.5.2.8.3 A non-zero value shall indicate that the SDLS protocol has detected an error; the values taken by this parameter are defined in reference [15].

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

3.5.3 MAPA SERVICE PRIMITIVES

3.5.3.1 General

The service primitives associated with the MAPA service are

a) MAPA.request;

b) MAPA_Notify.indication; and

c) MAPA.indication.

3.5.3.2 MAPA.request

3.5.3.2.1 Function

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

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

3.5.3.2.2 Semantics

The MAPA.request primitive shall provide parameters as follows:

```
MAPA.request             (MAPA_SDU,
                         GMAP ID,
                         SDU ID,
                         QoS)
```

3.5.3.2.3 When Generated

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

3.5.3.2.4 Effect on Receipt

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

3.5.3.3 MAPA_Notify.indication

3.5.3.3.1 Function

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

3.5.3.3.2 Semantics

The MAPA_Notify.indication primitive shall provide parameters as follows:

```
MAPA_Notify.indication      (GMAP ID,
                            SDU ID, QoS,
                            Notification Type)
```

3.5.3.3.3 When Generated

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

3.5.3.3.4 Effect on Receipt

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

3.5.3.4 MAPA.indication

3.5.3.4.1 Function

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

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

3.5.3.4.2 Semantics

The MAPA.indication primitive shall provide parameters as follows:

```
MAPA.indication          (MAPA_SDU,
                         GMAP ID,
                         QoS [optional], MAPA_SDU Loss Flag [optional],
                         Verification Status Code [optional])
```

3.5.3.4.3 When Generated

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

3.5.3.4.4 Effect on Receipt

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

3.6 VIRTUAL CHANNEL ACCESS SERVICE

3.6.1 OVERVIEW

The VCA Service provides transfer of a sequence of privately formatted service data units of variable length across a space link.

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the VCA Service. The user requests with a parameter of the service request primitive whether Type-A or Type-B should be applied for each service data unit.

3.6.2 GENERAL

3.6.2.1 For fixed-length Transfer Frames only on one Virtual Channel, the VCA Service shall not exist simultaneously with the VCP Service.

3.6.2.2 Service data units from different users shall not be multiplexed together within one Virtual Channel.

3.6.3 VCA SERVICE PARAMETERS

3.6.3.1 General

The parameters used by the VCA Service primitives shall conform to the specifications of the following subsections.

3.6.3.2 VCA_SDU

The VCA_SDU parameter shall contain a VCA_SDU to be transferred on the Virtual Channel identified by GVCID.

NOTE – The VCA_SDU is the service data unit transferred by the VCA Service. Restrictions on the VCA_SDUs transferred by the VCA Service are stated in 3.2.4.

3.6.3.3 GVCID

The GVCID parameter shall contain the GVCID of the Virtual Channel through which the VCA_SDU is to be transferred.

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

3.6.3.4 SDU ID

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

3.6.3.5 Service Type

3.6.3.5.1 The Service Type parameter shall be used to indicate whether the VCA_SDU should be transferred with the Sequence-Controlled Service type (Type-A) or the Expedited Service type (Type-B).

3.6.3.5.2 At the receiving end, the Service Type parameter is not used.

3.6.3.6 Notification Type

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

3.6.3.7 Verification Status Code

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

3.6.3.7.2 The Verification Status Code 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.

3.6.3.7.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [15].

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

3.6.4 VCA SERVICE PRIMITIVES

3.6.4.1 General

The service primitives associated with this service are:

a) VCA.request;

b) VCA_Notify.indication;

c) VCA.indication.

3.6.4.2 VCA.request

3.6.4.2.1 Function

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

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

3.6.4.2.2 Semantics

The VCA.request primitive shall provide parameters as follows:

```
VCA.request              (VCA_SDU,
                          GVCID,
                          SDU ID,
                          Service Type)
```

3.6.4.2.3 When Generated

The VCA service user shall generate a VCA.request primitive when a VCA_SDU is ready for transfer.

3.6.4.2.4 Effect On Receipt

Receipt of the VCA.request primitive shall cause the service provider to transfer the VCA_SDU.

3.6.4.3 VCA_Notify.indication

3.6.4.3.1 Function

At the sending end, the service provider shall pass a VCA_Notify.indication primitive to the VCA Service user to notify the user of an event associated with the transfer of a VCA_SDU.

3.6.4.3.2 Semantics

The VCA.indication primitive shall provide parameters as follows:

VCA_Notify.indication (GVCID,
SDU ID,
Service Type,
Notification Type)

3.6.4.3.3 When Generated

The service provider shall generate a VCA_Notify.indication primitive in response to an event associated with the transfer of a VCA_SDU.

3.6.4.3.4 Effect On Receipt

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

3.6.4.4 VCA.indication

3.6.4.4.1 Function

At the receiving end, the service provider shall pass a VCA.indication primitive to the VCA Service user to deliver a VCA_SDU.

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

3.6.4.4.2 Semantics

The VCA.indication primitive shall provide parameters as follows:

```
VCA.indication          (VCA_SDU,
                        GVCID,
                        Service Type (optional),
                        Verification Status Code (optional))
```

3.6.4.4.3 When Generated

The service provider shall generate a VCA.indication primitive when a VCA_SDU is ready for delivery.

3.6.4.4.4 Effect On Receipt

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

3.7 MAP OCTET STREAM SERVICE

3.7.1 OVERVIEW OF MAP OCTET STREAM SERVICE

The MAP Octet Stream Service provides transfer of a serial string of octets, whose internal structure and boundaries are unknown to the service provider, across a space link. The service is unidirectional, asynchronous, and sequence-preserving. Based upon the QoS parameter selected by the user, either Sequence-Controlled or Expedited service can be provided. The octet stream is transferred using variable-length Transfer Frames exclusively (fixed-length USLP Frames are prohibited), since no idle data is inserted for variable-length USLP Frames, making for a more efficient transfer mechanism.

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

3.7.2 MAP OCTET STREAM SERVICE PARAMETERS

3.7.2.1 General

The parameters used by the MAP Octet Stream Service primitives shall conform to the specifications contained in subsections 3.7.2.2 through 3.7.2.7.

3.7.2.2 Octet Stream Data

The parameter Octet Stream Data shall be the SDU transferred by the MAP Octet Stream Service.

NOTE – Restrictions on the Octet Stream Data transferred by the MAP Octet Stream Service are stated in 3.2.4.

3.7.2.3 GMAP ID

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

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

3.7.2.4 SDU ID

The SDU ID parameter shall contain a user-supplied sequence number to be used to identify the associated MAP Octet Stream in subsequent MAPA_Notify.indication primitives.

3.7.2.5 QoS

At the sending end, the QoS parameter shall indicate whether the MAP Octet Stream should be transferred with the Sequence-Controlled QoS or the Expedited QoS.

3.7.2.6 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 MAP 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 VCF Count in the Transfer Frames.

NOTE – As the contents 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 VC, and not necessarily a disruption of the Octet Stream Data itself.

3.7.2.7 Verification Status Code

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

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

3.7.2.7.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [15].

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

3.7.3 MAP OCTET STREAM SERVICE PRIMITIVES

3.7.3.1 General

The service primitives associated with the MAP Octet Stream Service are

a) OCTET_STREAM.request;

b) OCTET_STREAM.indication; and

c) OCTET_STREAM_Notify.indication.

3.7.3.2 OCTET_STREAM.request

3.7.3.2.1 Function

At the sending end, the MAP 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 VC.

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

3.7.3.2.2 Semantics

The OCTET_STREAM.request primitive shall provide parameters as follows:

OCTET_STREAM.request (Octet Stream Data,
GMAP ID, SDU ID, QoS)

3.7.3.2.3 When Generated

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

3.7.3.2.4 Effect on Receipt

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

NOTES

1 The OCTET_STREAM.request primitive is used to transfer portions of Octet Stream Data across the space link on the specified VC.

2 Since the service interface specification is an abstract specification, the implementation of the Octet Stream Data parameter is not constrained; that is, it can be a continuous octet stream, a delimited octet stream, or individual bits.

3.7.3.3 OCTET_STREAM.indication

3.7.3.3.1 Function

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

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

3.7.3.3.2 Semantics

The OCTET_STREAM.indication primitive shall provide parameters as follows:

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

3.7.3.3.3 When Generated

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

3.7.3.3.4 Effect on Receipt

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

NOTES

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

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

3.7.3.4 OCTET_STREAM_Notify.indication

3.7.3.4.1 Function

At the sending end, the service provider shall pass an OCTET_STREAM_Notify.indication primitive to the MAP Octet Stream Service user to notify the user of an event associated with the transfer of an Octet Stream.

3.7.3.4.2 Semantics

The OCTET_STREAM_Notify.indication primitive shall provide parameters as follows:

OCTET_STREAM_Notify.indication (GMAP ID,
SDU ID, QoS,
Notification Type)

3.7.3.4.3 When Generated

The sending-end service provider shall generate an OCTET_STREAM_Notify.indication in response to an event associated with the transfer of a portion of an Octet Stream.

3.7.3.4.4 Effect on Receipt

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

3.8 USLP MASTER CHANNEL OPERATIONAL CONTROL FIELD SERVICE

3.8.1 OVERVIEW

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

Only one user on the sending end, identified with the MCID of the Master Channel, can use the USLP_MC_OCF Service on a given MC. Only one user on the receiving end, identified with the MCID of the Master Channel, can use the USLP_MC_OCF Service on a given MC. Only one USLP_MC_OCF service can exist within an MC, and that service may be utilized by multiple VCs that are designated by either the MC Managed Parameter 'Inclusion of OCF Required' for fixed length frames or the VC Managed Parameter 'Inclusion of OCF Allowed' for variable length frames to support the service.

NOTE – For the USLP_MC_OCF Service, the sending end is the one inserting the OCF (e.g., the end running the FARM) while the receiving end is the one extracting the OCF (e.g., the end running the FOP).

When OCFs contain a Frame Security Report (FSR) or a CLCW/PLCW, the USLP_MC_OCF User at the sending end (i.e., where FARM and SDLS secured link Recipient is located) supplies the OCFs to be included in USLP Transfer Frames. This Recommended Standard does not specify how the USLP_MC_OCF User receives CLCWs/PLCWs or FSRs and how they are multiplexed before supplying them to the USLP Service Provider.

When OCFs contain an FSR or a CLCW/PLCW, the USLP_MC_OCF User at the receiving end (i.e., where SDLS secured link Initiator and FOP is located) receives from the MC Demultiplexing Function the OCFs included in USLP Transfer Frames from the MC Demultiplexing Function. This Recommended Standard does not specify how the USLP_MC_OCF user demultiplexes the received OCFs, the interfaces and methods by which FSRs or CLCWs/PLCWs are delivered to the SDLS function at the Initiator end, or the pertinent FOP.

3.8.2 USLP_MC_OCF SERVICE PARAMETERS

3.8.2.1 General

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

3.8.2.2 OCF_SDU

The parameter OCF_SDU shall be the SDU transferred by the USLP_MC_OCF Service in the OCF of Transfer Frames of a VC.

NOTE – Restrictions on the OCF_SDU transferred by the USLP_MC_OCF Service are stated in 3.2.6.

3.8.2.3 GVCID

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

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

3.8.2.4 OCF_SDU Loss Flag

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

3.8.2.4.2 If implemented, the flag shall be derived by examining the VCF Count in the Transfer Frames.

3.8.3 USLP_MC_OCF SERVICE PRIMITIVES

3.8.3.1 General

The service primitives associated with this service are

a) USLP_MC_OCF.request; and

b) USLP_MC_OCF.indication.

3.8.3.2 USLP_MC_OCF.request

3.8.3.2.1 Function

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

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

3.8.3.2.2 Semantics

The USLP_MC_OCF.request primitive shall provide parameters as follows:

```
USLP_MC_OCF.request      (OCF_SDU,
                          GVCID)
```

3.8.3.2.3 When Generated

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

3.8.3.2.4 Effect on Receipt

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

NOTE – The USLP_MC_OCF.request primitive is used to transfer OCF_SDUs across the space link on the specified VC.

3.8.3.3 USLP_MC_OCF.indication

3.8.3.3.1 Function

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

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

3.8.3.3.2 Semantics

The USLP_MC_OCF.indication primitive shall provide parameters as follows:

```
USLP_MC_OCF.indication  (OCF_SDU,
                         GVCID,
                         OCF_SDU Loss Flag [optional])
```

3.8.3.3.3 When Generated

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

3.8.3.3.4 Effect on Receipt

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

NOTE – The USLP_MC_OCF.indication primitive is used to deliver OCF_SDUs to the USLP_MC_OCF Service user identified by the GVCID.

3.9 VIRTUAL CHANNEL FRAME SERVICE

3.9.1 OVERVIEW OF VCF SERVICE

The VCF Service provides transfer of a sequence of fixed- or variable-length USLP Transfer Frames of a VC, created by an independent protocol entity, across a space link. The service is unidirectional, asynchronous, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of SDUs delivered to the receiving user.

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

3.9.2 VCF SERVICE PARAMETERS

3.9.2.1 General

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

3.9.2.2 Frame

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

NOTES

1 The Frame parameter is the SDU transferred by the VCF Service.

2 The format of the GVCID parameter is defined in 4.1.

3 Restrictions on the USLP Transfer Frames transferred by the VCF Service are referenced in 3.2.7.

3.9.2.3 GVCID

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

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

3.9.2.4 Frame Loss Flag

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

3.9.2.4.2 If implemented, the flag shall be derived by examining the VCF Count in the Transfer Frames.

3.9.3 VCF SERVICE PRIMITIVES

3.9.3.1 General

The service primitives associated with this service are

a) VCF.request; and

b) VCF.indication.

3.9.3.2 VCF.request

3.9.3.2.1 Function

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

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

3.9.3.2.2 Semantics

The VCF.request primitive shall provide parameters as follows:

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

3.9.3.2.3 When Generated

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

3.9.3.2.4 Effect on Receipt

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

NOTE – The VCF.request primitive is used to transfer Transfer Frames of a VC across the space link.

3.9.3.3 VCF.indication

3.9.3.3.1 Function

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

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

3.9.3.3.2 Semantics

The VCF.indication primitive shall provide parameters as follows:

```
VCF.indication          (Frame,
                         GVCID,
                         Frame Loss Flag [optional])
```

3.9.3.3.3 When Generated

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

3.9.3.3.4 Effect on Receipt

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

NOTE – The VCF.indication primitive is used to deliver Transfer Frames of a VC to the VCF Service user identified by the GVCID.

3.10 MASTER CHANNEL FRAME SERVICE

3.10.1 OVERVIEW OF MCF SERVICE

The MCF Service provides transfer of a sequence of fixed- or variable-length USLP Transfer Frames of an MC, created by an independent protocol entity, across a space link. The service is unidirectional, asynchronous, and sequence-preserving. The service does not guarantee completeness, but it may signal gaps in the sequence of SDUs delivered to a receiving user.

Only one user can use this service on an MC, and the user is identified with the MCID of the MC. SDUs from different users are not multiplexed together within one MC.

3.10.2 MCF SERVICE PARAMETERS

3.10.2.1 General

The parameters used by the MCF Service primitives shall conform to the specifications contained in subsections 3.10.2.2 through 3.10.2.4.

3.10.2.2 Frame

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

NOTES

1 The Frame parameter is the SDU transferred by the VCF Service.

2 The format of the Frame parameter is defined in 4.1.

3 Restrictions on the USLP Transfer Frames transferred by the MCF Service are referenced in 3.2.7.

3.10.2.3 MCID

The MCID parameter shall contain the MCID of the MC on which the USLP Frame is to be transferred.

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

3.10.2.4 Frame Loss Flag

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

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

3.10.3 MCF SERVICE PRIMITIVES

3.10.3.1 General

The service primitives associated with this service are

a) MCF.request; and

b) MCF.indication.

3.10.3.2 MCF.request

3.10.3.2.1 Function

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

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

3.10.3.2.2 Semantics

The MCF.request primitive shall provide parameters as follows:

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

3.10.3.2.3 When Generated

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

3.10.3.2.4 Effect on Receipt

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

NOTE – The MCF.request primitive is used to transfer Transfer Frames of an MC across the space link.

3.10.3.3 MCF.indication

3.10.3.3.1 Function

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

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

3.10.3.3.2 Semantics

The MCF.indication primitive shall provide parameters as follows:

```
MCF.indication           (Frame,
                          MCID,
                          Frame Loss Flag [optional])
```

3.10.3.3.3 When Generated

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

3.10.3.3.4 Effect on Receipt

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

NOTE – The MCF.indication primitive is used to deliver Transfer Frames of an MC to the VCF Service user identified by the MCID.

3.11 INSERT SERVICE

3.11.1 OVERVIEW OF INSERT SERVICE

The Insert Service provides transfer of privately formatted, fixed-length, octet-aligned SDUs in fixed-length Transfer Frames across a space link in a mode that 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 SDUs delivered to a receiving user. The presence and length of the Insert Zone is controlled by the Physical Channel 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. SDUs from different users are not multiplexed together within one Physical Channel.

The Insert Service is provided only if the managed parameter ‘Physical Channel Transfer Frame Type’ is ‘Fixed Length’.

3.11.2 INSERT SERVICE PARAMETERS

3.11.2.1 General

The parameters used by the Insert Service primitives shall conform to the specifications contained in subsections 3.11.2.2 through 3.11.2.4.

3.11.2.2 IN_SDU

The IN_SDU parameter shall be the SDU transferred by the Insert Service.

NOTE – Restrictions on the IN_SDUs transferred by the Insert Service are stated in 3.2.8.

3.11.2.3 Physical Channel Name

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

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

3.11.2.4 IN_SDU Loss Flag

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

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

3.11.3 INSERT SERVICE PRIMITIVES

3.11.3.1 General

The service primitives associated with this service are

a) INSERT.request; and

b) INSERT.indication.

3.11.3.2 INSERT.request

3.11.3.2.1 Function

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

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

3.11.3.2.2 Semantics

The INSERT.request primitive shall provide parameters as follows:

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

3.11.3.2.3 When Generated

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

3.11.3.2.4 Effect on Receipt

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

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

3.11.3.3 INSERT.indication

3.11.3.3.1 Function

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

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

3.11.3.3.2 Semantics

The INSERT.indication primitive shall provide parameters as follows:

```
INSERT.indication          (IN_SDU,
                            Physical Channel Name,
                            IN_SDU Loss Flag [optional])
```

3.11.3.3.3 When Generated

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

3.11.3.3.4 Effect on Receipt

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

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

3.12 COPs MANAGEMENT SERVICE

3.12.1 OVERVIEW

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

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

3.12.2 COPs MANAGEMENT SERVICE PARAMETERS

3.12.2.1 General

The parameters used by the COPs Management Service primitives shall conform to the specifications of 3.12.2.2 through 3.12.2.7.

3.12.2.2 GVCID/Port ID

3.12.2.2.1 The GVCID parameter shall contain the GVCID of the VC for which the COP-1 is managed.

NOTE – The GVCID consists of an MCID and a VCID and is the SAP address of the COP-1 Management Service.

3.12.2.2.2 The Port ID parameter shall contain the Port ID for which the COP-P is managed (see reference [10]).

3.12.2.3 Directive ID

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

3.12.2.4 Directive Type

The Directive Type parameter shall contain the type of Directive.

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

3.12.2.5 Directive Qualifier

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

3.12.2.6 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with a Directive. The values taken by this parameter are defined in reference [9] for COP-1 or reference [10] for COP-P.

3.12.2.7 Notification Qualifier

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

3.12.3 COPs MANAGEMENT SERVICE PRIMITIVES

3.12.3.1 General

The service primitives associated with the COPs Management Service are

a) Directive.request;

b) Directive_Notify.indication; and

c) Async_Notify.indication.

3.12.3.2 Directive.request

3.12.3.2.1 Function

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

3.12.3.2.2 Semantics

The Directive.request primitive shall provide parameters as follows:

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

3.12.3.2.3 When Generated

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

3.12.3.2.4 Effect on Receipt

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

3.12.3.3 Directive_Notify.indication

3.12.3.3.1 Function

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

3.12.3.3.2 Semantics

The Directive_Notify.indication primitive shall provide parameters as follows:

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

3.12.3.3.3 When Generated

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

3.12.3.3.4 Effect on Receipt

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

3.12.3.4 Async_Notify.indication

3.12.3.4.1 Function

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

3.12.3.4.2 Semantics

The Async_Notify.indication primitive shall provide parameters as follows:

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

3.12.3.4.3 When Generated

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

3.12.3.4.4 Effect on Receipt

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

4 PROTOCOL SPECIFICATION WITHOUT SDLS OPTION

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

4.1 PROTOCOL DATA UNIT

4.1.1 USLP TRANSFER FRAME

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

a) Transfer Frame Primary Header (4 to 14 octets, mandatory);

b) Transfer Frame Insert Zone (integral number of octets, optional);

c) Transfer Frame Data Field (integral number of octets, mandatory);

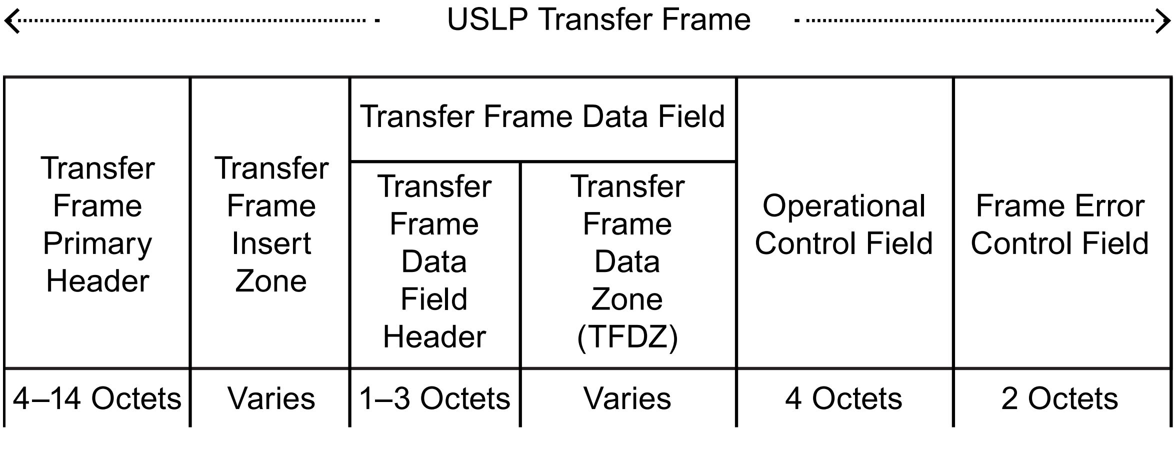
d) Operational Control Field (4 octets, optional); and

e) Frame Error Control Field (2 octets, optional).

NOTES

1 The major fields of the USLP Transfer Frame are shown in figure 4-1. While the Transfer Frame Primary Header and the TFDF are always present, conditions for the presence of the Insert Zone, OCF, and FECF are further discussed in section 4 below.

2 In this document, the unqualified term ‘Transfer Frame’ refers to the ‘non-truncated Transfer Frame’.



Transfer Frame Primary Header	Transfer Frame Insert Zone	Transfer Frame Data Field		Operational Control Field	Frame Error Control Field
		Transfer Frame Data Field Header	Transfer Frame Data Zone (TFDZ)		
4–14 Octets	Varies	1–3 Octets	Varies	4 Octets	2 Octets

Figure 4-1: USLP Transfer Frame Structural Components

4.1.2 TRANSFER FRAME PRIMARY HEADER

4.1.2.1 General

4.1.2.1.1 The Transfer Frame shall contain either a non-truncated (see figure 4-1) or a truncated (see figure D-1) Transfer Frame Primary Header.

NOTE – In this document, the unqualified term 'Transfer Frame Primary Header' refers to the 'non-truncated Transfer Frame Primary Header'.

4.1.2.1.2 The (non-truncated) Transfer Frame Primary Header shall consist of 13 fields, positioned contiguously, in the following sequence:

a) TFVN (4 bits);

b) SCID (16 bits);

c) Source or Destination Identifier (1 bit);

d) VCID (6 bits);

e) MAP ID (4 bits);

f) End of Frame Primary Header Flag (1 bit);

g) Frame Length (16 bits);

h) Bypass/Sequence Control Flag (1 bit);

i) Protocol Control Command Flag (1 bit);

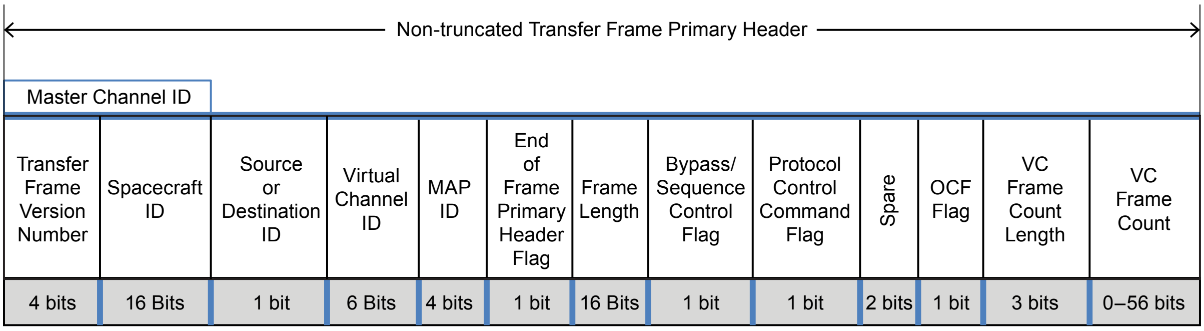
j) Reserve Spares (2 bits);

k) OCF Flag (1 bit);

l) VCF Count Length (3 bits);

m) VCF Count (0 to 56 bits).

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



Transfer Frame Version Number	Spacecraft ID	Source or Destination ID	Virtual Channel ID	MAP ID	End of Frame Primary Header Flag	Frame Length	Bypass/ Sequence Control Flag	Protocol Control Command Flag	Spare	OCF Flag	VC Frame Count Length	VC Frame Count
4 bits	16 Bits	1 bit	6 Bits	4 bits	1 bit	16 Bits	1 bit	1 bit	2 bits	1 bit	3 bits	0–56 bits

Figure 4-2: Non-truncated Transfer Frame Primary Header

4.1.2.2 Master Channel Identifier

4.1.2.2.1 General

4.1.2.2.1.1 Bits 0–19 of the Transfer Frame Primary Header shall contain the MCID.

4.1.2.2.1.2 The MCID shall consist of

a) TFVN (4 bits, mandatory); and

b) SCID (16 bits, mandatory).

4.1.2.2.2 Transfer Frame Version Number

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

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

NOTE – This Recommended Standard defines the USLP Version-4 Transfer Frame.

4.1.2.2.3 Spacecraft Identifier

Bits 4–19 of the Transfer Frame Primary Header shall contain the SCID.

NOTE – SANA assigns SCIDs according to the procedures in reference [11].

4.1.2.3 Source-or-Destination Identifier

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

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

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

a) ‘0’ = SCID refers to the *source* of the Transfer Frame; and

b) ‘1’ = SCID refers to the *destination* of the Transfer Frame.

NOTE – Although not limited to proximity environments, a discussion of the assignment of source vs. destination SCID is provided in reference [F14].

4.1.2.4 Virtual Channel Identifier

4.1.2.4.1 Bits 21–26 of the Transfer Frame Primary Header shall contain the VCID.

4.1.2.4.2 The VCID shall be used to identify the VC.

NOTE – There are no restrictions on the selection of VCIDs 0–62; in particular, VCs are not required to be numbered consecutively. VCID 63 is reserved for Only Idle Data (OID) Transfer Frames (see 4.1.4.1.5 through 4.1.4.1.7).

4.1.2.5 Multiplexer Access Point Identifier

4.1.2.5.1 Bits 27–30 of the Transfer Frame Primary Header shall contain the MAP ID.

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

4.1.2.5.3 When multiple MAPs are in use on a VC,

a) a given SDU shall be assigned to only one MAP at a time; and

b) the entire SDU must be completed before a new SDU associated with any other MAP is transferred on that VC.

NOTES

1 The MAP ID provides the identification from 1 to up to 16 possible MAP Channels within a VC.

2 There are no restrictions on the selection of MAPs, except for OID Frames (see 4.1.4.1.8). In particular, MAPs are not required to be numbered consecutively.

3 MAPs identify SAPs for the VC and may be designated to receive Sequence-Controlled or Expedited SDUs.

4.1.2.6 End of Frame Primary Header Flag

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

4.1.2.6.2 The End of Frame Primary Header Flag shall signal the use of the truncated Transfer Frame Primary Header (figure D-2) in lieu of the complete header. The End of Frame Primary Header Flag is interpreted as follows:

a) ‘0’ = Transfer Frame Primary Header is not truncated and consists of the 13 fields defined in 4.1.2.1.

b) ‘1’ = The Transfer Frame Primary Header is truncated and is defined as containing only the first contiguous six fields of the Transfer Frame Primary Header. (See annex D, Truncated Transfer Frame Primary Header.)

NOTE – A Transfer Frame containing a truncated Transfer Frame Primary Header is called a truncated Transfer Frame. This frame is envisioned for transmitting short-length forward-link or direct-from-Earth telecommands. (See annex D for frame format and requirements.)

4.1.2.7 Frame Length

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

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

4.1.2.7.3 The count shall be measured from the first octet of the Transfer Frame Primary Header to the last octet of the FECF (if present), to the last octet of the OCF (if present), or to the last octet of the TFDF (if both the FECF and the OCF are omitted).

NOTES

1 If the managed parameter ‘Physical Channel Transfer Frame Type’ is ‘Fixed Length’, then the values allowed for the Transfer Frame Length can be constrained by the specifications contained in references [3], [4], and [5].

2 The Maximum Transfer Frame Length of a variable-length Frame or an uncoded fixed-length Frame is constrained only by the size of the Frame Length field.

3 The length count *C* is expressed as:

C = (Total Number of Octets in the Transfer Frame) − 1

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

5 In general, Transfer Frame Type can be either 'Fixed Length' or 'Variable Length' for a given Physical, Master, or VC. If the Transfer Frame Type is 'Fixed Length' for the Physical Channel, then it is fixed for both the subordinate MC and VC.

4.1.2.8 Bypass/Sequence Control Flag and Protocol Control Command Flag

4.1.2.8.1 Bypass/Sequence Control Flag

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

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

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

b) Expedited QoS: setting the Bypass/Sequence Control Flag to value '1' shall specify that this Transfer Frame is an Expedited Transfer Frame, and the Frame Acceptance Checks of the FARM by the receiving end shall be bypassed.

NOTES

1 This field is equivalent to the QoS Indicator in reference [10].

2 The 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 an Expedited mode (for Expedited Transfer Frames), according to the setting of the Bypass/Sequence Control Flag.

3 For COP-1 over Telecommand Channel Coding in reference [6], all Transfer Frames received by the receiving end first undergo a basic set of USLP Frame and TC Channel Coding Validation Checks, which are applied regardless of the setting of the Bypass/Sequence Control Flag (see 4.3.11.3).

4 For COP-P over Proximity-1 Channel Coding in reference [7], Frame Validation is carried out by the Proximity-1 C&S Sublayer, since the FECF is defined outside of the Version-3 Transfer Frame.

4.1.2.8.2 Protocol Control Command Flag

4.1.2.8.2.1 Bit 49 of the Transfer Frame Primary Header shall contain the Protocol Control Command Flag.

4.1.2.8.2.2 The single-bit Protocol Control Command Flag shall be used to specify whether the TFDF is conveying Protocol Control Commands or user data:

a) setting the Protocol Control Command Flag to value ‘0’ shall indicate that the TFDF contains user data;

b) setting the Protocol Control Command Flag to value ‘1’ shall indicate that the TFDF contains protocol control information.

NOTE – This field is equivalent to the PDU Type in reference [10].

4.1.2.8.3 Combined States of the Bypass Flag and Protocol Control Command Flag

The combined states of the Bypass Flag and Protocol Control Command Flag shall be interpreted by the receiving end as shown in table 4-1.

Table 4-1: Interpretation of the Bypass/Sequence Control and Protocol Control Command Flags

Bypass/ Sequence-Control Flag	Protocol Control Command Flag	Interpretation
0	0	TFDF carries user data, subject to Frame Acceptance Checks under control of the FARM. These USLP Frames are used for Sequence-Controlled Service of COP-1 or COP-P. Type-AD Transfer Frames are used for COP-1 and Sequence Controlled U-Frames are used for COP-P.
0	1	Reserved for future application.
1	0	TFDF carries user data, with Frame Acceptance Checks bypassed under control of the FARM. These USLP Frames use the BD (Expedited) Service of the COP(-1/-P).
1	1	TFDF carries Protocol Control Commands, with Frame Acceptance Checks bypassed under control of the FARM. These USLP Frames are used for Sequence-Controlled Service of COP-1 or COP-P. Type-BC Transfer Frames are used for COP-1 and SPDUs are used for COP-P.

4.1.2.9 Reserve Spares

4.1.2.9.1 Bits 50–51 of the Transfer Frame Primary Header shall contain the reserved spares.

4.1.2.9.2 This 2-bit field is reserved for future definition by CCSDS and shall be set to ‘00’.

4.1.2.10 Operational Control Field Flag

4.1.2.10.1 Bit 52 of the Transfer Frame Primary Header shall contain the OCF Flag.

4.1.2.10.2 The OCF Flag shall indicate the presence or absence of the OCF. It shall be ‘1’ if the OCF is present; it shall be ‘0’ if the OCF is not present.

4.1.2.11 Virtual Channel Frame Count Length

4.1.2.11.1 Bits 53–55 of the Transfer Frame Primary Header shall contain the length of the VCF Count field.

4.1.2.11.2 This 3-bit field shall define the length of the VCF Count field.

4.1.2.11.3 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

4.1.2.11.4 The VCF Count Length for a given VC is invariant.

NOTES

1 The Frame Count Length is VC-dependent; that is, this protocol maintains a separate Frame Count for each of the VCs.

2 The maximum VCF Count is constrained by the size of the largest Expedited or Sequence-Controlled Transfer Frame count for that VC.

4.1.2.12 Virtual Channel Frame Count

4.1.2.12.1 The VCF Count field shall be absent when the value of the VCF Count Length field equals ‘000’.

4.1.2.12.2 If present, the VCF Count shall be the final field in the Transfer Frame Header starting in bit 56.

4.1.2.12.3 The VCF Count shall contain a sequential binary count, that is, modulo (maximum VCF Count +1) of each Sequence-Controlled Transfer Frame transmitted within a specific VC (see table 4-2).

4.1.2.12.4 The VCF Count shall be independent per VC and shall increment by 1 for each Sequence-Controlled Frame sent (i.e., the Bypass/Sequence Control Flag is set to '0'). In this case, the Frame Count is called the Sequence-Controlled Counter.

4.1.2.12.5 The VCF Count shall be independent per VC and shall increment by 1 for each Expedited Frame sent (i.e., the Bypass/Sequence Control Flag is set to '1'). In this case, the Frame Count is called the FARM-B (Expedited Frame) Counter.

NOTE – Two separate VCF Counters (i.e., one Expedited and one Sequence-Controlled) can be maintained by VC.

4.1.2.12.6 The VCF Count should not be reset before reaching the maximum VCF Count.

NOTES

1 The purpose of this field is to provide individual accountability for each VC. It verifies the sequentiality of USLP Frames within a VC, thus enabling systematic concatenation of SDUs that extend across USLP Frame boundaries. It also enables verification of the sequentiality of incoming Sequence-Controlled Frames by the FARM.

2 The procedure for assigning the VCF Count (i.e., Frame Sequence Number) to Transfer Frames is defined for COP-1 in reference [9] and for COP-P in reference [10].

3 If the VCF Count is reset because of an unavoidable re-initialization, the completeness of a sequence of Transfer Frames in the related VC cannot be determined, and the reassembly of SDUs cannot be guaranteed.

4 TC and Proximity-1 both require a sequence control count as well as an expedited count for the same VC.

4.1.3 TRANSFER FRAME INSERT ZONE

4.1.3.1 The use of this field shall be optional.

4.1.3.2 Insert Service shall be periodic.

NOTE – The presence, absence, and length of the Insert Zone are controlled by the Physical Channel Managed Parameters.

4.1.3.2.1 If present, the Transfer Frame Insert Zone shall follow, without gap, the Transfer Frame Primary Header (see figure 4-1).

4.1.3.2.2 If the Physical Channel Managed Parameter ‘Presence of Insert Zone’ is set to ‘Present’, then the Insert Zone shall exist in every Transfer Frame transmitted within the same Physical Channel, whose Physical Channel Managed Parameter ‘Frame Type’ equals ‘Fixed Length’, including OID Transfer Frames.

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

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

4.1.4 TRANSFER FRAME DATA FIELD

4.1.4.1 General

4.1.4.1.1 The TFDF shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone, if present (see figure 4-1).

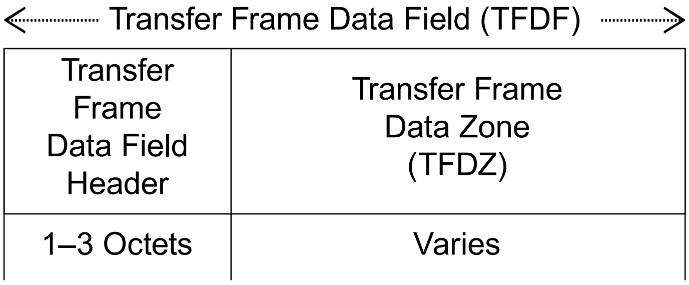
4.1.4.1.2 The TFDF shall

a) contain an integer number of octets; and

b) be of variable length 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 OCF and/or FECF (if any of these are present).

4.1.4.1.3 The TFDF shall consist of a TFDF Header followed by a TFDZ (see figure 4-3 below).

4.1.4.1.4 The TFDF, which shall contain an integral number of octets, may vary in length up to a maximum of 65529 octets.



Transfer Frame Data Field Header	Transfer Frame Data Zone (TFDZ)
1–3 Octets	Varies

Figure 4-3: Transfer Frame Data Field

NOTE – The length of the TFDZ associated with a VC can be constrained by the associated managed parameter ‘VC Transfer Frame Type’ to be either fixed or variable (see 5.4).

4.1.4.1.5 A Transfer Frame with a TFDZ containing idle data shall be transmitted under the following conditions:

a) no valid TFDF is available for transmission at release time for any of the VCs on the physical channel;

b) the managed parameter ‘Physical Channel Transfer Frame Type’ equals ‘Fixed Length’.

NOTE – Such a Transfer Frame is called an OID Transfer Frame.

4.1.4.1.6 The VCID of an OID Transfer Frame shall be set to the value of ‘all ones’ (i.e., 63 decimal).

4.1.4.1.7 VCID 63 shall be the only VC used for OID Transfer Frame transmission.

4.1.4.1.8 The MAP ID of an OID Transfer Frame shall be set to the value of ‘0’.

4.1.4.1.9 The TFDF Header of an OID Transfer Frame shall consist of the TFDZ Construction Rule set to ‘001’, the UPID set to ‘Only Idle Data’, and the Last Valid Octet pointer set to the last octet of the TFDZ.

NOTE – This is a complete MAPA_SDU containing idle data.

4.1.4.1.10 The TFDZ of an OID Transfer Frame shall be filled in by the mandatory Pseudo Noise (PN) sequence generated by a 32-cell Linear Feedback Shift Register (LFSR) with polynomial $D^0 + D^1 + D^2 + D^{22} + D^{32}$, as shown in annex H.

NOTE – Annex H contains example implementations of this LFSR.

4.1.4.1.10.1 An LFSR that follows the Fibonacci form (see figure H-1) shall be initialized at device start-up with an ‘all-ones’ seed and shall not be restarted.

4.1.4.1.10.2 An LFSR that follows the Galois form (see figure H-2) shall be initialized at device start-up with a ‘00000000001111111111111111111101’ seed and shall not be restarted.

NOTES

1 OID Transfer Frames are not sent on VCs that contain valid Packets.

2 OID Transfer Frames are sent to maintain synchronization at the receiver and also to transmit data in the Transfer Frame Insert Zone when there is neither user data nor protocol control data to send in the TFDZ. This is the case when reference [3], [4], or [5] is in use.

3 Idle data in the TFDZ of an OID Transfer Frame is not to be confused with the Idle Packet specified in references [12] and [13].

4 Maintaining a VCF Count for OID Transfer Frames is optional. Details are found in the use of the optional parameters, IN_SDU Loss Flag (3.11.2.4) and OCF_SDU Loss Flag (3.8.2.4).

5 A random pattern is used in the OID Transfer Frame Data Field to avoid frame-reception problems resulting from insufficient randomization.

6 The UPID for OID is registered in the SANA UPID registry (reference [14]).

4.1.4.1.11 An OID TFDF shall not contain any valid user or protocol control data within it.

4.1.4.1.12 An OID Transfer Frame may contain an Insert Zone (indicated by managed parameter) and a FECF (indicated by managed parameter).

4.1.4.2 Transfer Frame Data Field Header

4.1.4.2.1 General

4.1.4.2.1.1 The TFDF Header shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone, if present (see figure 4-1).

4.1.4.2.1.2 The TFDF Header shall consist of 1 to 3 octets and contain the following fields:

a) TFDZ Construction Rules (3 bits, mandatory);

b) UPID (5 bits, mandatory); and

c) First Header/Last Valid Octet Pointer (16 bits, optional).

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

TFDZ Construction Rules	USLP Protocol Identifier	First Header/ Last Valid Octet Pointer (Optional)
3 bits	5 bits	16 bits

Figure 4-4: Transfer Frame Data Field Header

4.1.4.2.2 TFDZ Construction Rules

4.1.4.2.2.1 General

4.1.4.2.2.1.1 The type of TFDZ (fixed vs. variable length) is dependent upon the value of the ‘VC Transfer Frame Type’ managed parameter as follows:

a) if ‘VC Transfer Frame Type’ equals ‘Fixed Length’, then the TFDZ is of fixed length;

b) if ‘VC Transfer Frame Type’ equals ‘Variable Length’, then the TFDZ is of variable length.

4.1.4.2.2.1.2 Bits 0–2 of the TFDF Header shall contain one of the TFDZ Construction Rules defined in 4.1.4.2.2.2.1 through 4.1.4.2.2.2.8.

NOTE – The TFDZ Construction Rules are used to identify how the protocol organizes the data within the TFDZ in order to transport it. The eight TFDZ Construction Rules are summarized in table 4-3.

4.1.4.2.2.1.3 A MAPA_SDU, a VCA_SDU, or a single Packet SDU may be segmented with portions thereof placed within the TFDZ of successive USLP Frames within the same SAP, that is, GMAP ID (for MAPs) or GVCID (for VCs).

4.1.4.2.2.1.4 A MAPA_SDU or a VCA_SDU must always begin in the first octet of the TFDZ in the first USLP Frame carrying that MAPA_SDU or VCA_SDU.

4.1.4.2.2.1.5 The initial Packet placed in a variable-length TFDZ must always begin in the first octet of the TFDZ in the first USLP Frame carrying that Packet.

4.1.4.2.2.1.6 The remaining portions of a Packet, MAPA_SDU, or VCA_SDU placed in a variable-length TFDZ that is longer than the maximum length of the USLP Frame shall be transported in USLP Frames that have the same SAP (GMAP ID for MAPs or GVCID for VCs) and have contiguous VCF Counts.

NOTE – The eight TFDZ Construction Rules are contained in 4.1.4.2.2.2.1 through 4.1.4.2.2.2.8.

4.1.4.2.2.2 Itemized Construction Rules

4.1.4.2.2.2.1 Packets Spanning Multiple Frames. TFDZ Construction Rule '000' shall indicate a fixed-length TFDZ whose contents are CCSDS Packets that are concatenated together and span Transfer Frame boundaries. The First Header Pointer (FHP) is required for packet extraction.

4.1.4.2.2.2.2 Start of a MAPA_SDU or VCA_SDU. TFDZ Construction Rule '001' shall indicate a fixed-length TFDZ that contains either a portion of or a complete MAPA_SDU (or VCA_SDU) that starts in the first octet of the TFDZ. The Last Valid Octet Pointer shall be set to binary 'all ones' if the complete MAPA_SDU (or VCA_SDU) is not fully contained within the TFDZ. Otherwise the value in the Last Valid Octet Pointer shall be the delimiter of the MAPA_SDU (or VCA_SDU) and point to the last valid octet of the MAPA_SDU (or VCA_SDU).

4.1.4.2.2.2.3 Continuing Portion of a MAPA_SDU (or VCA_SDU). The TFDZ Construction Rule '010' shall indicate a fixed-length TFDZ that contains a portion of a MAPA_SDU (or VCA_SDU) that was started in a previous TFDZ. The Last Valid Octet Pointer is required to delimit the end of the MAPA_SDU (or VCA_SDU) and shall contain binary 'all ones' if the end of the MAPA_SDU (or VCA_SDU) is not contained in this TFDZ.

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

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

NOTE – An example is video.

4.1.4.2.2.2.5 Starting Segment. The TFDZ Construction Rule '100' shall indicate a variable-length TFDZ containing only the starting segment of an SDU, that is, either a MAPA_SDU, VCA_SDU, or Packet that is longer than the maximum allowed transmission unit for that SAP, that is, GMAP ID (MAPs) or GVCID (VCs).

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

4.1.4.2.2.2.6 Continuing Segment. The TFDZ Construction Rule '101' shall indicate a variable-length TFDZ containing only a continuing segment (i.e., continuing portion of an SDU contained in the previous TFDZ for either this GMAP ID or GVCID).

4.1.4.2.2.2.7 Last Segment. The TFDZ Construction Rule '110' shall indicate a variable-length TFDZ containing only the last segment of the SDU being transferred for this GMAP ID or GVCID.

4.1.4.2.2.2.8 No Segmentation. The TFDZ Construction Rule '111' shall indicate a variable-length TFDZ that is not segmented, containing either

a) one MAPA_SDU;

b) one VCA_SDU;

c) one or more complete Packets;

d) one or more Control Commands (either COP-1 or COP-P or SDLS); or

e) one or more Proximity-1 SPDUs.

NOTES

1 A series of complete packets can be placed into the TFDZ of a variable-length USLP Frame if the sum of their individual lengths does not exceed the maximum TFDZ size for the maximum frame length allowed for that VC, and if they have the same QoS.

2 The TFDZ Construction Rules are summarized in table 4-3.

Table 4-3: Summary of the TFDZ Construction Rules

TFDZ Construction Rule Value	Applicable to either Fixed or Variable-Length TFDZs	TFDZ Construction Rule Requirement
000	Fixed	4.1.4.2.2.2.1 Packets Spanning Multiple Frames
001	Fixed	4.1.4.2.2.2.2 Start of MAPA_SDU or VCA_SDU (Complete or Portion)
010	Fixed	4.1.4.2.2.2.3 Continuing Portion of MAPA_SDU or VCA_SDU
011	Variable	4.1.4.2.2.2.4 Octet Stream
100	Variable	4.1.4.2.2.2.5 Starting Segment
101	Variable	4.1.4.2.2.2.6 Continuing Segment
110	Variable	4.1.4.2.2.2.7 Last Segment
111	Variable	4.1.4.2.2.2.8 No Segmentation

4.1.4.2.3 USLP Protocol Identifier

4.1.4.2.3.1 Bits 3–7 of the TFDF Header shall contain the UPID.

4.1.4.2.3.2 The UPID shall identify the CCSDS recognized protocol, procedure, or type of data contained within the TFDZ.

4.1.4.2.3.3 The UPID shall be one of the values contained in the SANA UPID registry (reference [14]).

4.1.4.2.4 First Header/Last Valid Octet Pointer

4.1.4.2.4.1 The presence of the First Header/Last Valid Octet Pointer Field is optional and shall be used only for TFDZ Construction Rules '000', '001', and '010'.

4.1.4.2.4.2 If present, the 16-bit First Header/Last Valid Octet Pointer Field shall follow, without gap, the UPID field and shall contain the offset to a specific octet within the TFDZ.

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

4.1.4.2.4.3 When the value in the TFDZ construction rule is '000' binary, the First Header/Last Valid Octet field shall contain the offset within the TFDZ to the first octet of the first packet header that starts within the TFDZ. In this case, this field is designated as the FHP.

NOTE – The function of the FHP is to enable the packet extraction process to restart whenever a Transfer Frame is lost on that VC.

4.1.4.2.4.4 When the value in the TFDZ construction rule is '000' binary, and when no packet starts nor ends within the TFDZ, then the FHP shall be set to binary 'all ones'.

NOTES

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

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

3 The locations of any subsequent packets within the same TFDZ will be determined by calculating the locations using the length field of these packets.

4 If the last packet in the TFDZ of Transfer Frame *M* spills over into USLP Frame *M*+1 of the same VC, then the FHP in USLP Frame *M*+1 points to the start of the next packet within USLP Frame *M*+1.

5 If the last packet in the TFDZ of Transfer Frame *M* spills over into USLP Frame *M*+1 of the same VC but USLP Frame *M*+1 is missing, then the FHP in the next USLP Frame received in that VC ignores the residue of the split packet and points to the start of the next packet that begins in that USLP Frame.

4.1.4.2.4.5 When the value in the TFDZ Construction Rules is binary ‘001’ or ‘010’, the First Header/Last Valid Octet Pointer field shall contain the offset to the last octet of the MAPA_SDU or VCA_SDU being transferred, with the remaining octets composed of idle data (a project-specified idle pattern). In this case this field is designated as the Last Valid Octet Pointer.

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

NOTE – If the length of the TFDZ is fixed and the MAPA_SDU or VCA_SDU contains an insufficient number of user data octets remaining to complete 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 idle data).

4.1.4.3 Transfer Frame Data Zone

4.1.4.3.1 The TFDZ shall follow, without gap, the TFDF Header.

4.1.4.3.2 The data contained within any given TFDZ shall be associated with one and only one GMAP ID (or GVCID), QoS, and UPID.

4.1.4.3.3 The TFDZ shall contain the data defined by the UPID.

NOTES

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

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

3 When the TFDZ Construction Rule is ‘000’, the first and last packets of the TFDZ are not necessarily complete, since the first packet may be a continuation of a packet begun in the previous TFDZ, and the last packet may continue in the subsequent TFDZ of the same VC.

4.1.4.3.4 If a fixed-length TFDZ is partially completed with Packets when the required release time for a Transfer Frame of a VC has been reached, an Encapsulation Idle Packet (reference [13]) shall complete the remainder of the TFDZ.

NOTES

1 This event can occur based on the timeliness criteria contained in the managed parameters.

2 Idle data in the TFDZ is not to be confused with the Idle Packet specified in reference [12] or [13].

4.1.5 OPERATIONAL CONTROL FIELD

4.1.5.1 The presence of the OCF is signaled; its presence or absence shall be signaled by the OCF Flag in the Transfer Frame Primary Header for each applicable VC.

4.1.5.2 If present, the OCF shall occupy the four octets following, without gap, the TFDF, if the TFDF is present, or the Insert Zone, if the TFDF is not present, or the Transfer Frame Primary Header, if the Insert Zone and the TFDF are not present (see figure 4-1).

4.1.5.2.1 Bit 0 of the OCF shall contain a Type Flag with the following meanings:

a) Type Flag '0': the OCF holds a Type-1-Report that contains either a CLCW or a PLCW; and

b) Type Flag '1': the OCF holds a Type-2-Report.

NOTE – The value of the Type Flag can vary between Transfer Frames on the same VC that carries this field.

4.1.5.2.2 In a Type-2 Report, bit 1 of the OCF shall indicate the use of this report as follows:

a) if this bit is '0', the contents of the report are project-specific;

b) if this bit is '1', the contents of the report contain an SDLS FSR defined in reference [16].

4.1.5.3 If the MC managed parameter 'Inclusion of OCF Required' is 'true', then an OCF shall be included in every USLP Frame of that MC.

4.1.5.4 If the managed parameter 'VC Transfer Frame Type' is 'Variable Length' and the associated VC managed parameter, 'Inclusion of OCF Allowed' is set to 'true', then the OCF shall be included as desired based upon mission rules.

NOTES

1 The purpose of this field is to provide a standardized mechanism for reporting a small number of real-time functions (such as supporting the reporting mechanism of the COP-1/COP-P retransmission control procedure or SDLS); currently the use for retransmission control (Type-1 Reports) has been defined by CCSDS in reference [F6] for COP-1 CLCW and reference [10] for COP-P PLCW.

2 In Type-2 Reports, the value of bit 1 of the OCF may vary between Transfer Frames on the same VC that carries this field.

3 The mission could specify for a variable-length USLP Frame that one will only include a new OCF or may require the inclusion of an OCF for a number of times until a new OCF_SDU is received.

4 It is up to the spacecraft designer to ensure that VCs carrying OCFs are transmitted frequently enough to not disrupt the intended behavior of the COP in use that may time out if OCFs are not delivered in a timely fashion.

4.1.6 FRAME ERROR CONTROL FIELD

4.1.6.1 Overview

The purpose of the FECF is to provide a capability for detecting errors that may have been introduced into the Transfer Frame during the transmission-and-data-handling process.

4.1.6.2 General

4.1.6.2.1 The presence (or absence) of the FECF shall be specified by the managed parameter, 'Presence of Frame Error Control' in table 5-1.

NOTE – Its presence or absence is established by management based upon the value of the Physical Channel Managed Parameter 'Presence of Frame Error Control'.

4.1.6.2.2 If present, the FECF shall occupy the last 16 bits of every Transfer Frame transmitted within the same Physical Channel throughout a Mission Phase.

4.1.6.2.3 The FECF shall be computed using the 16-bit coding procedure specified in annex B.

NOTES

1 Whether this field should be used on a particular Physical Channel is determined based on the mission requirements for data quality and the selected options for the underlying Synchronization and Channel Coding Sublayer. This field may be mandatory depending on the selected options for the Synchronization and Channel Coding Sublayer as specified in references [3], [6], or [7], or by management for other reasons besides coding.

2 The CRC-16 procedure provides an undetected bit error rate of approximately 10^{-5} (see annex B for FECF coding procedures).

3 The CRC-16 procedure is identical to the one defined in references [F6], [F19], and [F20].

4.2 PROTOCOL PROCEDURES AT THE SENDING END

4.2.1 OVERVIEW

This subsection describes procedures at the sending end associated with each of the functions shown in figure 4-5 (which is identical to figure 2-6). In the figure, data flow from top to bottom. The figure identifies data-handling functions performed by the protocol entity at the sending end, and shows logical relationships among these functions. The 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 the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.

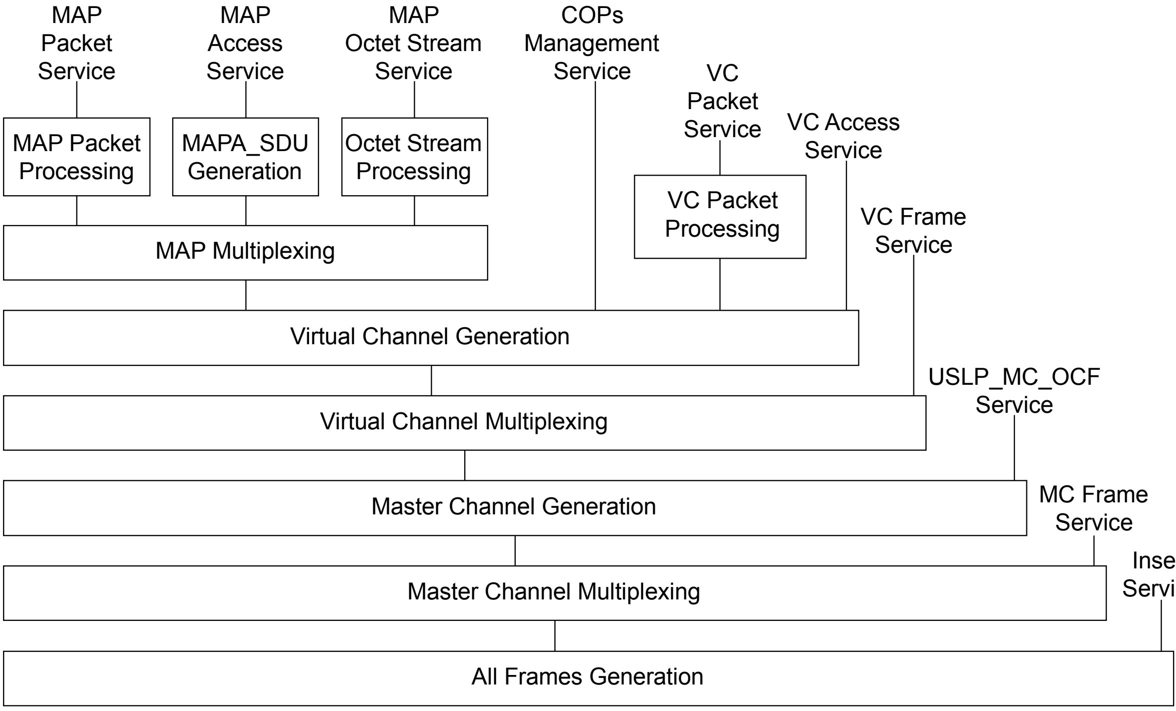


Figure 4-5: Internal Organization of Protocol Entity (Sending End)

4.2.2 MAP PACKET PROCESSING FUNCTION

4.2.2.1 MAP Packet Processing for Fixed-Length TFDZ

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

NOTES

1 When a packet spans a Transfer Frame, the associated TFDZ Construction Rule used is ‘000’.

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

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

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

NOTE – The associated TFDZ Construction Rule used is '000'.

4.2.2.1.4 The FHP field shall be set to indicate the location of the first octet of the first Packet occurring within the TFDZ.

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

NOTES

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

2 An abstract model of the MAPP Processing Function for fixed TFDFs is illustrated in figure 4-6.

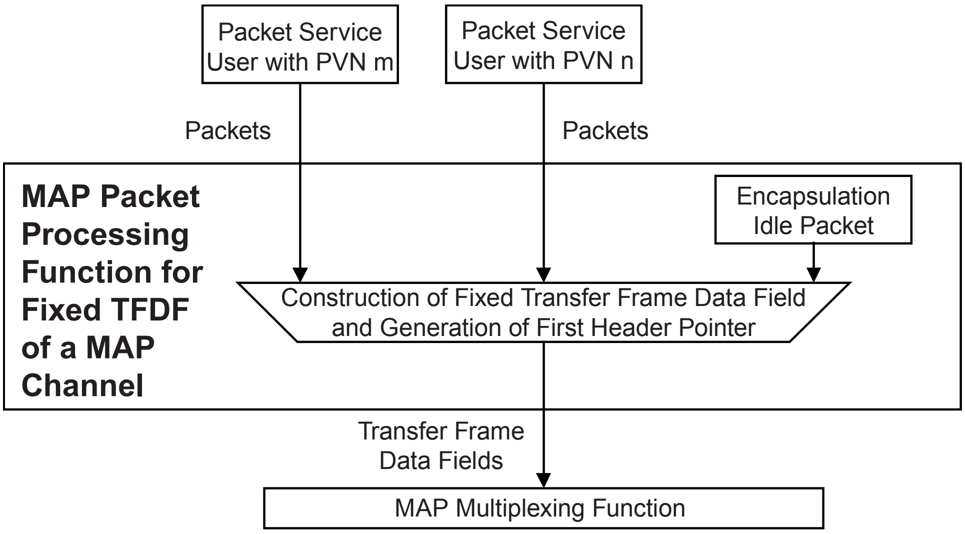

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
MAP Packet Processing Function for Fixed TFDF of a MAP Channel
Encapsulation Idle Packet
Construction of Fixed Transfer Frame Data Field and Generation of First Header Pointer
Transfer Frame Data Fields
MAP Multiplexing Function

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

4.2.2.2 MAP Packet Processing for Variable-Length TFDZ

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

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

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

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

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

NOTE – The associated TFDZ Construction Rule used is '100'.

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

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

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

NOTE – The associated TFDZ Construction Rule used is '110'.

4.2.2.2.7 The portions of a Packet shall be transferred in consecutive Transfer Frames of that VC with the same QoS without being interlaced with any other Packets or portions in the same VC (with the same QoS).

NOTE – Packet completeness can be verified by monitoring the sequentiality of the VC frame count.

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

a) multiple complete Packets may be placed into a TFDF with a single TFDF Header preceding them;

b) the blocked Packets plus the TFDF Header must fit within the maximum size TFDF permitted for the VC.

NOTES

1 The associated TFDZ Construction Rule used is '111'.

2 An abstract model of the MAPP Processing Function for variable TFDFs is illustrated in figure 4-7.

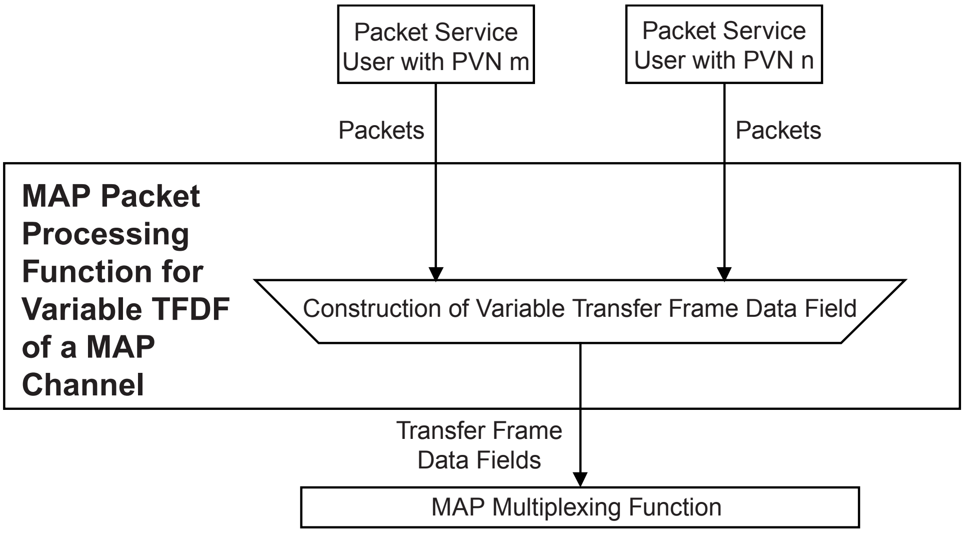

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
MAP Packet Processing Function for Variable TFDF of a MAP Channel
Construction of Variable Transfer Frame Data Field
Transfer Frame Data Fields
MAP Multiplexing Function

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

4.2.3 MAPA_SDU GENERATION FUNCTION

4.2.3.1 MAPA_SDU Generation Function for Fixed-Length TFDZ

4.2.3.1.1 The MAPA_SDU Generation Function shall be used to transfer variable-length user-defined SDUs (MAPA_SDUs) in fixed-length TFDZs of Transfer Frames of a MAP Channel.

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

4.2.3.1.2 If the MAPA_SDU to be transferred exceeds the maximum transmission unit size of the TFDZ, the MAPA_SDU Generation Function shall divide it into portions that are compatible with insertion into the TFDZ and generate a TFDF Header consisting of the TFDZ Construction Rule, UPID, and Last Valid Octet Pointer, forming the TFDFs.

NOTE – The TFDZ Construction rule for the first portion is set to ‘001’ while the TFDZ Construction Rule for TFDFs containing the remaining portions will be set to ‘010’ (see 4.1.4.2.2.2).

4.2.3.1.3 If the MAPA_SDU is completely contained in the first TFDZ, then the MAPA_SDU Generation Function shall set the TFDZ Construction Rule to ‘001’ and fill the remainder of the TFDZ with a project-specified idle-data pattern, and the Last Valid Octet Pointer shall be set to the location of the end of the contained MAPA_SDU.

4.2.3.1.4 If a MAPA_SDU did not complete in a previous TFDZ then the next portion of that same MAPA_SDU shall start in the first octet of the TFDZ of the following USLP Frame in that VC in the same MAP Channel.

NOTE – The associated TFDZ Construction Rule used is ‘010’.

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

4.2.3.1.6 If the fixed-length TFDZ is not completed with MAPA_SDU data at release time, the MAPA_SDU Generation Function shall fill the remainder of the TFDZ with a project-specified idle-data pattern.

NOTES

1 There can be multiple instances of the MAPA_SDU Generation Function for each VC identified by different MAPs. This is possible because a TFDZ that contains a MAPA_SDU can only contain a single MAPA_SDU, thus satisfying the one MAP and one UPID requirement.

2 An abstract model of the MAPA_SDU Generation Function for fixed-length TFDFs is illustrated in figure 4-8.

3 The boundary between the end of the MAPA_SDU Data and the beginning of the idle data is indicated by the Last Valid Octet Pointer.

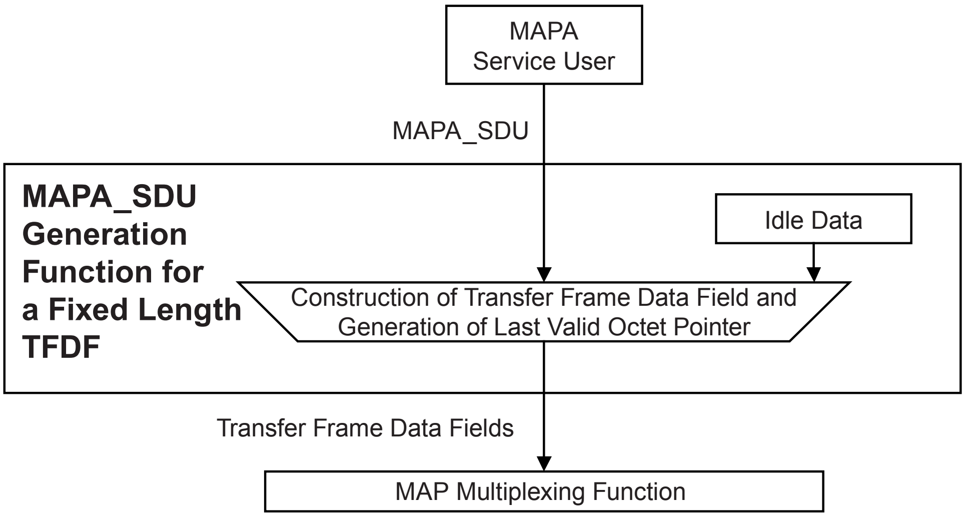

MAPA
Service User
MAPA_SDU
MAPA_SDU
Generation
Function for
a Fixed Length
TFDF
Idle Data
Construction of Transfer Frame Data Field and
Generation of Last Valid Octet Pointer
Transfer Frame Data Fields
MAP Multiplexing Function

Figure 4-8: Abstract Model of MAPA_SDU Generation Function for Fixed-Length TFDFs

4.2.3.2 MAPA_SDU Generation Function for a Variable-Length TFDZ

4.2.3.2.1 The MAPA_SDU Generation Function shall be used to transfer variable-length user-defined SDUs (MAPA_SDUs) in variable-length Data Zones of Transfer Frames of a MAP Channel.

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

4.2.3.2.2 If the MAPA_SDU to be transferred exceeds the maximum transmission unit size of the TFDZ, the MAPA_SDU Generation Function shall divide it into segments that are compatible with insertion into the TFDZ and attach a TFDF Header to each segment, forming a TFDF.

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

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

NOTE – The associated TFDZ Construction Rule used for the starting segment is '100' and for continuing segments is '101'.

4.2.3.2.5 The TFDF containing the last segment of the MAPA_SDU shall contain the TFDF Header and the remainder of the MAPA_SDU.

NOTE – The associated TFDZ Construction Rule used for the last segment is '110'.

4.2.3.2.6 If the MAPA_SDU to be transferred is contained within the maximum unit size of the TFDZ, the MAPA_SDU Generation Function shall attach a TFDF Header to it, forming a TFDF.

NOTE – The associated TFDZ Construction Rule used for no segmentation is '111'.

4.2.3.2.7 The portions of a MAPA_SDU shall be transferred in consecutive Transfer Frames of the MAP Channel with the same QoS without being interlaced with any other MAPA_SDUs or portions in the same MAP Channel (with the same QoS).

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

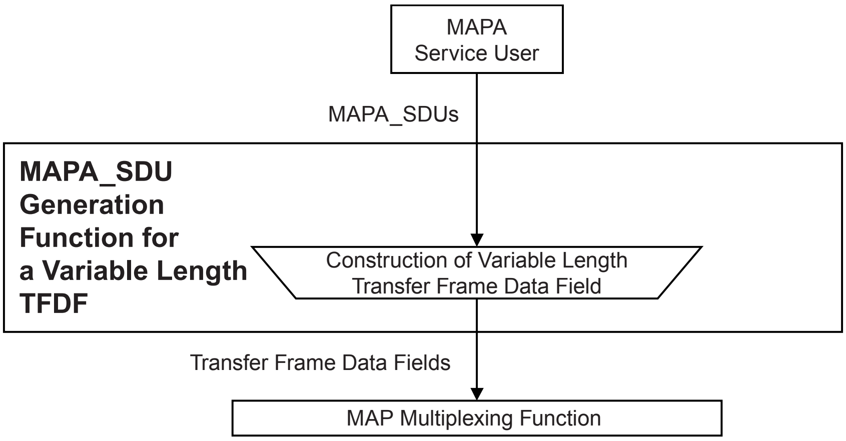

MAPA
Service User
MAPA_SDUs
MAPA_SDU Generation Function for a Variable Length TFDF
Construction of Variable Length Transfer Frame Data Field
Transfer Frame Data Fields
MAP Multiplexing Function

Figure 4-9: Abstract Model of MAPA_SDU Generation Function for Variable-Length TFDFs

4.2.4 MAP OCTET STREAM PROCESSING FUNCTION

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

NOTES

1 One cannot transfer a MAP Octet Stream over fixed-length Transfer Frames.

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

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

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

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

NOTES

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

2 The release of an Octet Stream TFDF may be subject to release timing constraints as per managed parameter 'Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released' in table 5-3.

3 An abstract model of the MAP Octet Stream Processing Function is illustrated in figure 4-10.

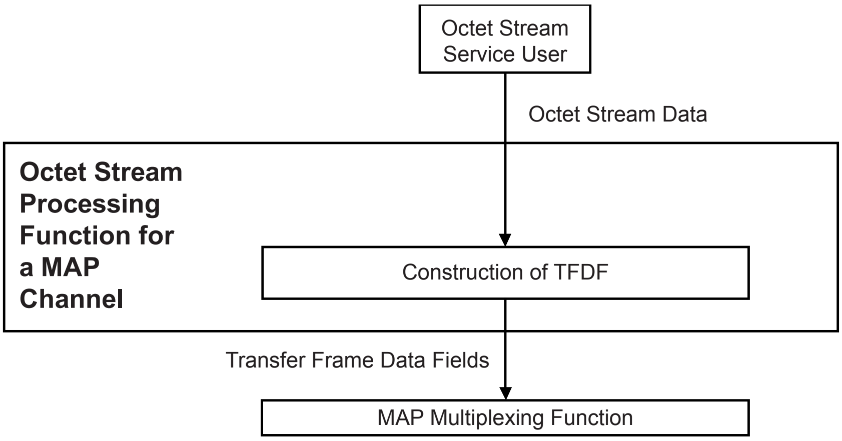

Octet Stream Service User
Octet Stream Data
Octet Stream Processing Function for a MAP Channel
Construction of TFDF
Transfer Frame Data Fields
MAP Multiplexing Function

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

4.2.5 MAP MULTIPLEXING FUNCTION

4.2.5.1 The MAP Multiplexing Function shall be used to multiplex the contents of the TFDZ of different MAP Channels of a VC.

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

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

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

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

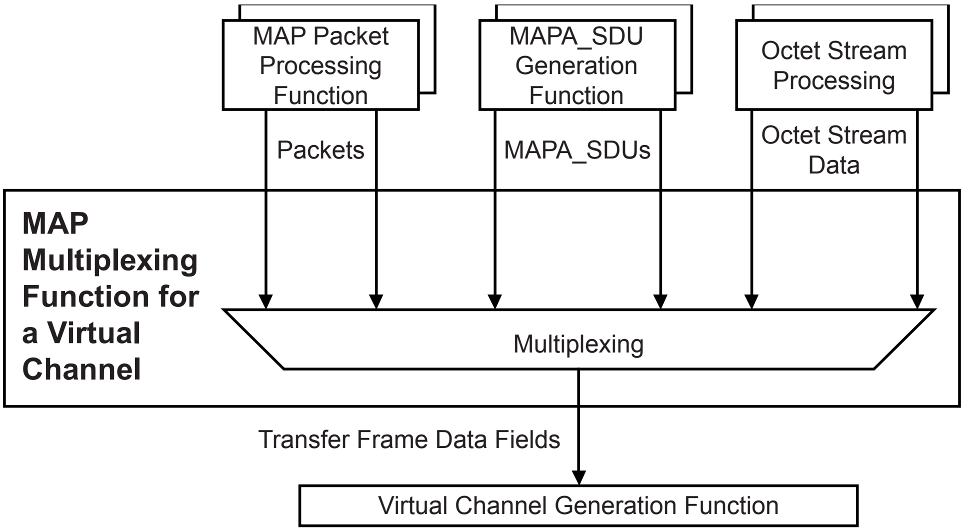

MAP Packet Processing Function
MAPA_SDU Generation Function
Octet Stream Processing
Packets
MAPA_SDUs
Octet Stream Data
MAP Multiplexing Function for a Virtual Channel
Multiplexing
Transfer Frame Data Fields
Virtual Channel Generation Function

Figure 4-11: Abstract Model of MAP Multiplexing Function

4.2.6 VC PACKET PROCESSING FUNCTION

4.2.6.1 VC Packet Processing for Fixed-Length TFDZ

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

NOTES

1 When a packet spans a Transfer Frame, the associated TFDZ Construction Rule used is ‘000’.

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

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

4.2.6.1.3 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.

NOTE – The associated TFDZ Construction Rule used is ‘000’.

4.2.6.1.4 The FHP field shall be set to indicate the location of the first octet of the first Packet occurring within the TFDZ.

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

NOTES

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

2 An abstract model of the VC Processing Function for fixed-length TFDFs is illustrated in figure 4-12.

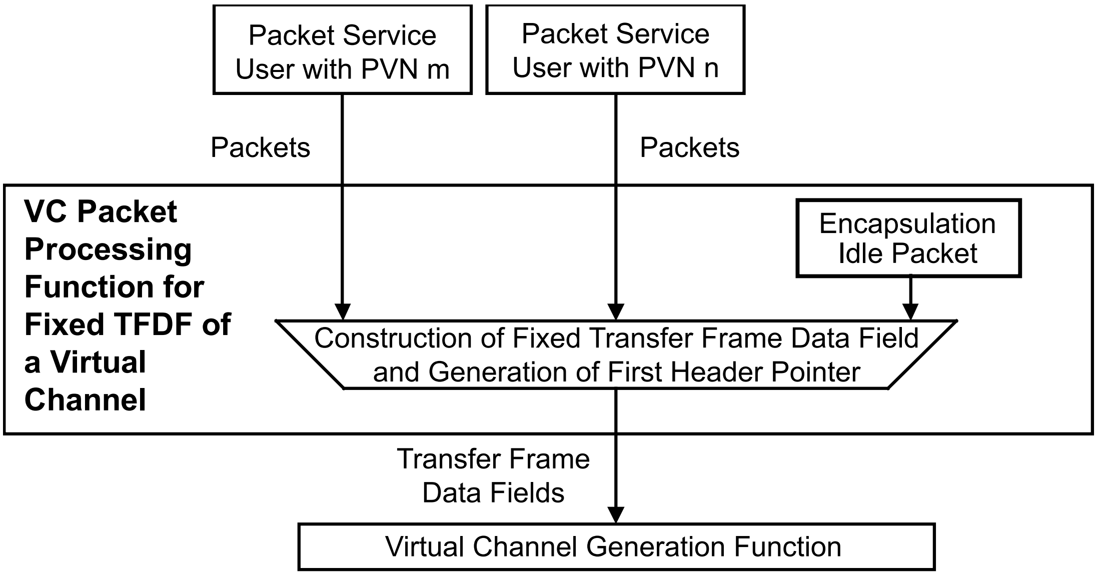

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
VC Packet Processing Function for Fixed TFDF of a Virtual Channel
Encapsulation Idle Packet
Construction of Fixed Transfer Frame Data Field and Generation of First Header Pointer
Transfer Frame Data Fields
Virtual Channel Generation Function

Figure 4-12: Abstract Model of Packet Processing Function for Fixed-Length TFDFs

4.2.6.2 VC Packet Processing for Variable-Length TFDZ

4.2.6.2.1 The VC Packet Processing Function for a variable-length TFDZ shall be used to transfer variable-length Packets in the variable-length Data Zone of Transfer Frames of a Virtual Channel.

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

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

4.2.6.2.3 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.

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

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

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

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

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

NOTE – The associated TFDZ Construction Rule used is '110'.

4.2.6.2.7 The portions of a Packet shall be transferred in consecutive Transfer Frames of that VC with the same QoS without being interlaced with any other Packets or portions in the same VC (with the same QoS).

NOTE – Packet completeness can be verified by monitoring the sequentiality of the VC frame count.

4.2.6.2.8 If blocking of Packets is permitted on a particular Virtual Channel, then:

a) multiple complete Packets may be placed into a TFDF with a single TFDF Header preceding them;

b) the blocked Packets plus the TFDF Header must fit within the maximum size TFDF permitted for the VC.

NOTES

1 The associated TFDZ Construction Rule used is '111'.

2 An abstract model of the VC Packet Processing Function for variable TFDFs is illustrated in figure 4-13.

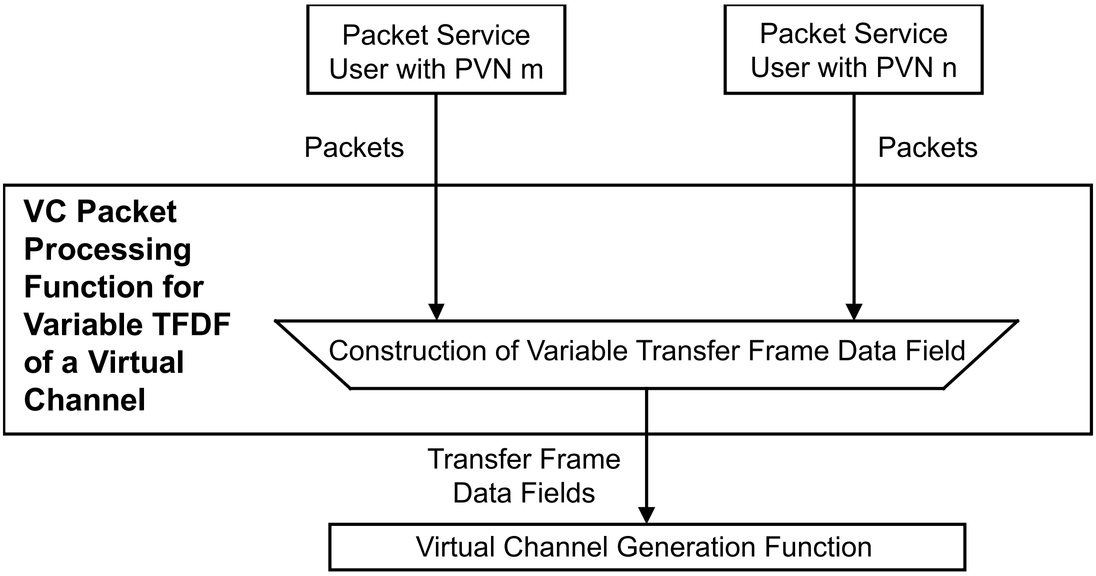

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
VC Packet Processing Function for Variable TFDF of a Virtual Channel
Construction of Variable Transfer Frame Data Field
Transfer Frame Data Fields
Virtual Channel Generation Function

Figure 4-13: Abstract Model of Virtual Packet Processing Function for Variable-Length TFDFs

4.2.7 VIRTUAL CHANNEL GENERATION FUNCTION

NOTE – The VC 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 VC. It also performs most of the operations required to move SDUs reliably from the sending end to the receiving end. There is an instance of the VC Generation Function for each VC.

4.2.7.1 General

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

a) the Frame Initialization Procedure;

b) the Frame Operation Procedure (FOP-1/FOP-P), which is a sub-procedure of the Communications Operation Procedure (COP-1/COP-P); and

c) the Frame Finalization Procedure.

4.2.7.2 Frame Initialization Procedure

The Frame Initialization Procedure shall accept Transfer Frame Data Fields (TFDFs) from the MAP Multiplexing Function, the VC Packet Processing Function, or a VCA Service User (one VCA_SDU is treated as one TFDF) and generate a partially complete USLP transfer frame that includes a Transfer Frame Primary Header and a Transfer Frame Data Field.

NOTE – Only a few (static) fields of the Transfer Frame Primary Header are filled in by this procedure.

4.2.7.3 FOP Procedure

4.2.7.3.1 The FOP shall accept TFDFs from the MAP Multiplexing Function and shall control transmission and retransmission of USLP Frames by examining the report contained in the CLCW/PLCWs and generating COP Control Commands.

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

NOTES

1 This Recommended Standard does not specify the interfaces and methods by which CLCWs are delivered to the FOP. The detailed specifications of the FOP-1/FOP-P are given in references [9] and [10].

2 Most of the Control Commands only cause internal processing in the FOP while some of them cause the generation of Transfer Frames, carrying Control Commands for configuring COPs (e.g., ‘Set V(R)’).

4.2.7.4 The Frame Finalization Procedure

The Frame Finalization Procedure shall fill in the values of the remainder of the transfer frame fields by completing the Transfer Frame Primary Header of each transfer frame or Control Command by adding the value delivered by the FOP (i.e., Frame Sequence Number).

NOTES

1 Values supplied by the FOP are inserted in fields in the Transfer Frame Primary Header.

2 The FOP supplies the Bypass/Sequence Control Flag and the Protocol Control Command Flag.

3 If the Bypass/Sequence Control Flag is set to '0', the Frame Sequence Number supplied by the FOP is placed in the VCF Count field.

4 The actual transmission of a Transfer Frame can be delayed by the next function, that is, the VC Multiplexing Function by the multiplexing algorithm defined by management (see 4.2.8.3). The maximum delay is specified by the managed parameter, 'Maximum delay in milliseconds between releases of USLP Frames of the same VC' in table 5-3.

5 An abstract model of the VC Generation Function is illustrated in figure 4-14.

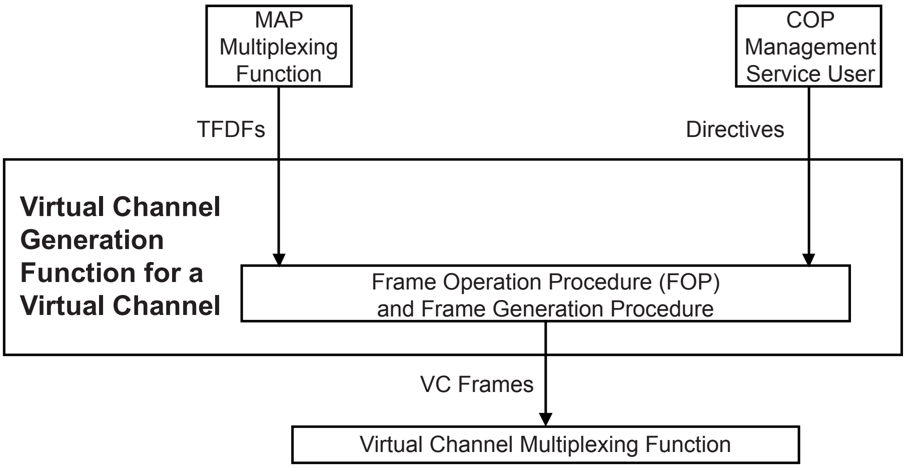

MAP Multiplexing Function
COP Management Service User
TFDFs
Directives
Virtual Channel Generation Function for a Virtual Channel
Frame Operation Procedure (FOP) and Frame Generation Procedure
VC Frames
Virtual Channel Multiplexing Function

Figure 4-14: Abstract Model of Virtual Channel Generation Function

4.2.8 VIRTUAL CHANNEL MULTIPLEXING FUNCTION

4.2.8.1 The VC Multiplexing Function shall be used to multiplex Transfer Frames of different VCs of an MC.

NOTE – There is an instance of the VC Multiplexing Function for each MC that has multiple VCs.

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

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

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

4.2.8.4 An abstract model of the VC Multiplexing Function is illustrated in figure 4-15.

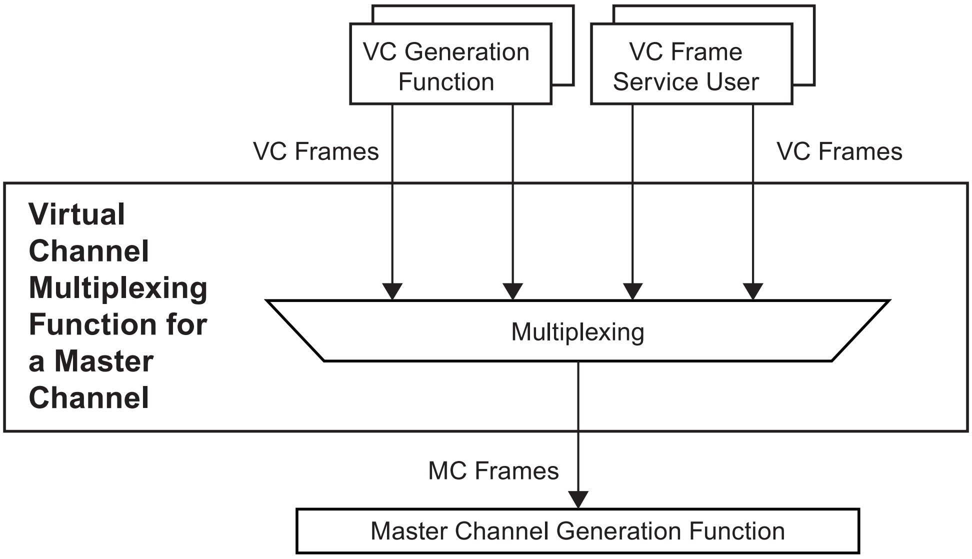

VC Generation Function
VC Frame Service User
VC Frames
VC Frames
Virtual Channel Multiplexing Function for a Master Channel
Multiplexing
MC Frames
Master Channel Generation Function

Figure 4-15: Abstract Model of Virtual Channel Multiplexing Function

4.2.9 MASTER CHANNEL GENERATION FUNCTION

4.2.9.1 The MC Multiplexing Function shall be used to insert OCF_SDUs into Transfer Frames of a VC/MC.

NOTE – There is an instance of the MC Generation Function for each MC.

4.2.9.2 If there is a user of the USLP_MC_OCF Service for a particular VC, an OCF_SDU supplied by the user shall be placed in the OCF of that VC's Transfer Frame according to management (i.e., selective insertion, behavior for fixed-length or variable-length USLP Frames, and/or other).

NOTES

1 No MCF Count is generated in USLP.

2 The VC managed parameter 'Inclusion of OCF Allowed' (applicable to variable-length USLP Frames) and the MC managed parameter 'Inclusion of OCF Required' (applicable to fixed-length USLP Frames) determine if an OCF_SDU can be included in a specific VC.

3 An abstract model of the MC Generation Function is illustrated in figure 4-16.

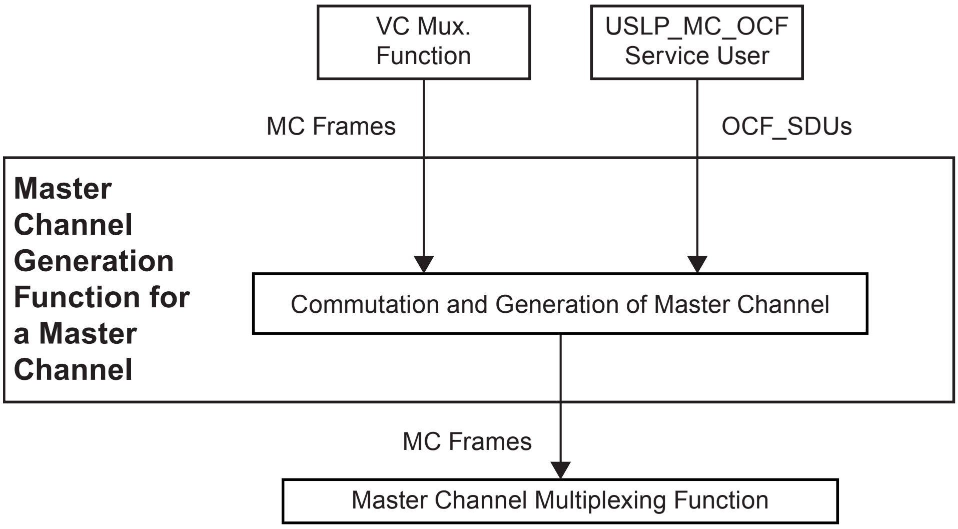

VC Mux. Function
USLP_MC_OCF Service User
MC Frames
OCF_SDUs
Master Channel Generation Function for a Master Channel
Commutation and Generation of Master Channel
MC Frames
Master Channel Multiplexing Function

Figure 4-16: Abstract Model of the Master Channel Generation Function

4.2.10 MASTER CHANNEL MULTIPLEXING FUNCTION

4.2.10.1 The MC Multiplexing Function shall be used to multiplex Transfer Frames of different MCs of a Physical Channel.

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

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

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

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

4.2.10.4 When reference [3], [4], or [5] is in use (i.e., the managed parameter 'Physical Channel Transfer Frame Type' equals 'Fixed Length'), the MC 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.

NOTES

1 It is not required to maintain a VCF Count for OID Transfer Frames.

2 When reference [6] or [7] is in use (i.e., the Physical Channel Transfer Frame Type equals 'Variable Length'), the MC Multiplexing Function does not create an OID Transfer Frame.

3 An abstract model of the MC Multiplexing Function is illustrated in figure 4-17.

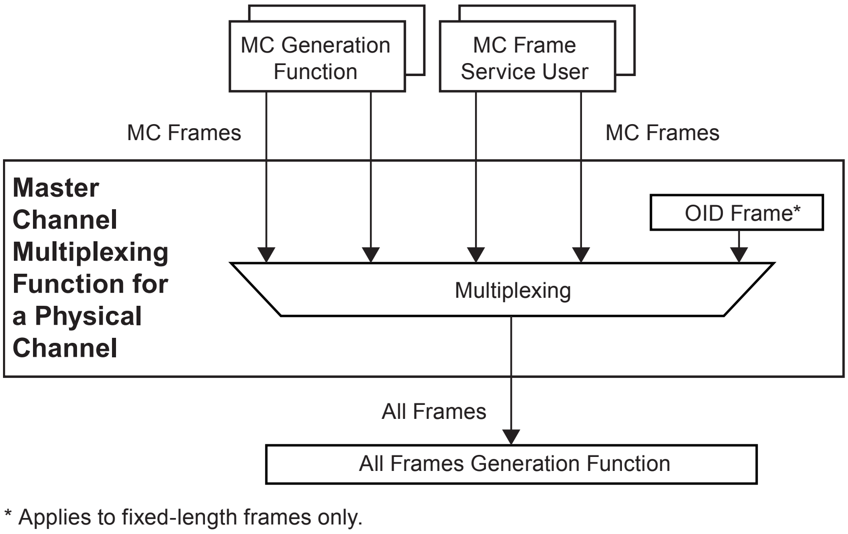

MC Generation Function
MC Frame Service User
MC Frames
MC Frames
Master Channel Multiplexing Function for a Physical Channel
OID Frame*
Multiplexing
All Frames
All Frames Generation Function
* Applies to fixed-length frames only.

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

4.2.11 ALL FRAMES GENERATION FUNCTION

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

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

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

4.2.11.3 If the FECF is present, check bits shall be generated, using the encoding procedure described in annex B, and inserted into the FECF.

4.2.11.4 If the FECF is present, it must be present in all the Transfer Frames transmitted in a particular Physical Channel.

4.2.11.5 Externally generated Transfer Frames associated with the VCF and MCF Services shall always bypass the error control encoding functions specified above.

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

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

4.2.11.7 Each data unit delivered by the All Frames Generation Function shall consist of one or more Transfer Frames as specified by the ‘Maximum Number of Frames Given to the C&S Sublayer as a Single Data Unit’ Managed Parameter.

4.2.11.8 When reference [6] is used as the Synchronization and Channel Coding Sublayer, the All Frames Generation Function may request the Synchronization and Channel Coding Sublayer to perform repeated transmissions of a data unit, that is, ‘systematic retransmissions’ as described in 2.4.2, unless the data unit contains a USLP Frame carrying service data on the Expedited Service (i.e., Type-BD frame in COP-1).

NOTES

1 When systematic retransmissions of a data unit are requested, the additional delay for the retransmissions can be taken into account when deciding the delivery time for the following data unit.

2 An abstract model of the All Frames Generation Function is illustrated in figure 4-18.

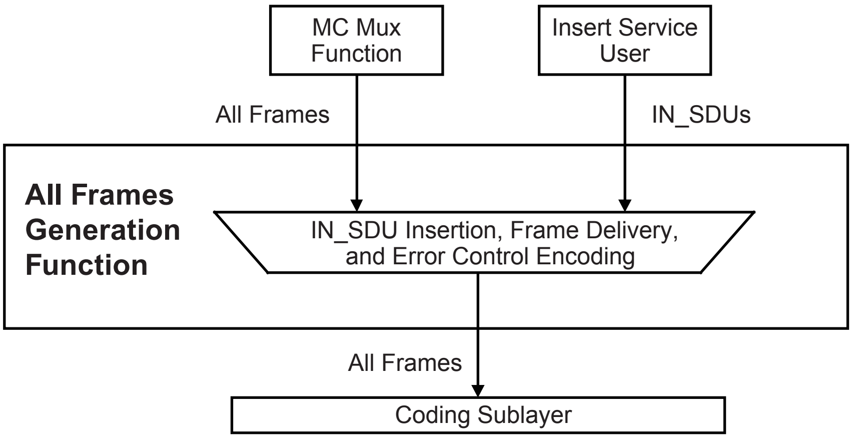

MC Mux Function
Insert Service User
All Frames
IN_SDUs
All Frames Generation Function
IN_SDU Insertion, Frame Delivery, and Error Control Encoding
All Frames
Coding Sublayer

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

4.3 PROTOCOL PROCEDURES AT THE RECEIVING END

4.3.1 OVERVIEW

This subsection describes procedures at the receiving end associated with each of the functions shown in figure 4-19 (which is identical to figure 2-7). In the figure, data flows from bottom to top. The figure identifies data-handling functions performed by the protocol entity at the receiving end and shows logical relationships among these functions. The 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 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.

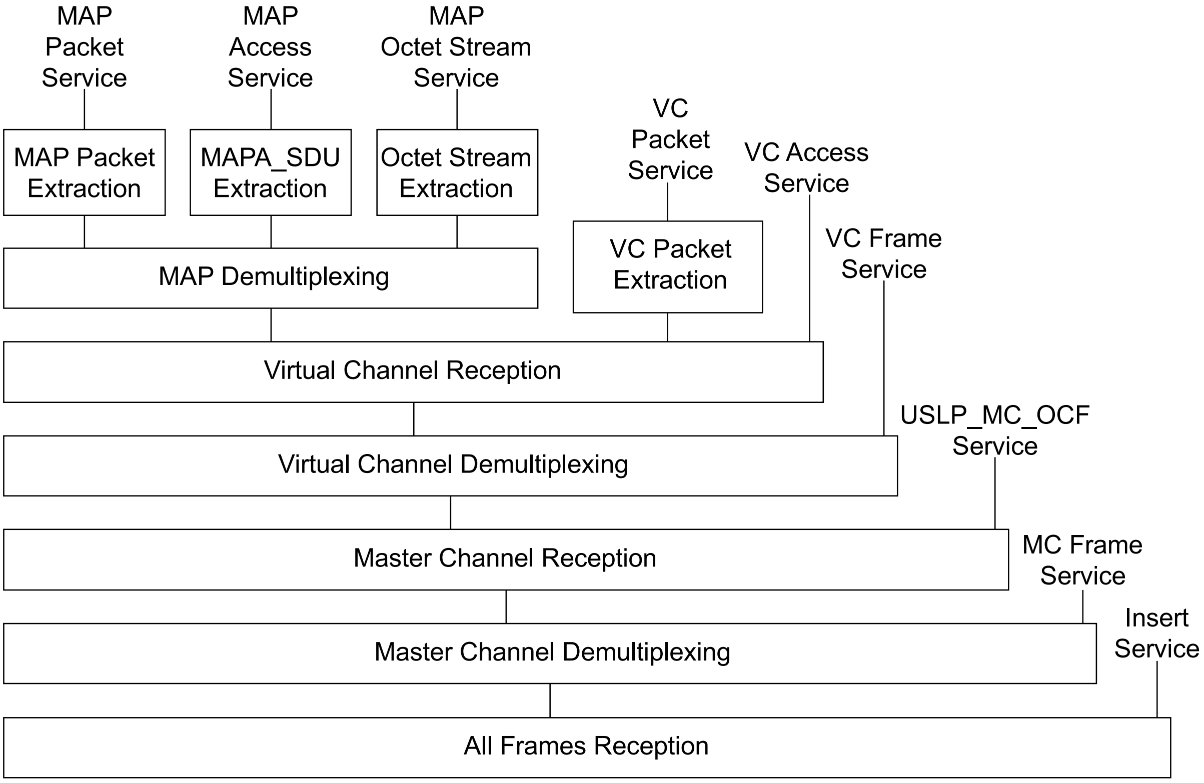


Figure 4-19: Internal Organization of Protocol Entity (Receiving End)

4.3.2 MAP PACKET EXTRACTION FUNCTION

4.3.2.1 Discussion

The extraction of Packets from fixed-length TFDFs is different than from variable-length TFDFs. The TFDZ Construction Rules identify the method used to load the Packets in the TFDF, as well as identify the set of procedures used to extract the packets.

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

4.3.2.2 MAP Packet Extraction Function for a Fixed-Length TFDZ

NOTE – The MAPP Extraction Function used to extract variable-length Packets from the fixed-length TFDZs is associated with TFDZ Construction Rule ‘000’ when a packet spans multiple USLP Frames.

4.3.2.2.1 The MAPP Extraction Function shall extract Packets from TFDZs received from the MAP Demultiplexing Function.

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

4.3.2.2.3 If the last Packet removed from the TFDZ is incomplete, the MAPP Extraction Function shall retrieve its remainder from the beginning of the next sequential TFDF received on the same GMAP ID (VCID + MAP ID).

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

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

NOTES

1 Incomplete Packets are not required to be delivered in cross-support situations. Idle Packets are discarded. TFDFs that contain only idle data are also discarded.

2 An abstract model of the MAPP Extraction Function for fixed-length TFDFs is illustrated in figure 4-20.

4.3.2.2.6 MAPPs associated with an invalid MAP ID shall be discarded.

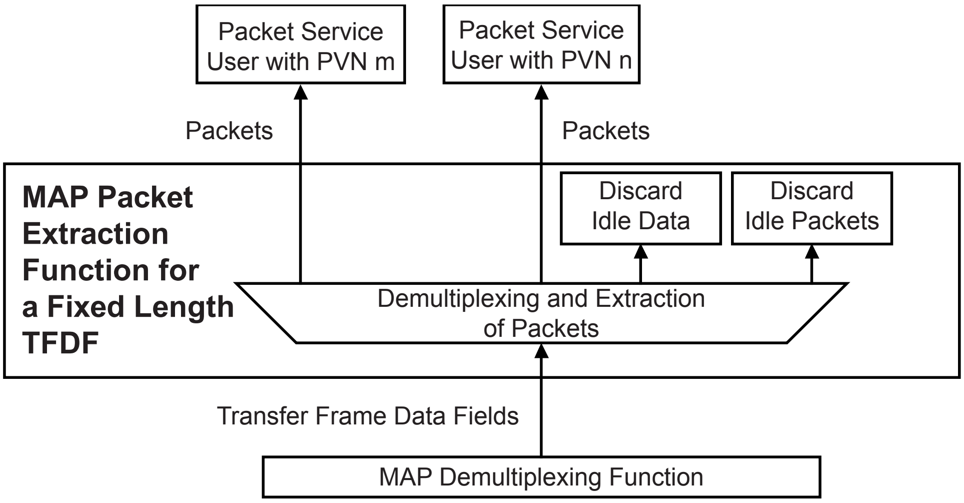

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
MAP Packet Extraction Function for a Fixed Length TFDF
Discard Idle Data
Discard Idle Packets
Demultiplexing and Extraction of Packets
Transfer Frame Data Fields
MAP Demultiplexing Function

Figure 4-20: Abstract Model of MAP Packet Extraction Function for Fixed-Length TFDFs

4.3.2.3 MAP Packet Extraction Function for a Variable-Length TFDZ

NOTE – The MAPP Extraction Function used to extract variable-length Packets from variable-length TFDFs on a MAP Channel is associated with Construction Rules '100', '101', '110' and '111'.

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

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

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

NOTE – Incomplete Packets are not required to be delivered in cross-support situations.

4.3.2.3.4 MAPPs associated with an invalid MAP ID shall be discarded.

NOTE – An abstract model of the MAPP Extraction Function for a variable-length TFDZ is illustrated in figure 4-21.

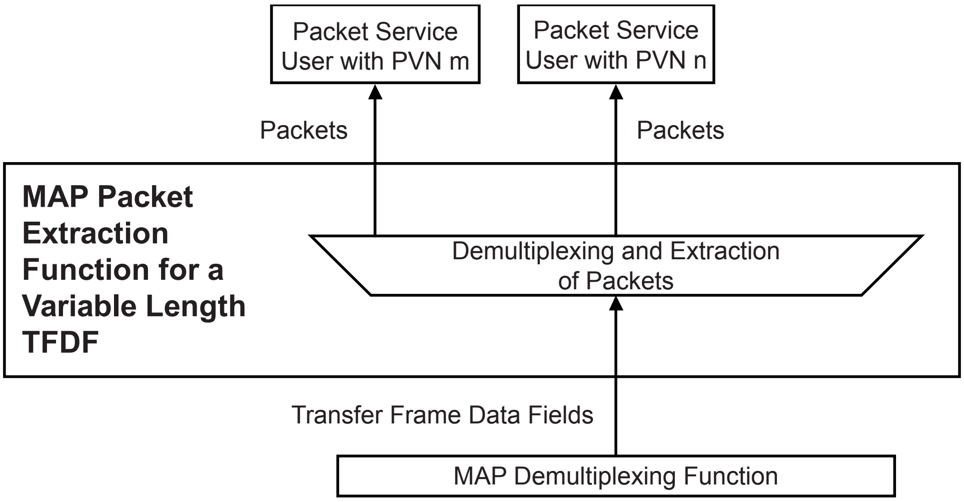

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
MAP Packet Extraction Function for a Variable Length TFDF
Demultiplexing and Extraction of Packets
Transfer Frame Data Fields
MAP Demultiplexing Function

Figure 4-21: Abstract Model of MAP Packet Extraction Function for Variable-Length TFDFs

4.3.3 MAPA_SDU EXTRACTION FUNCTION

4.3.3.1 Overview

The MAPA_SDU Extraction Function extracts one and only one variable-length MAPA_SDU from TFDFs delivered by the MAP Demultiplexing Function and provides it to the MAPA Service user. A different set of TFDZ Construction rules is used to extract a MAPA_SDU from a fixed-length TFDF than from a variable-length TFDF.

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

4.3.3.2 MAPA_SDU Extraction Function for Fixed-Length TFDFs

4.3.3.2.1 The MAPA_SDU Extraction Function shall extract the MAPA_SDU from fixed-length TFDFs received from the MAP Demultiplexing Function based upon the GMAP ID.

4.3.3.2.2 Any idle data inserted by the MAPA_SDU Generation Function shall be removed and discarded prior to delivery to the MAPA Service user, using the value in the Last Valid Octet Pointer.

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

4.3.3.2.4 The Last Valid Octet Pointer that contains a value other than ‘all ones’ delimits the end of the MAPA_SDU within the TFDZ, and any octets that are beyond that pointer value till the end of the TFDZ shall be discarded.

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

4.3.3.2.6 MAPA_SDUs associated with an invalid MAP ID shall be discarded.

NOTE – An abstract model of the MAPA_SDU Extraction Function for fixed-length TFDFs is illustrated in figure 4-22.

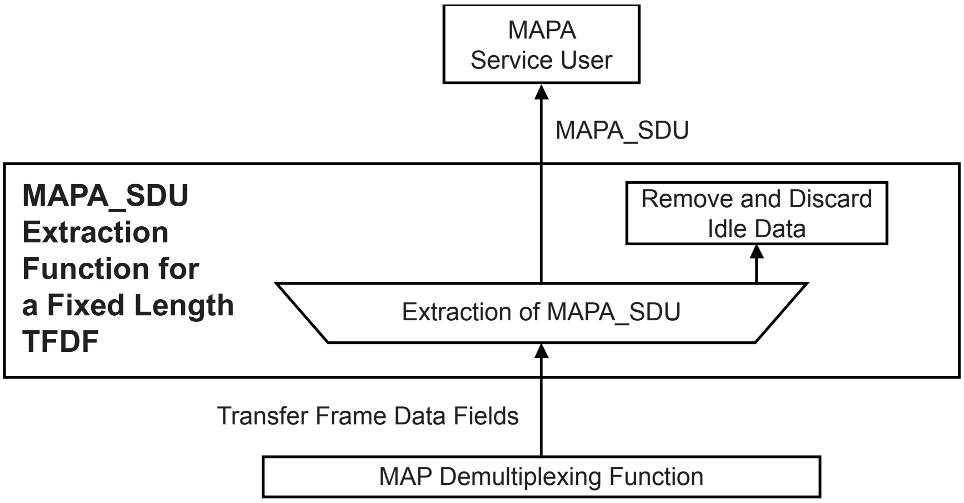

MAPA
Service User
MAPA_SDU
MAPA_SDU Extraction Function for a Fixed Length TFDF
Remove and Discard Idle Data
Extraction of MAPA_SDU
Transfer Frame Data Fields
MAP Demultiplexing Function

Figure 4-22: Abstract Model of MAPA_SDU Extraction Function for Fixed-Length TFDFs

4.3.3.3 MAPA_SDU Extraction Function for Variable-Length TFDFs

4.3.3.3.1 The MAPA_SDU Extraction Function shall extract one and only one variable-length MAPA_SDU from variable-length TFDZs associated with TFDZ Construction rules ‘100’, ‘101’, ‘110’ and ‘111’.

4.3.3.3.2 The MAPA_SDU Extraction Function shall extract the MAPA_SDU from variable-length TFDFs received from the MAP Demultiplexing Function.

4.3.3.3.3 A segmented MAPA_SDU within a variable-length TFDF shall be extracted and reconstructed from the TFDFs using the TFDZ Construction Rules ‘100’, ‘101’, and ‘110’.

4.3.3.3.4 A complete MAPA_SDU contained within a variable-length TFDF shall be extracted from a TFDF using the TFDZ Construction Rule ‘111’.

4.3.3.3.5 An incomplete MAPA_SDU shall be discarded in cross-support situations.

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

4.3.3.3.7 MAPA_SDUs associated with an invalid MAP ID shall be discarded.

NOTE – An abstract model of the MAPA_SDU Extraction Function for variable-length TFDFs is illustrated in figure 4-23.

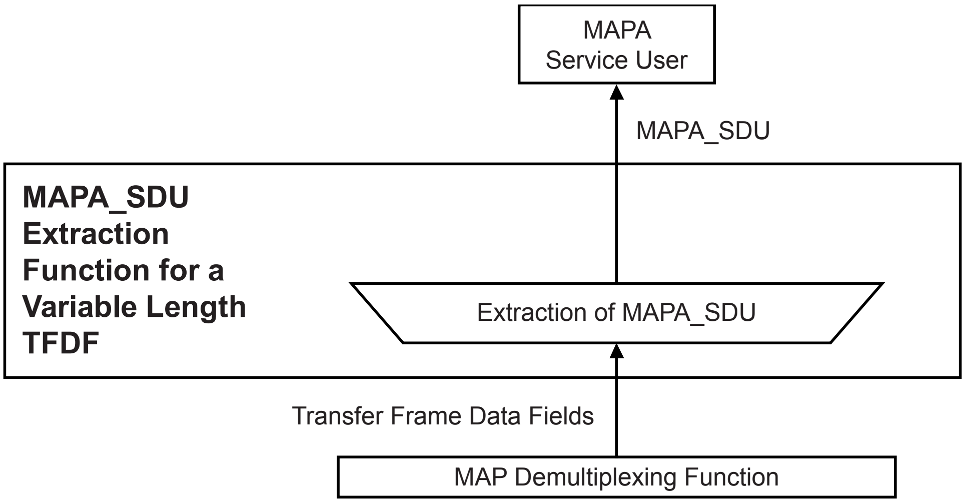

MAPA
Service User
MAPA_SDU
MAPA_SDU Extraction Function for a Variable Length TFDF
Extraction of MAPA_SDU
Transfer Frame Data Fields
MAP Demultiplexing Function

Figure 4-23: Abstract Model of MAPA_SDU Extraction Function for Variable-Length TFDFs

4.3.4 MAP OCTET STREAM EXTRACTION FUNCTION

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

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

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

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

4.3.4.4 Octet Stream Data associated with an invalid MAP ID shall be discarded.

NOTE – An abstract model of the MAP Octet Stream Extraction Function is illustrated in figure 4-24.

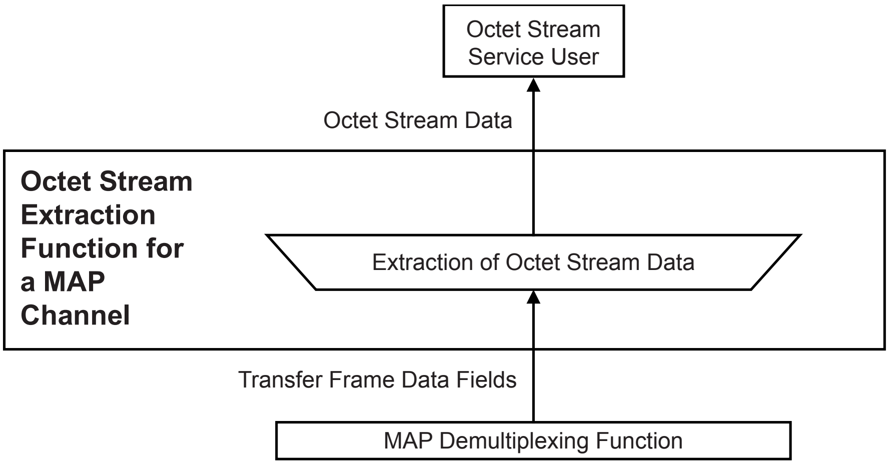

Octet Stream Service User
Octet Stream Data
Octet Stream Extraction Function for a MAP Channel
Extraction of Octet Stream Data
Transfer Frame Data Fields
MAP Demultiplexing Function

Figure 4-24: Abstract Model of MAP Octet Stream Extraction Function

4.3.5 MAP DEMULTIPLEXING FUNCTION

4.3.5.1 The MAP Demultiplexing Function shall be used to demultiplex SDUs from different MAP Channels on a VC.

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

4.3.5.2 The MAP Demultiplexing Function shall examine the MAP ID of the associated incoming SDUs and shall route them accordingly to the instances of the Packet Extraction, MAPA_SDU Extraction, and/or Octet Stream Extraction Functions.

4.3.5.3 SDUs associated with an invalid MAP ID shall be discarded.

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

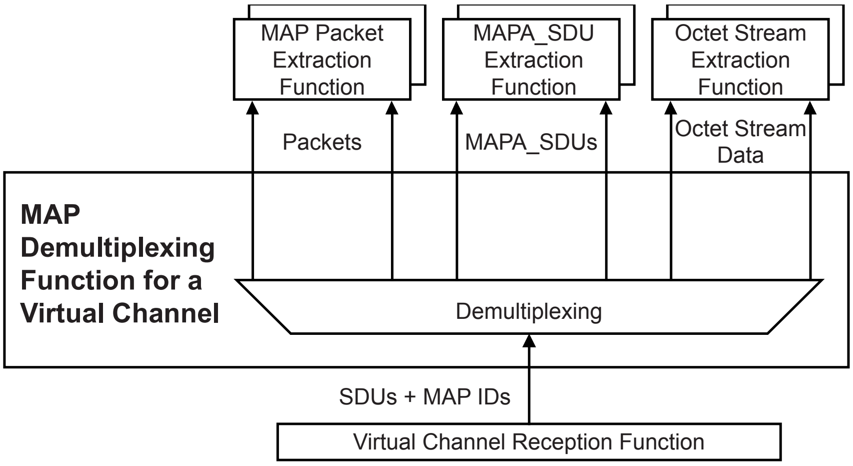

MAP Packet Extraction Function
MAPA_SDU Extraction Function
Octet Stream Extraction Function
Packets
MAPA_SDUs
Octet Stream Data
MAP Demultiplexing Function for a Virtual Channel
Demultiplexing
SDUs + MAP IDs
Virtual Channel Reception Function

Figure 4-25: Abstract Model of MAP Demultiplexing Function

4.3.6 VC PACKET EXTRACTION FUNCTION

4.3.6.1 Discussion

The extraction of Packets from fixed-length TFDFs is different than from variable-length TFDFs. The TFDZ Construction Rules identify the method used to load the Packets in the TFDF, as well as identify the set of procedures used to extract the packets.

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

4.3.6.2 VC Packet Extraction Function for a Fixed-Length TFDZ

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

4.3.6.2.1 The VC Packet Extraction Function shall extract Packets from TFDZs received from the Virtual Channel Reception Function.

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

4.3.6.2.3 If the last Packet removed from the TFDZ is incomplete, the VC Packet Extraction Function shall retrieve its remainder from the beginning of the next sequential TFDF received on the same GVCID.

4.3.6.2.4 The FHP for the next TFDF shall be used to determine the length of the remainder and, hence, the beginning of the next Packet to be extracted.

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

NOTES

3 Incomplete Packets are not required to be delivered in cross-support situations. Idle Packets are discarded. TFDFs that contain only idle data are also discarded.

4 An abstract model of the VC Packet Extraction Function for fixed-length TFDFs is illustrated in figure 4-26.

4.3.6.2.6 MAPPs associated with an invalid MAP ID shall be discarded.

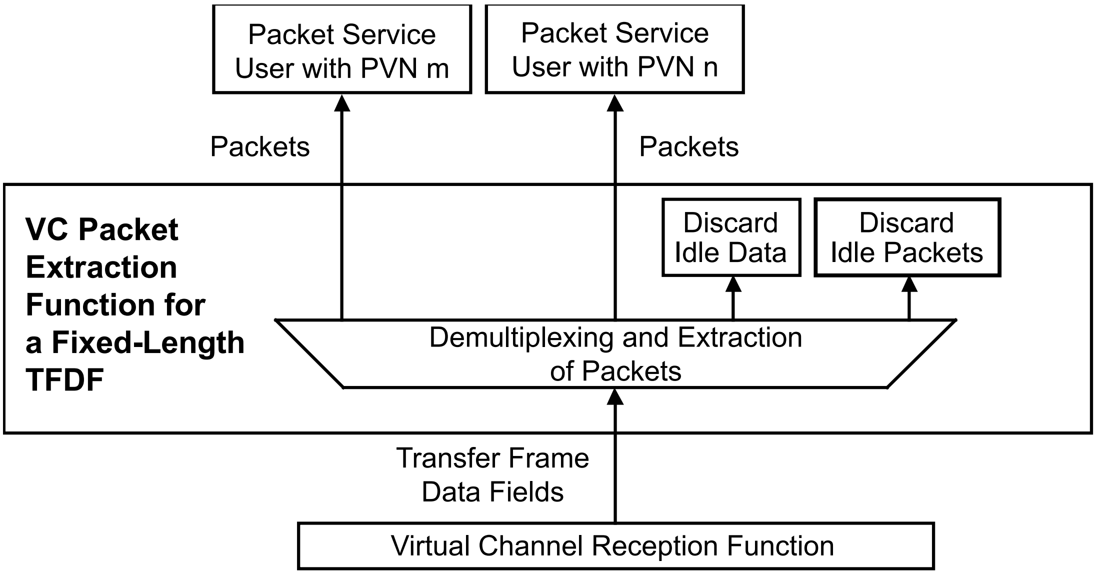

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
VC Packet Extraction Function for a Fixed-Length TFDF
Discard Idle Data
Discard Idle Packets
Demultiplexing and Extraction of Packets
Transfer Frame Data Fields
Virtual Channel Reception Function

Figure 4-26: Abstract Model of VC Packet Extraction Function for Fixed-Length TFDFs

4.3.6.3 VC Packet Extraction Function for a Variable-Length TFDZ

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

4.3.6.3.1 The VC Packet Extraction Function shall extract Packets from variable-length TFDFs received from the Virtual Channel Reception Function.

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

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

NOTE – Incomplete Packets are not required to be delivered in cross-support situations.

4.3.6.3.4 VC Packets associated with an invalid VCID shall be discarded.

NOTE – An abstract model of the VC Packet Extraction Function for a variable-length TFDZ is illustrated in figure 4-27.

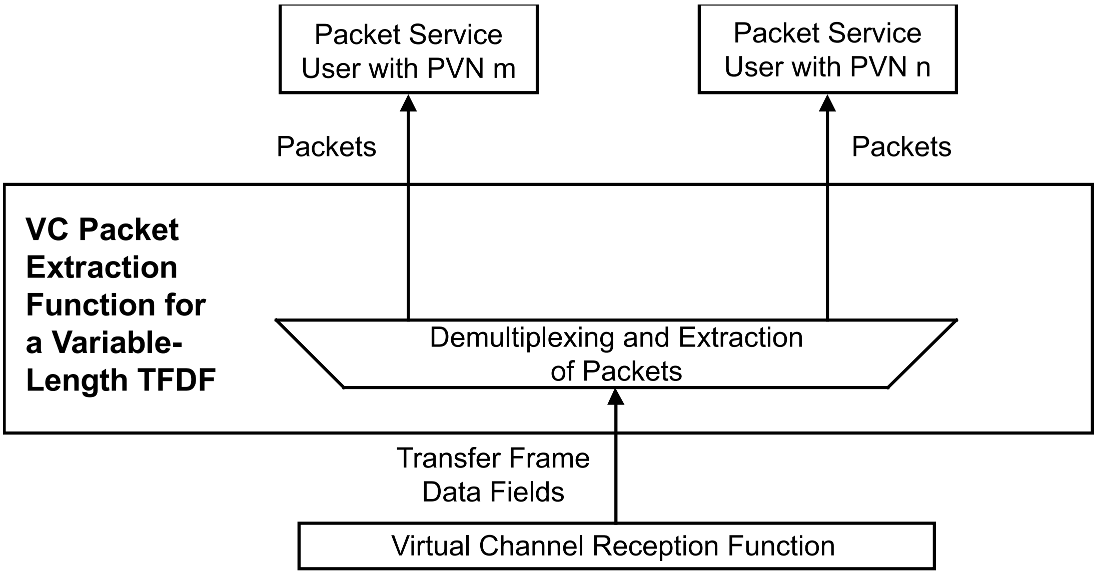

Packet Service User with PVN m
Packet Service User with PVN n
Packets
Packets
VC Packet Extraction Function for a Variable-Length TFDF
Demultiplexing and Extraction of Packets
Transfer Frame Data Fields
Virtual Channel Reception Function

Figure 4-27: Abstract Model of VC Packet Extraction Function for Variable-Length TFDFs

4.3.7 VIRTUAL CHANNEL RECEPTION FUNCTION

4.3.7.1 The VC Reception Function shall:

a) be used to perform the Frame Acceptance and Reporting Mechanism (FARM/FARM-P) when applicable; and

b) decommutate fields of Transfer Frames of a VC.

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

4.3.7.2 The Frame Acceptance and Reporting Mechanism procedure, when applicable, shall perform the operations defined in reference [9] or [10] and, in case of successful checks, shall deliver the VC Frames to the Decommutation Procedure.

NOTES

1 The operation performed by the Frame Acceptance and Reporting Mechanism procedure (i.e., either FARM or FARM-P) include, for example, Accepting/Discarding Frames, generating CLCW/PLCW, etc., as per references [9] and [10].

2 When no FARM is applicable to the given VC, VC Frames are passed directly to the Decommutation Procedure without performing any check.

4.3.7.3 The Decommutation Procedure shall extract the contents of the Transfer Frame Data Zone (Packets, or MAPA_SDU/VCA_SDU, or Octet Stream Data) along with the associated MAP ID or VCID from the Transfer Frame Primary Header and deliver them to the appropriate receiving function as shown in figure 4-28.

4.3.7.4 If a gap in the VCF Count is detected, a Loss Flag may (optionally) be delivered to the MAP Demultiplexing Function, VC Packet Extraction Function, or VCA Service User associated with this VC.

NOTE – An abstract model of the VC Reception Function is illustrated in figure 4-28.

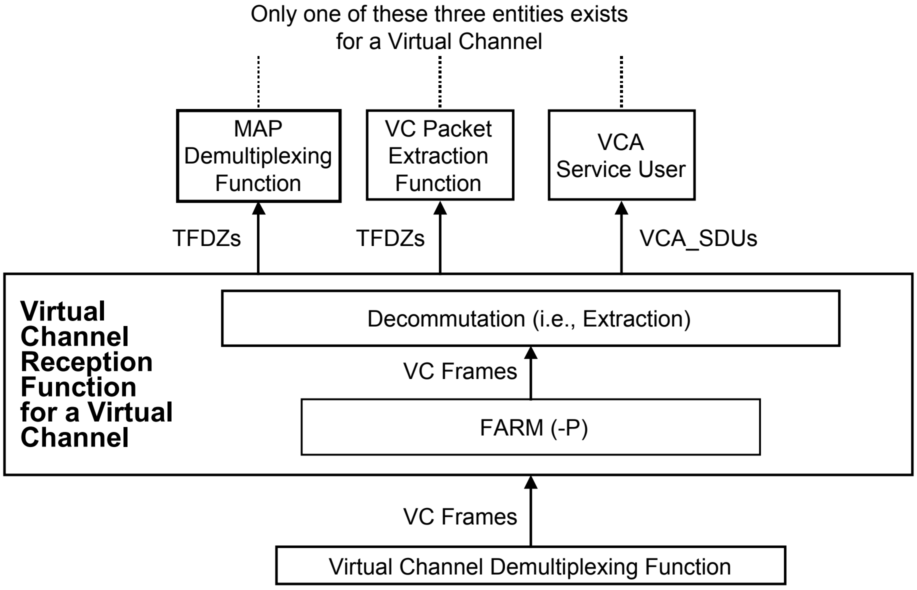

Only one of these three entities exists for a Virtual Channel
MAP Demultiplexing Function
VC Packet Extraction Function
VCA Service User
TFDZs
TFDZs
VCA_SDUs
Virtual Channel Reception Function for a Virtual Channel
Decommutation (i.e., Extraction)
VC Frames
FARM (-P)
VC Frames
Virtual Channel Demultiplexing Function

Figure 4-28: Abstract Model of Virtual Channel Reception Function

4.3.8 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION

4.3.8.1 The VC Demultiplexing Function shall be used to demultiplex Transfer Frames of different VCs of an MC.

NOTE – There is an instance of the VC Demultiplexing Function for each MC that has multiple VCs.

4.3.8.2 The VC Demultiplexing Function shall examine the VCID in the incoming stream of Transfer Frames and route them to the instances of the VC Reception Function and, if present, to the VCF Service users.

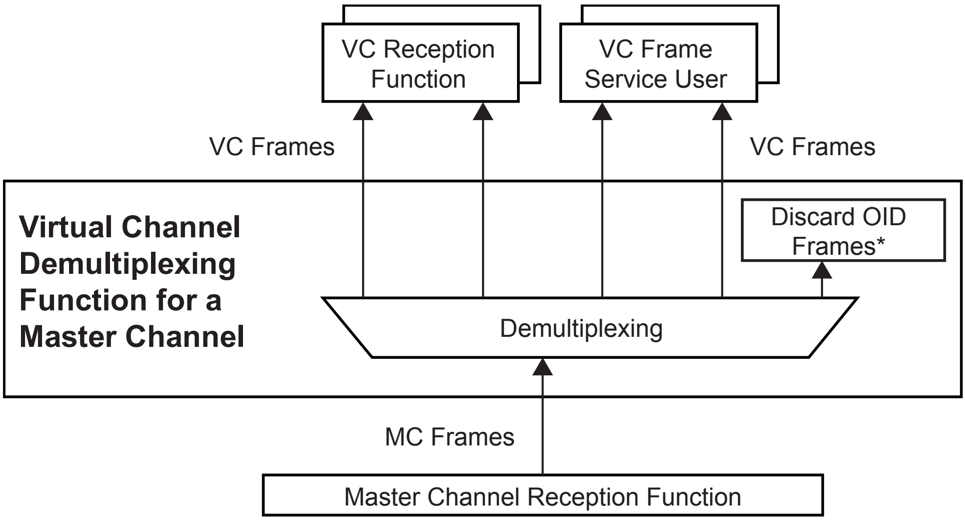
4.3.8.3 If a gap in the VCF Count is detected, a Loss Flag may (optionally) be delivered to the users.

NOTES

1 Any OID Transfer Frames encountered are discarded, once the Insert Zone (if present) is extracted.

2 Transfer Frames with an invalid VCID are also discarded.

3 An abstract model of the VC Demultiplexing Function is illustrated in figure 4-29.



* Applies to fixed-length frames only.

Figure 4-29: Abstract Model of Virtual Channel Demultiplexing Function

4.3.9 MASTER CHANNEL RECEPTION FUNCTION

4.3.9.1 The MC Reception Function shall be used to extract OCF_SDUs contained in the OCF from Transfer Frames of a VC/MC.

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

4.3.9.2 Whenever a Transfer frame contains an OCF, an OCF_SDU shall be extracted from the OCF of the Transfer Frames and delivered to the USLP_MC_OCF Service user.

NOTE – Eventually the USLP_MC_OCF user demultiplexes the received OCFs according to GVCID values.

4.3.9.3 All Transfer frames shall be passed unaltered to the VC Demultiplexing Function.

NOTE – An abstract model of the MC Reception Function is illustrated in figure 4-30.

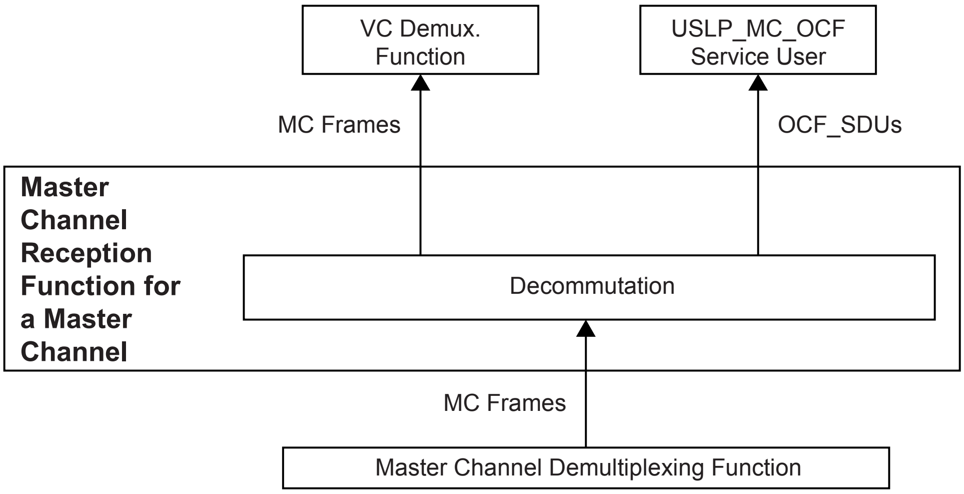

VC Demux. Function
USLP_MC_OCF Service User
MC Frames
OCF_SDUs
Master Channel Reception Function for a Master Channel
Decommutation
MC Frames
Master Channel Demultiplexing Function

Figure 4-30: Abstract Model of Master Channel Reception Function

4.3.10 MASTER CHANNEL DEMULTIPLEXING FUNCTION

4.3.10.1 The MC Demultiplexing Function shall be used to demultiplex Transfer Frames of different MCs of a Physical Channel.

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

4.3.10.2 The MC Demultiplexing Function shall examine the MCID in the incoming stream of Transfer Frames and route them to the instances of the VC Demultiplexing Function and, if present, to the MCF Service users.

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

4.3.10.4 Transfer Frames with an invalid MCID shall be discarded.

NOTE – An abstract model of the MC Demultiplexing Function is illustrated in figure 4-31.

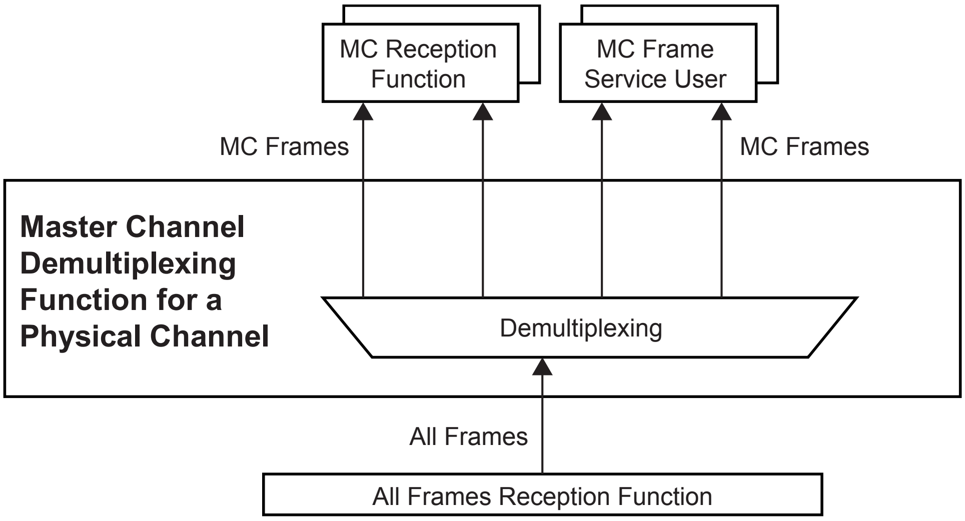

MC Reception Function
MC Frame Service User
MC Frames
MC Frames
Master Channel Demultiplexing Function for a Physical Channel
Demultiplexing
All Frames
All Frames Reception Function

Figure 4-31: Abstract Model of Master Channel Demultiplexing Function

4.3.11 ALL FRAMES RECEPTION FUNCTION

4.3.11.1 General

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

4.3.11.1.1 If the Synchronization and Channel Coding Sublayer in use is *TM Synchronization and Channel Coding* (reference [3]), *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications* (reference [4]), *CCSDS Space Link Protocols over ETSI DVB-S2 Standard* (reference [5]), or *Proximity-1 Space Link Protocol—Coding and Synchronization Sublayer* (reference [7]), the All Frames Reception Function may optionally discard a Transfer Frame that has been validated by the Synchronization and Channel Coding Sublayer if the Synchronization and Channel Coding Sublayer indicates that the USLP Frame was received with error.

NOTE – The Synchronization and Channel Coding Sublayers defined in references [3], [4], [5], and [7] deliver validated Transfer Frames and can optionally signal whether individual USLP Frames were received error free or with error.

4.3.11.1.2 If the FECF 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 FECF to determine if the Transfer Frame contains a detected error.

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

4.3.11.1.4 If the optional Insert Service is used, the All Frames Reception Function shall extract the IN_SDU from the Insert Zone of the incoming stream of Transfer Frames, whose managed parameter 'Physical Channel Transfer Frame Type' is 'Fixed Length', regardless of their GMAP ID, and deliver them to the Insert Service user.

4.3.11.1.5 If error protection of the IN_SDU is not required, extraction of the IN_SDU may be performed prior to decoding of FECF described above.

4.3.11.1.6 If the Synchronization and Channel Coding Sublayer used is *TC Synchronization and Channel Coding* (reference [6]), then the All Frames Reception Function shall be used to reconstitute one or more Transfer Frames from the error free data stream provided by the TC Channel Coding Sublayer. Thereafter, the following two procedures shall be performed to determine whether the reconstituted Transfer Frames are valid or not in the following order:

a) Frame Delimiting and Fill Data Removal Procedure (4.3.11.2); and

b) USLP Frame Validation Check Procedure (4.3.11.3).

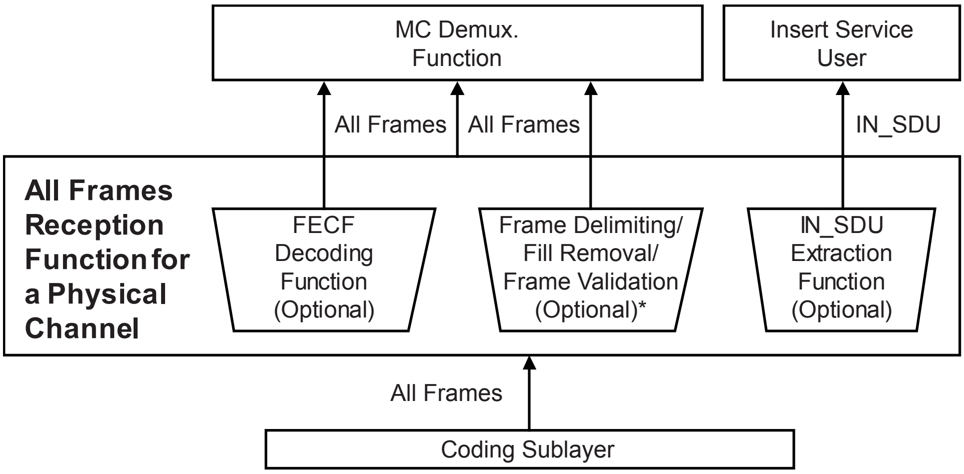
NOTES

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

2 The 'Maximum Number of Frames Given to the C&S Sublayer as a Single Data Unit' is specified as a Managed Parameter in table 5-1.

3 The USLP Frame Validation Check Procedure is used to perform Frame Validation Checks on all Transfer Frames reconstituted by the Frame Delimiting and Fill Data Removal Procedure.

4 An abstract model of the All Frames Reception Function is illustrated in figure 4-32.



*Applicable only when TC Synchronization and Channel Coding (reference [6]) is used.

Figure 4-32: Abstract Model of All Frames Reception Function

4.3.11.2 Frame Delimiting and Fill Data Removal Procedure

NOTES

1 At the sending end, USLP passes one or more Transfer Frames to the TC Synchronization and Channel Coding Sublayer at a time.

2 The TC Synchronization and Channel Coding Sublayer encodes the Transfer Frame(s) to protect it/them from errors that may be introduced as it is transmitted through the space link. Fill Data may have to be inserted by the TC Synchronization and Channel Coding Sublayer to ensure correct transmission of all valid data.

4.3.11.2.1 The receiving end of this protocol shall receive as an input from the TC Synchronization and Channel Coding Sublayer a series of error-free data octets, corresponding to the decoded Transfer Frame, possibly incomplete or containing fill data.

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

4.3.11.2.2 If one or more valid Frame Length fields are detected by the Frame Delimiting and Fill Data Removal 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 USLP Frame Validation Check Procedure (see 4.3.11.3) as it is delimited.

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

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

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

a) assume that the octets following the final expected octet of the USLP Frame are Fill Data appended by the sending end of the Synchronization and Channel Coding Sublayer to complete the last codeword (see reference [6]);

b) discard that Fill Data;

c) pass the USLP Frame to the USLP Frame Validation Check Procedure (see 4.3.11.3).

NOTES

1 Because the receiving end of the TC Synchronization and Channel Coding Sublayer cannot distinguish between valid data and Fill Data, the Fill Data needs to be stripped by this protocol.

2 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 more trailing octets of Fill Data exist (up to six are possible for the BCH code, up to seven for the (128,64) LDPC code, and up to 31 for the (512,256) LDPC code), this procedure might attempt to interpret the Fill Data as a new Transfer Frame Header. In most cases, the subsequent Frame Validation Checks (see 4.3.11.3) 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. The only exception is with the BCH code, if randomization is used, and if the fill data is added after randomization at the transmitting side. In this case, the fill octets will be derandomized without having been randomized, and there is no guarantee that the result will fail the Frame Validation Checks.

3 After each Transfer Frame is reconstituted by the Frame Delimiting and Fill Data Removal Procedure, it will next be subjected to a set of tests called USLP Frame Validation Checks.

4.3.11.3 USLP Frame Validation Check Procedure

4.3.11.3.1 The USLP Frame Validation Check Procedure shall be used if *TC Synchronization and Channel Coding* (reference [6]) is the specification used for the Synchronization and Channel Coding Sublayer immediately below this procedure.

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

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

4.3.11.3.4 The USLP Frame Validation Checks shall consist of the following tests:

a) The Transfer Frame must have an expected TFVN.

b) The Transfer Frame must have one of the expected MCIDs (TFVN and SCIDs).

c) The Transfer Frame Header must not contain any values that are inconsistent with the implemented features for that spacecraft.

d) The number of octets present must be consistent with

   1) the value of the Frame Length field if the Transfer Frame is non-truncated, or

   2) the value of the Truncated Transfer Frame Length managed parameter (see table 5-3) if the Transfer Frame is truncated.

e) If the FECF is present, the recomputed CRC value for the Transfer Frame must match the content of the FECF.

5 MANAGED PARAMETERS WITHOUT SDLS OPTION

5.1 OVERVIEW

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

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.

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

5.2 MANAGED PARAMETERS FOR A PHYSICAL CHANNEL

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

Table 5-1: Managed Parameters for a Physical Channel

Managed Parameter	Allowed Values
Physical Channel Name	Character String
Physical Channel Transfer Frame Type	Fixed Length or Variable Length
Transfer Frame Length (octets)	Integer
TFVN	'1100' binary
MC Multiplexing Scheme	Mission Specific
Presence of Insert Zone	Present ('1'), Absent ('0')
Insert Zone Length (octets)	Integer
Presence of Frame Error Control	Present ('1'), Absent ('0')
Maximum Number of Transfer Frames Given to the C&S Sublayer as a Single Data Unit.	Integer
Maximum Value of the Repetitions Parameter to the Coding and Synchronization Sublayer	Integer

NOTES

1 The Physical Channel Transfer Frame Type is fixed-length when USLP is used over references [3], [4], and [5]. It is variable-length when USLP is used over references [6] and [7].

2 The value of the Transfer Frame Length shows either the USLP Frame length (when Transfer Frame Type is fixed-length) or the maximum allowed USLP Frame length (when Transfer Frame Type is variable-length).

3 The number of allowed values of the Transfer Frame Length may be constrained by the actual coding scheme selected when USLP is used over references [3], [4], and [5].

4 The MC multiplexing function generates OID Frames for fixed-length Transfer Frames when USLP is used over references [3], [4], or [5]. OID Frames are not generated for variable-length Transfer Frames.

5 Insert Zone may only be present when Physical Channel Frame Type is equal to Fixed Length.

5.3 MANAGED PARAMETERS FOR A MASTER CHANNEL

The managed parameters associated with an MC shall conform to the definitions in table 5-2.

Table 5-2: Managed Parameters for a Master Channel

Managed Parameter	Allowed Values
MC Transfer Frame Type	Fixed Length or Variable Length
SCID	Integer (0 to 65535)
VCIDs	Selectable Set of Integers (from 0 to 62 with 63 reserved)
VC Multiplexing Scheme	Mission Specific
Inclusion of OCF Required (only valid if VC Transfer Frame Type = Fixed Length)	True ('1'), False ('0')

NOTES

1 The value of the TFVN is the same for all Transfer Frames on an MC.

2 For VCID the binary value of 'all ones' (i.e., 63) is always valid as it is reserved for OID Transfer Frames by 4.1.4.1.5; that is, the number of valid VCIDs always includes value 63 and the Selectable Set of Integers defined above.

3 MC Transfer Frame Type must be 'Fixed Length' if the Physical Channel Transfer Frame Type is 'Fixed Length'.

5.4 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

The managed parameters associated with each valid VC shall conform to the definitions in table 5-3.

Table 5-3: Managed Parameters for a Virtual Channel

Managed Parameter	Allowed Values
VC Transfer Frame Type	Fixed Length or Variable Length
VCID	One value from a selectable set of integers (from 0 to 62, with 63 reserved)
VC Count Length for Sequence Control QoS	Integer (maximum 56-bit)
VC Count Length for Expedited QoS	Integer (maximum 56-bit)
COP in Effect	COP-1, COP-P, None
CLCW Version Number	1
CLCW Reporting Rate	as required to support COP-1
MAP IDs	Selectable set of integers (from 0 to 15)
MAP Multiplexing Scheme	Mission Specific
Truncated Transfer Frame Length (octets)	Integer (range 6 to 32—see note 5)
SDU Type (Data Field Content)	CCSDS Packet, VCA_SDU
Inclusion of OCF Allowed (only valid if VC Transfer Frame Type = Variable Length)	True (‘1’), False (‘0’)
Inclusion of OCF Required (only valid if VC Transfer Frame Type = Fixed Length)	True (‘1’), False (‘0’)
Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying service data on the Sequence-Controlled Service	Integer (see note 6)
Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying COP Control Commands	Integer (see note 6)
Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released	Integer

Maximum delay in milliseconds between releases of USLP Frames of the same VC	Integer

NOTES

1 VCID value 63 (i.e., the binary ‘all ones’) is reserved for OID Transfer Frames by 4.1.4.1.5.

2 VC Transfer Frame Type must be ‘Fixed-Length’, when either the Physical Channel or MC Transfer Frame Type is ‘Fixed-Length’.

3 Table 6-2 lists additional managed parameters required if SDLS is used.

4 ‘Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released’ managed parameter is used within the MAPP, MAPA, and MAP Octet Stream Services.

5 Annex D provides details on truncated Transfer Frames and for the minimum applicable length.

6 This parameter is only valid if coding sublayer is provided by reference [6].

5.5 MANAGED PARAMETERS FOR A MAP CHANNEL

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

Table 5-4: Managed Parameters for a MAP Channel

Managed Parameter	Allowed Values
MAP ID	One value from a selectable set of integers (from 0 to 15)
SDU Type (Data Field Content)	CCSDS Packet, MAPA_SDU, Octet Stream Data
UPID supported	Integer (see reference [14])

5.6 MANAGED PARAMETERS FOR PACKET TRANSFER

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

Table 5-5: Managed Parameters for Packet Transfer

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

6 PROTOCOL SPECIFICATION WITH SDLS OPTION

6.1 OVERVIEW

This section specifies the PDU and the procedures of USLP with support for the SDLS Protocol (reference [15]). If the USLP protocol entity supports SDLS, it has managed parameters for each VC or each MAP to indicate whether SDLS is in use for that channel (see 6.6). Section 4 contains the specification of the protocol without the SDLS option.

6.2 USE OF SDLS PROTOCOL

If SDLS as defined in reference [15] is required over the USLP space data link, then the SDLS protocol shall be used.

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

6.3 USLP TRANSFER FRAME WITH SDLS

6.3.1 OVERVIEW

To support the use of the SDLS security features, a Security Header and a Security Trailer are defined for a USLP Transfer Frame. The use of SDLS can vary between VCs and between MAPs on a Virtual Channel, so a managed parameter defined in either table 6-2 or table 6-3 of 6.6 indicates the presence of the Security Header. If the Security Header is present, then SDLS is in use for the VC or MAP. This subsection specifies the USLP Transfer Frames on a VC or a MAP that is using SDLS.

If a VC or MAP is not using SDLS, then the USLP Frames are as specified in 4.1.

Transfer frames containing protocol control information, that is, Protocol Control Command Flag in the Transfer Frame Primary Header set to ‘1’, do not contain the Security Header and Security Trailer and are therefore not protected by SDLS. When SDLS applies, the Security Header and Security Trailer (optional) are placed before and after the TFDF, and they reduce the length of the TFDF compared to a USLP Frame without SDLS. Figure 6-1 compares the USLP Frame fields for a USLP Frame without SDLS and a USLP 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 presence of the FECF is controlled by the managed parameters (see section 5).

6.3.2 TRANSFER FRAME PRIMARY HEADER IN A FRAME WITH SDLS

6.3.2.1 The non-truncated Transfer Frame Primary Header for a USLP Frame with SDLS shall conform to the specifications of 4.1.2.

6.3.2.2 The truncated Transfer Frame Primary Header for a USLP truncated Transfer Frame with SDLS shall conform to the specifications of D1.2.5.

NOTE – The Transfer Frame Primary Header (whether truncated or not) is the same for a USLP Frame without SDLS and a USLP Frame with SDLS. (See annex D for the applicability of SDLS for the truncated Transfer Frame.)

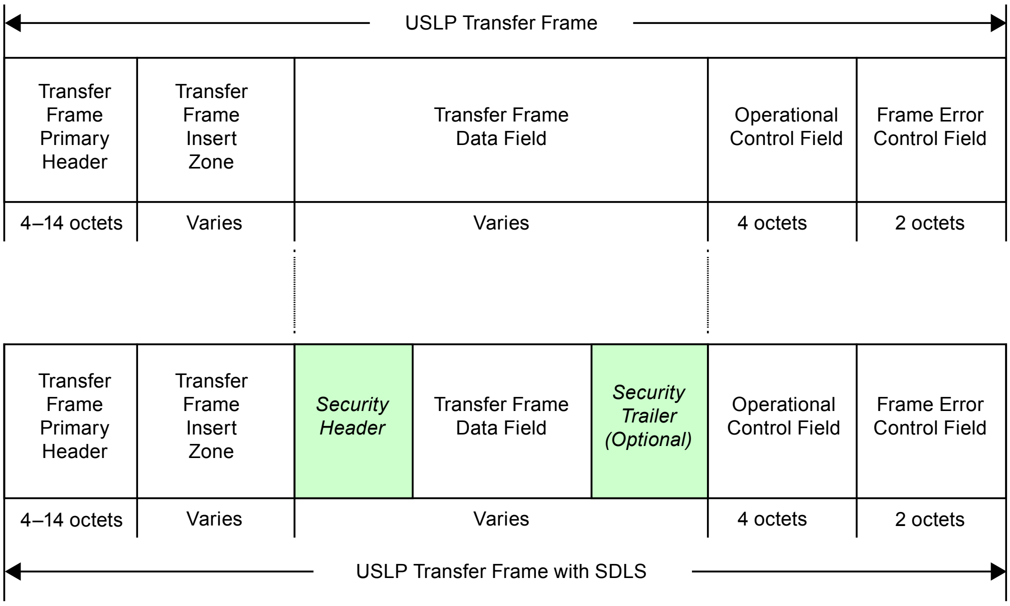

USLP Transfer Frame
Transfer Frame Primary Header
Transfer Frame Insert Zone
Transfer Frame Data Field
Operational Control Field
Frame Error Control Field
4–14 octets
Varies
Varies
4 octets
2 octets
Transfer Frame Primary Header
Transfer Frame Insert Zone
Security Header
Transfer Frame Data Field
Security Trailer (Optional)
Operational Control Field
Frame Error Control Field
4–14 octets
Varies
Varies
4 octets
2 octets
USLP Transfer Frame with SDLS

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

6.3.3 TRANSFER FRAME INSERT ZONE IN A FRAME WITH SDLS

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

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

6.3.4 SECURITY HEADER

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

NOTES

1 The presence of the Security Header is a managed parameter of the VC (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 [15].

3 The length of the Security Header is an integral number of octets and is a managed parameter of the MAP or VC.

6.3.5 TRANSFER FRAME DATA FIELD IN A FRAME WITH SDLS

6.3.5.1 The TFDF of a USLP Frame with SDLS shall conform to the specifications of 4.1.4.1.1 through 4.1.4.1.2 as modified by 6.3.5.2.

6.3.5.2 In a Transfer Frame with SDLS, the TFDF shall

a) follow, without gap, the Security Header;

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

b) contain an integer number of octets equal to the Transfer Frame length, minus

1) the lengths of the Transfer Frame Primary Header and of the Security Header;

2) the lengths of the Transfer Frame Insert Zone, of the Security Trailer, OCF, and of the FECF, if any of these are present.

6.3.6 SECURITY TRAILER

If present, the Security Trailer shall follow, without gap, the TFDF.

NOTES

1 The use of the Security Trailer is optional in a USLP Transfer Frame with SDLS. The presence of the Security Trailer is a managed parameter of the VC or MAP (see 6.6).

2 The requirements for the length and contents of the Security Trailer are specified in reference [15].

3 The length of the Security Trailer is an integral number of octets and is a managed parameter of the MAP or VC.

6.3.7 OPERATIONAL CONTROL FIELD IN A FRAME WITH SDLS

6.3.7.1 The OCF of a USLP Frame with SDLS shall conform to the specifications of 4.1.5.2 through 4.1.5.2.2 as modified by 6.3.7.2.

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

6.3.8 FRAME ERROR CONTROL FIELD IN A FRAME WITH SDLS

6.3.8.1 The FECF of a USLP Frame with SDLS shall conform to the specifications of 4.1.6.2, B1, as modified by 6.3.8.2.

6.3.8.2 In a Transfer Frame with SDLS, the FECF, if present, shall occupy two octets following, without gap,

a) the OCF if this is present;

b) the Security Trailer if this is present and the OCF is not present;

c) the TFDF if the OCF and the Security Trailer are not present.

6.4 SENDING END PROTOCOL PROCEDURES WITH SDLS

6.4.1 OVERVIEW

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

The SDLS ApplySecurity Function may interface with USLP at either the VC Generation Function (4.2.6) or the VC Multiplexing Function (4.2.8). 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 MAP or more than one VC or MAP 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 or MAP each are present. The physical location of the security processing may not be the same for all VCs (or MAPs), at the sending end or at the receiving end. This case can be supported by placing the SDLS interface in the VC Generation Function in which the greatest flexibility in managing the security function occurs.

6.4.2 ORDER OF PROCESSING BETWEEN USLP, COP, AND SDLS PROTOCOLS

6.4.2.1 Virtual Channel Generation Function

In the Virtual Channel Generation Function at the sending end, the order of processing between the functions of the USLP, COP, and SDLS protocols shall occur as follows:

a) the Frame Initialization Procedure including SDLS;

b) the SDLS ApplySecurity Function;

c) the FOP, which is a subprocedure of the COP and an integral part of the Virtual Channel Generation Function (final step of processing by the function);

d) the Frame Finalization Procedure including SDLS.

6.4.2.2 Discussion

For completeness, figure 6-2 shows the order of processing between USLP, COP, and SDLS functions at both the sending and receiving ends in the context of the lower layers of the CCSDS protocol stack. In addition, in table 6-1, a reference is provided to the section within each applicable CCSDS document per numbered step in the figure. The detailed processing of the Virtual Channel Generation Function under the SDLS option differs from that of 4.2.7.

6.4.2.3 Frame Initialization Procedure Including SDLS

6.4.2.3.1 The Frame Initialization Procedure (4.2.7.2) shall be applied.

6.4.2.3.2 A Security Header field and, if in use (see 6.6), a Security Trailer field, whose values at this point are not yet known, shall be added to the partially complete TC transfer frame.

6.4.2.4 SDLS ApplySecurity Function

6.4.2.4.1 The SDLS ApplySecurity Function (reference [15]) shall be called to process a USLP Transfer Frame to apply security features to the frame.

6.4.2.4.2 The input parameters of the function shall include the partially formatted frame and the identifiers of the Virtual Channel or the MAP channel.

NOTE – When the function is called, SDLS applies encryption and/or authentication to the data in the frame. The way the Transfer Frame data is passed between the Virtual Channel Generation Function and the SDLS ApplySecurity Function is implementation dependent. Reference [15] defines which transfer frame data fields apply for this function.

6.4.2.5 FOP Procedure

When COP-1 is used, the FOP-1 Procedure specified in reference [9] shall be applied. When COP-P is used, the FOP-P Procedure specified in reference [10] shall be applied.

6.4.2.6 Frame Finalization Procedure Including SDLS

6.4.2.6.1 The Frame Finalization Procedure (4.2.7.4) shall be applied.

6.4.2.6.2 The values provided by the SDLS ApplySecurity Function shall be inserted into the Security Header and, if in use (see 6.6), Security Trailer within the Transfer Frame.

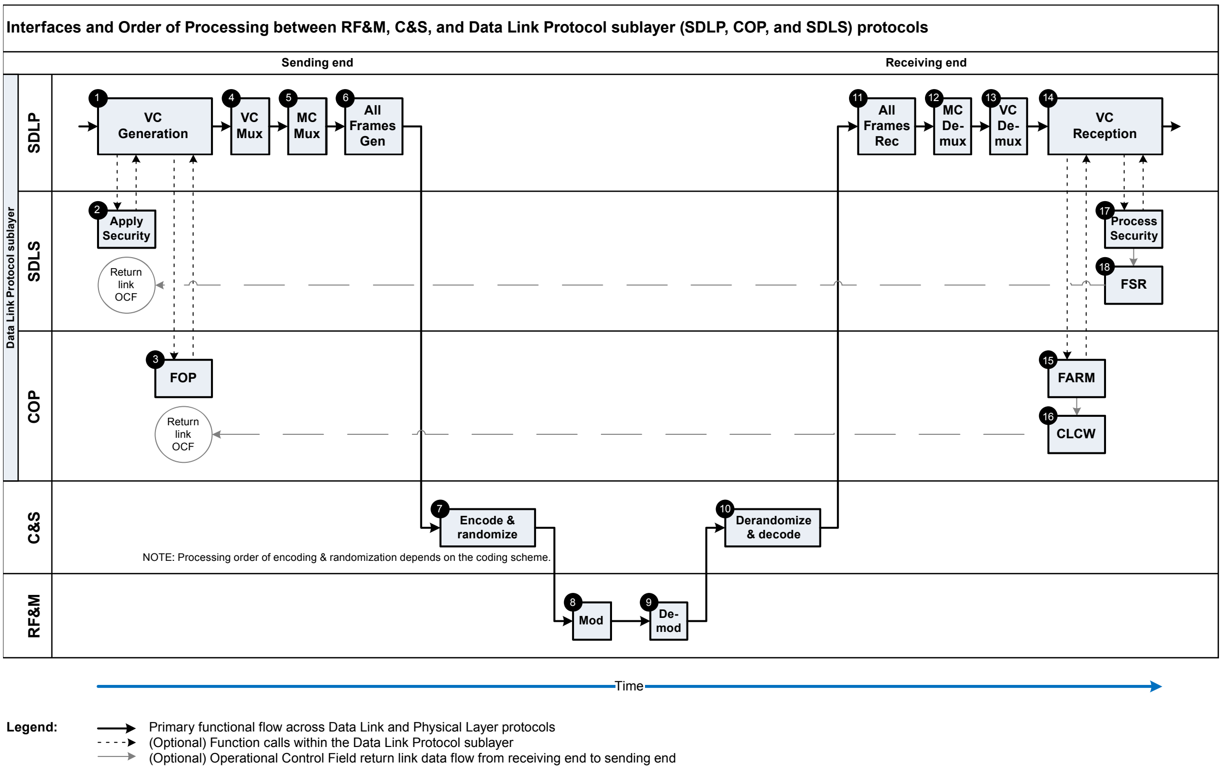

Interfaces and Order of Processing between RF&M, C&S, and Data Link Protocol sublayer (SDLP, COP, and SDLS) protocols
Sending end
Receiving end
Data Link Protocol sublayer
SDLP
SDLS
COP
C&S
RF&M
1 VC Generation
4 VC Mux
5 MC Mux
6 All Frames Gen
11 All Frames Rec
12 MC De-mux
13 VC De-mux
14 VC Reception
2 Apply Security
Return link OCF
17 Process Security
18 FSR
3 FOP
Return link OCF
15 FARM
16 CLCW
7 Encode & randomize
10 Derandomize & decode
NOTE: Processing order of encoding & randomization depends on the coding scheme.
8 Mod
9 De-mod
Time
Legend:
Primary functional flow across Data Link and Physical Layer protocols
(Optional) Function calls within the Data Link Protocol sublayer
(Optional) Operational Control Field return link data flow from receiving end to sending end

Figure 6-2: Order of Processing between USLP, COP, and SDLS Functions

Table 6-1: CCSDS Order of Processing for USLP User Data Frames Using SDLS (Sequential with Figure 6-2)

Numbered Step in Figure 6-2	CCSDS Document	Section Referenced
***On the Ground (sending end)*:**		
1. Virtual Channel Generation Function with SDLS	This document	6.4.2.1
Frame Initialization Procedure including SDLS (Virtual Channel Generation function)	This document	6.4.2.3
2. SDLS ApplySecurity Function	This document	6.4.2.4
Encrypt only the Transfer Frame Data Field (if Encryption or Authenticated Encryption selected)	355.0-B-2	4.2.3.3
Populate the Security Header and the optional Security Trailer with the computed MAC (if Authentication or Authenticated Encryption selected)	355.0-B-2	4.2.3.4
3. FOP (Frame Operation Procedure)	This document	6.4.2.5
Frame Finalization Procedure including SDLS (Virtual Channel Generation function)	This document	6.4.2.6
4. Virtual Channel Multiplexing Function	This document	4.2.8
5. Master Channel Multiplexing Function	This document	4.2.10
6. All Frames Generation Function	This document	4.2.11
Compute and add CRC to FECF	This document	4.2.11.3
7. Encode and Randomize the Transfer Frame (When BCH encoding, Randomization is done first; the opposite for LDPC)	231.0-B-4	Section 3 (BCH) or 4 (LDPC) and section 6
8. Modulate onto Subcarrier/Carrier and transmit	401.0-B-32	2.2
***On the Spacecraft (receiving end)*:**		
9. Receive and Demodulate	401.0-B-32	2.2
10. Derandomize and Decode the Transfer Frame (Note the order is dependent upon the coding scheme)	231.0-B-4	6.3, 3.5, or 4.5
11. All Frames Reception Function with SDLS	This document	6.5.12
Frame Delimiting and Fill Removal Procedure (invalid code blocks reported by C&S Sublayer + fill removal)	This document	4.2.11.2
Frame Validation Check Procedure (includes optional CRC)	This document	4.2.11.3
12. Master Channel Demultiplexing Function	This document	4.3.10
13. Virtual Channel Demultiplexing Function	This document	4.3.8
14. Virtual Channel Reception Function	This document	4.3.7
15. FARM (Frame Acceptance and Reporting Mechanism, subprocedure of the COP)	This document	4.3.7.2
16. CLCW appears within either TM, AOS, or USLP OCF Field	This document	4.3.7.2
17. SDLS ProcessSecurity Function	This document	6.5.2.1 b)
Validate the MAC, if invalid, report security error in Frame Status Report placed into the OCF in telemetry frame (if Authentication or Authenticated Encryption selected)	355.0-B-2	4.2.4.4
Decrypt the Transfer Frame Data Field (if Encryption or Authenticated Encryption selected)	355.0-B-2	4.2.4.5
18. SDLS FSR (Frame Status Report) appears within either TM, AOS, or USLP OCF Field	132.0-B-3 732.0-B-4 This document	4.1.5.5 4.1.5.5 4.1.5.2.2
Thereafter, Transfer Frame Data Units provided to on-board processing (e.g., perform VC or MAP Packet Extraction function or MAP Demultiplexing function or provide to VCA Service User)	N/A	N/A

6.4.3 MAP PACKET PROCESSING FUNCTION WITH SDLS

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

6.4.3.2 When handling MAP Packets on a VC that uses SDLS, the MAP Packet Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAP Packet Processing Function generates a TFDZ to fit exactly within the TFDF (see 4.1.4).

6.4.4 MAPA_SDU GENERATION FUNCTION WITH SDLS

6.4.4.1 The MAPA_SDU Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.3.

6.4.4.2 When handling MAPA_SDUs on a VC that uses SDLS, the MAPA_SDU Generation Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAPA_SDU Generation Function generates a TFDZ to fit exactly within the TFDF (see 4.1.4).

6.4.5 MAP OCTET STREAM PROCESSING FUNCTION WITH SDLS

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

6.4.5.2 When handling Octet Stream Data on a VC that uses SDLS, the MAP Octet Stream Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAP Octet Stream Processing Function generates TFDZs to fit exactly within the TFDF (see 4.1.4).

6.4.6 MAP MULTIPLEXING FUNCTION WITH SDLS

The MAP Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.5.

6.4.7 VC PACKET PROCESSING FUNCTION WITH SDLS

6.4.7.1 The VC Packet Processing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.6.

6.4.7.2 When handling Packets on a VC that uses SDLS, the VC Packet Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

6.4.8 VIRTUAL CHANNEL GENERATION FUNCTION WITH SDLS

6.4.8.1 Discussion

There can be security configurations in which, for example, the physical location of the security processing is not the same for all Virtual Channels, at the sending end or at the receiving end. This case is supported by the order of processing within the Virtual Channel Generation Function. However, the use of multiple Virtual Channels sharing an SDLS Security Association is not supported.

6.4.8.2 Field Lengths and Positions

When assembling a user-data Transfer Frame on a Virtual Channel or MAP that uses SDLS, the Virtual Channel Generation Function shall

a) apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame;

b) conform to the specifications of 4.2.7.

NOTES

1 The lengths of the Security Header and Security Trailer are managed parameters of the Virtual Channel or MAP (see 6.6).

2 The Virtual Channel Generation Function contains the interface to the SDLS protocol. In this case, it calls the SDLS ApplySecurity function for the user data Transfer Frames that it assembles for Virtual Channels or MAPs that use SDLS.

3 The order of processing for the Virtual Channel Generation Function of a USLP protocol entity that supports SDLS is specified in 6.4.2.

6.4.9 VIRTUAL CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The VC Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.8.

6.4.10 MASTER CHANNEL GENERATION FUNCTION WITH SDLS

The Master Channel Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.9.

6.4.11 MASTER CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The MC Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.10.

6.4.12 ALL FRAMES GENERATION FUNCTION WITH SDLS

The All Frames Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.11.

6.5 RECEIVING END PROTOCOL PROCEDURES WITH SDLS

6.5.1 OVERVIEW

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

6.5.2 ORDER OF PROCESSING BETWEEN USLP, COP-1, AND SDLS PROTOCOLS

6.5.2.1 Virtual Channel Reception Function

The order of processing between the functions of USLP, COP, and SDLS protocols shall occur as follows in the Virtual Channel Reception Function at the receiving end:

a) the FARM, which is a subprocedure of the COP and an integral part of the Virtual Channel Reception Function;

b) the SDLS ProcessSecurity Function;

c) the Virtual Channel Reception Function (final step of processing by the function).

6.5.2.2 Discussion

Figure 6-2 above shows the order of processing between USLP, COP, and SDLS protocol functions at both the sending and receiving ends. The receiving side functional flow of the diagram proceeds as follows:

All Frames Reception Function:

- The All Frames Reception Function is the first procedure that receives valid octets from the Channel Coding sublayer. A decoding failure will cause the Frame Delimiting and Fill Removal Procedure within the All Frames Reception Function to discard all the transfer frames contained within a CLTU.
- The Frame Validation Check Procedure within the All Frames Reception Function includes the check of the FECF, if it is present. A transfer frame that fails any of these checks is discarded.

Virtual Channel Reception Function:

- FARM function: the FARM specified in reference [9] or reference [10] and uses the Frame Sequence Number in the Primary Header of an incoming Type-A Transfer Frame to perform the Frame Acceptance Checks. If the checks indicate that the frame is out of sequence, FARM discards the frame.
- SDLS ProcessSecurity function: Depending on the security features in use, the SDLS ProcessSecurity function specified in reference [15] can verify the authenticity of the frame, and it can decrypt the contents of the Transfer Frame Data Field. Any errors detected by the SDLS ProcessSecurity Function are reported to the Virtual Channel Reception Function. The way that Transfer Frame data is passed between the Virtual Channel Reception Function and the SDLS ProcessSecurity Function is implementation dependent.

  If the SDLS ProcessSecurity Function does not report an error, the Virtual Channel Reception Function extracts the Frame Data Unit from the frame and delivers it to its user (or Function). If the SDLS ProcessSecurity Function reports an error, the Virtual Channel Reception Function discards the frame; in this case, the optional Verification Status Code parameter can be used to inform the user of the relevant service. (See 3.3.2.9, 3.4.2.9, 3.5.2.8, and 3.6.3.7.)

NOTES

1 These error conditions are detected either by the FARM and/or by the Frame Validation Check Procedure. To understand the reporting mechanisms with the FARM, one would go either to COP-1 (reference [9]) or COP-P (currently specified in reference [10]) and examine the various frame rejection cases. These are reported in the CLCWs. Additionally and for links without ARQ, frame sequence counters are used to detect transfer frame gaps. For non-ARQ links, gaps are not reported but are detected by the receiving side via the frame sequence counter.

2 Whenever retransmission of Type-AD Transfer Frames or Sequence Controlled U-Frames is required in a series of frames that originally included both Type-AD frames or Sequence Controlled U-Frames and Type-BD or Expedited frames, the SDLS ProcessSecurity Anti-Replay function will reject any retransmitted frames, if authentication is used, based upon their lower anti-replay sequence count in comparison to the Type-BD or Expedited frame anti-replay sequence count (falsely labelling them as SDLS security failures). Therefore, mixing Type-AD/Sequence

Controlled U-Frames and Type-BD/Expedited frames on the same VC secured by SDLS is generally not advised unless Type-BD/Expedited frames are sent only after all Type-AD frames/Sequence Controlled U-Frames have been accepted on board.

6.5.3 MAP PACKET EXTRACTION FUNCTION WITH SDLS

6.5.3.1 The Packet Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.2.

6.5.3.2 When handling Packets on a MAP and VC that uses SDLS, the Packet Extraction Function shall apply the TFDF 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 TFDF (see 4.1.4).

6.5.4 MAPA_SDU EXTRACTION FUNCTION WITH SDLS

6.5.4.1 The MAPA_SDU Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.3.

6.5.4.2 When handling MAPA_SDUs on a MAP and VC that uses SDLS, the MAPA_SDU Extraction Function shall apply the TFDF specification in 6.3.5 to determine the expected length of the TFDZs that it receives.

NOTE – The MAPA_SDU Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).

6.5.5 MAP OCTET STREAM EXTRACTION FUNCTION WITH SDLS

6.5.5.1 The MAP Octet Stream Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.4.

6.5.5.2 When handling MAP Octet Stream Data on a MAP and VC that uses SDLS, the MAP Octet Stream Extraction Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZs that it receives.

NOTE – The MAP Octet Stream Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).

6.5.6 MAP DEMULTIPLEXING FUNCTION WITH SDLS

The MAP Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.5

6.5.7 VC PACKET EXTRACTION FUNCTION WITH SDLS

The VC Packet Extraction Function of a USLP protocol entity that supports SDLS shall

a) apply the Transfer Frame Data Field specification in 6.3.5 to determine the maximum expected length of the Frame Data Units that it receives;

b) conform to the specifications of 4.3.6.

6.5.8 VIRTUAL CHANNEL RECEPTION FUNCTION WITH SDLS

When handling a user-data Transfer Frame on a Virtual Channel or MAP that uses SDLS, the Virtual Channel Reception Function shall

a) apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame;

b) conform to the specifications of 4.3.7.

NOTES

1 The lengths of the Security Header and Security Trailer are managed parameters of the Virtual Channel or MAP (see 6.6).

2 The Virtual Channel Reception Function contains the interface to the SDLS protocol. In this case, it calls the SDLS ProcessSecurity function for the user data Transfer Frames that it handles for Virtual Channels or MAPs that use SDLS.

6.5.9 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The VC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.8.

6.5.10 MASTER CHANNEL RECEPTION FUNCTION WITH SDLS

The MC Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.9.

6.5.11 MASTER CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The MC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.10.

6.5.12 ALL FRAMES RECEPTION FUNCTION WITH SDLS

The All Frames Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.11.

6.6 ADDITIONAL MANAGED PARAMETERS FOR SDLS

6.6.1 The managed parameters associated with a Virtual Channel without Segment Headers when USLP supports the SDLS protocol shall conform to the definitions in table 5-3 and the additional definitions in table 6-2.

Table 6-2: Additional Managed Parameters for a Virtual Channel When the Unified Space Data Link Protocol Supports SDLS

Managed Parameter	Allowed Values
Presence of Space Data Link Security Header	Present (‘1’) / Absent (‘0’)
Presence of Space Data Link Security Trailer	Present (‘1’) / Absent (‘0’)
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, SDLS is in use for the VC, and it applies to all the Transfer Frames with the same GVCID, including truncated Transfer Frames defined in annex D.

2 The valid lengths for the Security Header and Security Trailer are specified in reference [15].

6.6.2 The managed parameters associated with a MAP when USLP supports the SDLS protocol shall conform to the definitions in table 5-4 and the additional definitions in table 6-3.

Table 6-3: Additional Managed Parameters for a MAP When TC 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 MAP.

2 The valid lengths for the Security Header and Security Trailer are specified in reference [15].

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Protocol Implementation Conformance Statement (PICS) Requirements List (RL) for an implementation of the *Unified Space Data Link Protocol* (CCSDS 732.1-B-3). The PICS 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.

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

- the implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;
- a supplier or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
- a user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (it should be noted that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);
- a tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 ABBREVIATIONS AND CONVENTIONS

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

Item Column

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

NOTE – The item-number prefix 'USLP' = 'Data Link Layer'.

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; condition stated below table.
O.<n>	optional, but support of at least one of the group of options labeled by the same numeral <n> is required.
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.

A1.3 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 a PICS. The implementer shall complete the RL by entering appropriate responses in the support or values supported column, using the notation described in A1.2. If a conditional requirement is inapplicable, N/A should be used. If a mandatory requirement is not satisfied, exception information must be supplied by entering a reference X*i*, where *i* is a unique identifier, to an accompanying rationale for the noncompliance.

A2 PICS PROFORMA FOR UNIFIED SPACE DATA LINK PROTOCOL (CCSDS 732.1-B-3)

A2.1 GENERAL INFORMATION

A2.1.1 Identification of PICS

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

A2.1.2 Identification of Implementation Under Test (IUT)

Implementation name	
Implementation version	
Special Configuration	
Other Information	

A2.1.3 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)	

A2.1.4 Identification of Specification

CCSDS 732.1-B-3	
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 PICS, with an explanation of why the implementation is non-conforming.	Yes [ ] No [ ]

A2.2 REQUIREMENTS LIST

Table A-1: USLP Service Data Units

Item	Description	Reference	Status	Support
USLP-1	Packet SDU	3.2.2	M	
USLP-2	MAPA SDU	3.2.3	M	
USLP-3	VCA SDU	3.2.4	M	
USLP-4	Octet Stream SDU	3.2.5	M	
USLP-5	OCF_SDU	3.2.6	M	
USLP-6	USLP Transfer Frame	3.2.7	M	
USLP-7	Insert Data SDU	3.2.8	M	

Table A-2: Service Parameters

Item	Description	Reference	Status	Values Allowed	Support
MAP Packet Service Parameters					
USLP-8	Packet	3.3.2.2	M		
USLP-9	GMAP ID	3.3.2.3	M		
USLP-10	PVN	3.3.2.4	M		
USLP-11	SDU ID	3.3.2.5	M		
USLP-12	QoS	3.3.2.6	M		
USLP-13	Notification Type	3.3.2.7	O	(see ref. [9])	
USLP-14	Packet Quality Indicator	3.3.2.8	O		
USLP-15	Verification Status Code	3.3.2.9	C2	(see ref. [15])	
VCP Service Parameters					
USLP-16	Packet	3.4.2.2	M		
USLP-17	GCVID	3.4.2.3	M		
USLP-18	PVN	3.4.2.4	M		
USLP-19	SDU ID	3.4.2.5	M		
USLP-20	Service Type	3.4.2.6	M		
USLP-21	Notification Type	3.4.2.7	O	(see ref. [9])	
USLP-22	Packet Quality Indicator	3.4.2.8	O		
USLP-23	Verification Status Code	3.4.2.9	C2	(see ref. [15])	
MAPA SDU Service Parameters					
USLP-24	MAPA_SDU	3.5.2.2	M		
USLP-25	GMAP ID	3.5.2.3	M		
USLP-26	SDU ID	3.5.2.4	M		
USLP-27	QoS	3.5.2.5	M		
USLP-28	MAPA_SDU Loss Flag	3.5.2.7	O		
USLP-29	Verification Status Code	3.5.2.8	C2	(see ref. [15])	

Item	Description	Reference	Status	Values Allowed	Support
VCA Service Parameters					
USLP-30	VCA_SDU	3.6.3.2	M		
USLP-31	GVCID	3.6.3.3	M		
USLP-32	SDU ID	3.6.3.4	M		
USLP-33	Service Type	3.6.3.5	M		
USLP-34	Notification Type	3.6.3.6	O	(see ref. [9])	
USLP-35	Verification Status Code	3.6.3.7	C2	(see ref. [15])	
MAP Octet Stream Service Parameters					
USLP-36	Octet Stream Data	3.7.2.2	M		
USLP-37	GMAP ID	3.7.2.3	M		
USLP-38	SDU ID	3.7.2.4	M		
USLP-39	QoS	3.7.2.5	M		
USLP-40	Octet Stream Data Loss Flag	3.7.2.6	O		
USLP-41	Verification Status Code	3.7.2.7	C2	(see ref. [15])	
USLP_MC_OCF Service Parameters					
USLP-42	OCF_SDU	3.8.2.2	M		
USLP-43	GVCID	3.8.2.3	M		
USLP-44	OCF_SDU Loss Flag	3.8.2.4	O		
VCF Service Parameters					
USLP-45	USLP Frame	3.9.2.2	M		
USLP-46	GVCID	3.9.2.3	M		
USLP-47	Frame Loss Flag	3.9.2.4	O		
MCF Service Parameters					
USLP-48	USLP Frame	3.10.2.2	M		
USLP-49	MCID	3.10.2.3	M		
USLP-50	Frame Loss Flag	3.10.2.4	O		
Insert Service Parameters					
USLP-51	IN_SDU	3.11.2.2	M		
USLP-52	Physical Channel Name	3.11.2.3	M		
USLP-53	IN_SDU Loss Flag	3.11.2.4	O		

Item	Description	Reference	Status	Values Allowed	Support
COPs Management Service Parameters					
USLP-54	GVCID	3.12.2.2.1	O.1		
USLP-55	Port ID	3.12.2.2.2	O.1	(see ref. [10] for COP-P)	
USLP-56	Directive ID	3.12.2.3	M		
USLP-57	Directive Type	3.12.2.4	M	(see refs. [9] for COP-1 and [10] for COP-P)	
USLP-58	Directive Qualifier	3.12.2.5	M	(see refs. [9] for COP-1 and [10] for COP-P)	
USLP-59	Notification Type	3.12.2.6	M	(see refs. [9] for COP-1 and [10] for COP-P)	
USLP-60	Notification Qualifier	3.12.2.7	M	(see refs. [9] for COP-1 and [10] for COP-P)	

C2: O if SDLS Option else N/A.

O.1: An Implementation may support none or at least one of these options.

Table A-3: Service Primitives

Item	Description	Reference	Status	Support
MAPP Service Primitives				
USLP-61	MAPP.request	3.3.3.2	M	
USLP-62	MAPP_Notify.indication	3.3.3.3	M	
USLP-63	MAPP.indication	3.3.3.4	M	
VCP Service Primitives				
USLP-64	VCP.request	3.4.3.2	M	
USLP-65	VCP_Notify.indication	3.4.3.3	M	
USLP-66	VCP.indication	3.4.3.4	M	
MAPA Service Primitives				
USLP-67	MAPA.request	3.5.3.2	M	
USLP-68	MAPA_Notify.indication	3.5.3.3	M	
USLP-69	MAPA.indication	3.5.3.4	M	
VCA Service Primitives				
USLP-70	VCA.request	3.6.4.2	M	
USLP-71	VCA_Notify.indication	3.6.4.3	M	
USLP-72	VCA.indication	3.6.4.4	M	
MAP Octet Stream Service Primitives				
USLP-73	OCTET_STREAM.request	3.7.3.2	M	
USLP-74	OCTET_STREAM.indication	3.7.3.3	M	
USLP-75	OCTET_STREAM_Notify.indication	3.7.3.4	M	
USLP_MC_OCF Service Primitives				
USLP-76	USLP_MC_OCF.request	3.8.3.2	M	
USLP-77	USLP_MC_OCF.indication	3.8.3.3	M	
VCF Service Primitives				
USLP-78	VCF.request	3.9.3.2	M	
USLP-79	VCF.indication	3.9.3.3	M	
MCF Service Primitives				
USLP-80	MCF.request	3.10.3.2	M	
USLP-81	MCF.indication	3.10.3.3	M	
Insert Service Primitives				
USLP-82	INSERT.request	3.11.3.2	M	
USLP-83	INSERT.indication	3.11.3.3	M	
COPs Management Service Primitives				
USLP-84	Directive.request	3.12.3.2	M	
USLP-85	Directive_Notify.indication	3.12.3.3	M	
USLP-86	Async_Notify.indication	3.12.3.4	M	

Table A-4: USLP Protocol Data Unit

Item	Description	Reference	Status	Support
USLP-87	USLP Transfer Frame	4.1.1	M	
USLP-88	Transfer Frame Primary Header	4.1.2	M	
USLP-89	Transfer Frame Insert Zone	4.1.3	M	
USLP-90	Transfer Frame Data Field	4.1.4	M	
USLP-91	Operational Control Field	4.1.5	M	
USLP-92	Frame Error Control Field	4.1.6	M	

Table A-5: Protocol Procedures

Item	Description	Reference	Status	Support
USLP-93	MAPP Processing Function	4.2.2	M	
USLP-94	MAPA_SDU Generation Function	4.2.3	M	
USLP-95	MAP Octet Stream Processing Function	4.2.4	M	
USLP-96	MAP Multiplexing Function	4.2.5	M	
USLP-97	Virtual Channel Packet Processing	4.2.6	M	
USLP-98	Virtual Channel Generation Function	4.2.6	M	
USLP-99	Virtual Channel Multiplexing Function	4.2.8	M	
USLP-100	Master Channel Generation Function	4.2.9	M	
USLP-101	Master Channel Multiplexing Function	4.2.10	M	
USLP-102	All Frames Generation Function	4.2.11	M	
USLP-103	MAPP Extraction Function	4.3.2	M	
USLP-104	MAPA_SDU Extraction Function	4.3.3	M	
USLP-105	MAP Octet Stream Extraction Function	4.3.4	M	
USLP-106	MAP Demultiplexing Function	4.3.5	M	
USLP-107	Virtual Channel Packet Extraction Function	4.3.6	M	
USLP-108	Virtual Channel Reception Function	4.3.6	M	
USLP-109	Virtual Channel Demultiplexing Function	4.3.8	M	
USLP-110	Master Channel Reception Function	4.3.9	M	
USLP-111	Master Channel Demultiplexing Function	4.3.10	M	
USLP-112	All Frames Reception Function	4.3.11	M	

Table A-6: Management Parameters

Item	Description	Reference	Status	Values Allowed	Support
Managed Parameters for a Physical Channel					
USLP-113	Physical Channel Name	Table 5-1	M	Character String	
USLP-114	Physical Channel Transfer Frame Type	Table 5-1	M	Fixed Length or Variable Length	
USLP-115	Transfer Frame Length	Table 5-1	M	Integer (C = 4 to 65535 octets) (see 4.1.2.7)	
USLP-116	TFVN	Table 5-1	M	‘1100’ binary	
USLP-117	MC Multiplexing Scheme	Table 5-1	M		
USLP-118	Presence of Insert Zone	Table 5-1	M	Present (‘1’), Absent (‘0’)	
USLP-119	Insert Zone Length (octets)	Table 5-1	M	Integer, 1–65514 octets	
USLP-120	Presence of Frame Error Control	Table 5-1	M	Present (‘1’), Absent (‘0’)	
USLP-121	[Requirement Deleted]				
USLP-122	Maximum Number of Transfer Frames Given to the Coding and Synchronization Sublayer as a single data unit	Table 5-1	M	Integer	
USLP-123	Maximum Value of the Repetitions Parameter to the Coding and Synchronization Sublayer	Table 5-1	M	Integer	
Managed Parameters for a Master Channel					
USLP-124	MC Transfer Frame Type	Table 5-2	M	Fixed Length or Variable Length	
USLP-125	SCID	Table 5-2	M	16-bit Integer	
USLP-126	VCIDs	Table 5-2	M	0 to 62	
USLP-127	VC Multiplexing Scheme	Table 5-2	M		
Managed Parameters for a Virtual Channel					
USLP-128	VC Transfer Frame Type	Table 5-3	M	Fixed Length or Variable Length	
USLP-129	VCID	Table 5-3	M	0 to 62	
USLP-130	VC Count Length for Sequence Control QoS	Table 5-3	M	Integer (max. 56 bit)	
USLP-131	VC Count Length for Expedited QoS	Table 5-3	M	Integer (max. 56 bit)	
USLP-132	COP in Effect	Table 5-3	M	COP-1, COP-P, None	
USLP-133	CLCW Version Number	Table 5-3	M	1	
USLP-134	CLCW Reporting Rate	Table 5-3	M		
USLP-135	MAP IDs	Table 5-3	M	0–15	
USLP-136	MAP Multiplexing Scheme	Table 5-3	O		

Item	Description	Reference	Status	Values Allowed	Support
USLP-137	Truncated Transfer Frame Length	Table 5-3	M	Integer	
USLP-138	SDU Type (Data Field Content)	Table 5-3	M	CCSDS Packet, VCA_SDU	
USLP-139	Inclusion of OCF Allowed (variable-length USLP Frames)	Table 5-3	M	True (‘1’), False (‘0’)	
USLP-140	Inclusion of OCF Required (fixed-length USLP Frames)	Table 5-2	M	True (‘1’), False (‘0’)	
USLP-141	Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying service data on the Sequence-Controlled Service	Table 5-3	M	Integer (see ref. [6])	
USLP-142	Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying COP Control Commands	Table 5-3	M	Integer (see ref. [6])	
USLP-143	Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released	Table 5-3	M	Integer	
USLP-144	Maximum delay in milliseconds between releases of USLP Frames of the same VC	Table 5-3	M	Integer	
Managed Parameters for a MAP Channel					
USLP-145	MAP ID	Table 5-4	M	0–15	
USLP-146	SDU Type	Table 5-4	M	CCSDS Packet, MAPA_SDU, Octet Stream Data	
USLP-147	UPID supported	Table 5-4	M	Integer (see reference [14])	
Managed Parameters for a Packet Transfer					
USLP-148	Valid PVNs	Table 5-5	M	Set of Integers	
USLP-149	Maximum Packet Length (octets)	Table 5-5	M	Integer	
USLP-150	Whether incomplete Packets are required to be delivered to the user at the receiving end	Table 5-5	M	Required, not required	

Table A-7: Protocol Specification with SDLS Option

Item	Description	Reference	Status	Support
USLP-151	SDLS Protocol	(see ref. [15])	O	
USLP-152	Security Header	6.3.4	C3	
USLP-153	Security Trailer	6.3.6	C4	
USLP-154	Transfer Frame Data Field in a USLP Frame with SDLS	6.3.5.2	C3	
USLP-155	Operational Control Field in a USLP Frame with SDLS	6.3.7.2	C3	
USLP-156	Frame Error Control Field in a USLP Frame with SDLS	6.3.8.2	C3	
USLP-157	MAP Packet Processing Function with SDLS	6.4.3	C3	
USLP-158	MAP Octet Stream Processing Function with SDLS	6.4.5	C3	
USLP-159	MAP Multiplexing Function with SDLS	6.4.6	C3	
USLP-160	Virtual Channel Packet Processing with SDLS	6.4.7	C3	
USLP-161	Virtual Channel Generation Function with SDLS	6.4.8	C3	
USLP-162	Virtual Channel Multiplexing Function with SDLS	6.4.9	C3	
USLP-163	Master Channel Generation Function with SDLS	6.4.10	C3	
USLP-164	Master Channel Multiplexing Function with SDLS	6.4.11	C3	
USLP-165	All Frames Generation Function with SDLS	6.4.12	C3	
USLP-166	MAP Packet Extraction Function with SDLS	6.5.3	C3	
USLP-167	MAPA SDU Extraction Function with SDLS	6.5.4	C3	
USLP-168	MAP Octet Stream Extraction Function with SDLS	6.5.5	C3	
USLP-169	MAP Demultiplexing Function with SDLS	6.5.6	C3	
USLP-170	Virtual Channel Packet Extraction Function with SDLS	6.5.7	C3	
USLP-171	Virtual Channel Reception Function with SDLS	6.5.8	C3	
USLP-172	Virtual Channel Demultiplexing Function with SDLS	6.5.9	C3	
USLP-173	Master Channel Reception Function with SDLS	6.5.10	C3	
USLP-174	Master Channel Demultiplexing Function with SDLS	6.5.11	C3	
USLP-175	All Frames Reception Function with SDLS	6.5.12	C3	

C3: M if SDLS Option else N/A.

C4: O if SDLS Option else N/A.

Table A-8: Additional Managed Parameters with SDLS Option

Item	Description	Reference	Status	Values Allowed	Support
USLP-176	Presence of Space Data Link Security Header	Table 6-2	C5	Present ('1') / Absent ('0')	
USLP-177	Presence of Space Data Link Security Trailer	Table 6-2	C5	Present ('1') / Absent ('0')	
USLP-178	Length of Space Data Link Security Header (octets)	Table 6-2	C5	Integer (see ref. [15])	
USLP-179	Length of Space Data Link Security Trailer (octets)	Table 6-2	C5	Integer (see ref. [15])	

C5: M if SDLS Option else N/A.

Table A-9: Frame Error Control Field Coding Procedures

Item	Description	Reference	Status	Support
USLP-180	CRC-16 FECF Encoding Procedure	B1.1	M	
USLP-181	CRC-16 FECF Decoding Procedure	B1.2	M	

Table A-10: Relationship of Version-3 and Version-4 Transfer Frames

Item	Description	Reference	Status	Support
USLP-182	V3-V4 equivalencies	Annex C	C6	

C6: M if gateway else N/A.

ANNEX B

FRAME ERROR CONTROL FIELD CODING PROCEDURES

(NORMATIVE)

B1 CRC-16 FRAME ERROR CONTROL FIELD CODING PROCEDURES

B1.1 CRC-16 FECF ENCODING PROCEDURE

B1.1.1 The FECF shall be computed by applying Cyclic Redundancy Check (CRC) techniques.

B1.1.2 The Encoding Procedure shall accept an (n–16)-bit Transfer Frame, excluding the FECF, and generate a systematic binary (n,n–16) block code by appending a 16-bit FECF as the final 16 bits of the codeword, where n is the length of the Transfer Frame.

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

B1.1.3 The equation for the contents of the FECF is:

$$\begin{aligned} \text{FECF} &= [(X^{16} \cdot \text{M}(X)) + (X^{(n\text{-}16)} \cdot \text{L}(X))] \text{ modulo G}(X) \\ &= \text{P}_0 \cdot X^{15} + \text{P}_1 \cdot X^{14} + \text{P}_2 \cdot X^{13} + \ldots + \text{P}_{14} \cdot X^1 + \text{P}_{15} \cdot X^0 \end{aligned}$$

where

all arithmetic is modulo 2;

FECF is the 16-bit FECF with the first bit transferred being the MSB P_0 taken as the coefficient of the highest power of X;

n is the number of bits in the encoded message;

M(X) is the (n−16)-bit information message to be encoded expressed as a polynomial with binary coefficients, with the first bit transferred being the MSB M_0 taken as the coefficient of the highest power of X;

L(X) is the presetting polynomial given by

$$\text{L}(X) = \sum_{i=0}^{15} X^i \ ;$$

G(X) is the generating polynomial given by

$$\text{G}(X) = X^{16} + X^{12} + X^5 + 1.$$

NOTES

1 The $X^{(n-16)} \cdot \text{L}(X)$ term has the effect of presetting the shift register to all ‘1’ state prior to encoding.

2 A possible FECF generator implementation is shown in figure B-1. For each USLP Frame, the shift register cells are initialized to ‘1’. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the sixteen FECF bits.

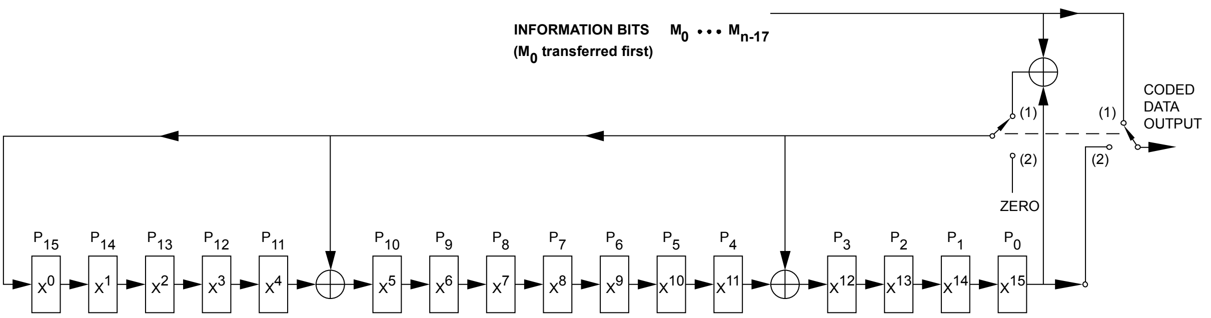


Figure B-1: Logic Diagram of the CRC-16 Encoder

B1.2 CRC-16 FECF DECODING PROCEDURE

B1.2.1 The error detection syndrome, S(*X*), is given by

$$S(X) = [(X^{16} \cdot C^*(X)) + (X^n \cdot L(X))] \text{ modulo } G(X)$$

where

- C*(*X*) is the received block, including the FECF, in polynomial form, with the first bit transferred being the MSB C_0* taken as the coefficient of the highest power of *X*; and
- S(*X*) is the syndrome polynomial, which will be zero if no error is detected and non-zero if an error is detected, with the MSB S_0 taken as the coefficient of the highest power of *X*.

B1.2.2 The received block C*(*X*) must 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, that is, with the MSB C_0 or E_0 taken as the binary coefficient of the highest power of *X*.

NOTE – A possible syndrome polynomial generator implementation is shown in figure B-2. For each USLP Frame, the shift register cells are initialized to ‘1’. The USLP Frame includes *n* bits, that is, (*n*−16) information message bits plus the 16 bits of the FECF. All the *n* bits of the USLP Frame are clocked into the input, and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be ‘zero’. A non-zero content indicates an erroneous block.

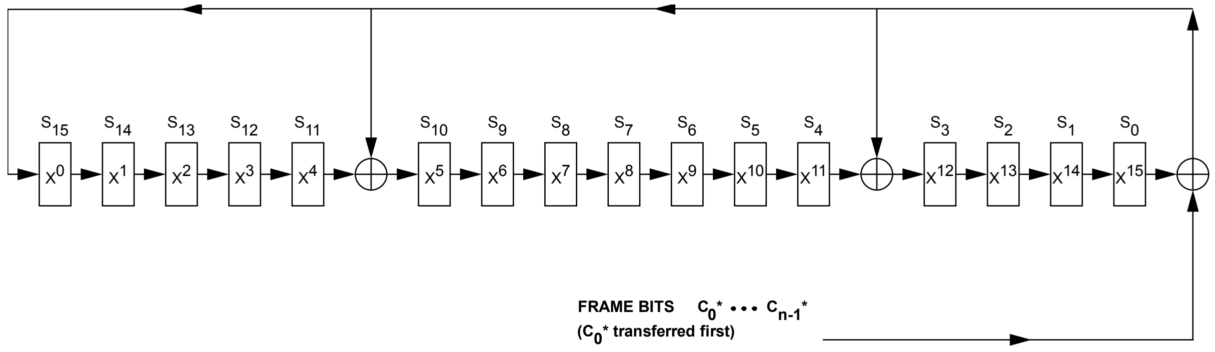


Figure B-2: Logic Diagram of the CRC-16 Decoder

ANNEX C

RELATIONSHIP OF PROXIMITY-1 (VERSION-3) TO USLP (VERSION-4) TRANSFER FRAMES

(NORMATIVE)

C1 TRANSFER FRAME PRIMARY HEADER

C1.1 GENERAL

Version-3 Transfer Frame header fields shall be mapped to Version-4 Transfer Frame primary header fields as specified in table C-1 below.

Table C-1: Relationship of Transfer Frame Header Fields between Version-3 and Version-4 Frames

Version-3 Frame (Proximity-1)		Version-4 Frame (USLP)	
Field Name	**Location in Frame**	**Field Name**	**Location in Frame**
TFVN	Bits 0–1	TFVN	Bits 0–3
QoS Indicator	Bit 2	Bypass Flag	Bit 48
PDU Type ID	Bit 3	Protocol Control Command Flag	Bit 49
Data Field Construction ID	Bits 4–5	TFDZ Construction Rules in TFDF Header	(see table C-2)
Spacecraft Identifier	Bits 6–15	Spacecraft Identifier	Bits 4–19
Physical Channel Identifier (PCID)	Bit 16	VCID	Use only Bit 21
N/A	N/A	VCID	Use Bits 22–26
Port ID	Bits 17–19	MAP ID	Use only Bits 28–30
Source-or-Destination Identifier	Bit 20	Source-or-Destination Identifier	Bit 20
Frame Length	Bits 21–31	Frame Length	Bits 32–47
Frame Sequence Number	Bits 32–39	Virtual Channel Frame Count	Bits 56–63
N/A	N/A	End of Frame Primary Header Flag	Bit 31
N/A	N/A	OCF Flag	Bit 52
N/A	N/A	VC Frame Count Length	Bit 53–55

C1.2 TRANSFER FRAME VERSION NUMBER

NOTE – The Version-3 TFVN has been extended by two bits to allow for future versions.

The USLP Transfer Frame shall use the last remaining value available in the 2-bit TFVN field (i.e., ‘11’) and append ‘00’ to it to complete the 4-bit Version-4 TFVN (‘*1100*’).

C1.3 QUALITY OF SERVICE INDICATOR

The Bypass Flag in bit 48 of the Version-4 Transfer Frame Primary Header shall be equivalent to the QoS Indicator field in bit 2 of the Version-3 Frame.

C1.4 PDU TYPE ID

The Protocol Control Command Flag in bit 49 of the Version-4 Transfer Frame Primary Header shall be equivalent to the PDU Type Indicator in bit 3 of the Version-3 Frame.

C1.5 DATA FIELD CONSTRUCTION ID

The TFDZ Construction Rules in bits 1–3 of the Version-4 TFDF Header shall replace the functionality of both the Data Field Construction ID (DFC ID) in bits 4–5 of the Version-3 Frame as well as the Segment Header Sequence Flags defined in the first two bits of the Proximity-1 Segment Header within the Version-3 TFDF. In addition, for segmented data only, the Version-4 UPID field (5 bits) shall replace the functionality of the Proximity-1 Pseudo Packet ID, which contains the temporary packet ID assigned by Proximity-1 to a user’s packet during segmentation. Reference [14] is the SANA registry of CCSDS-recognized protocol IDs. (Table C-2 below relates the contents of the Version-3 TFDF to the Version-4 TFDZ.)

Table C-2: TFDZ Construction Rules

Version-3 DFC ID	Version-3 DFC ID Interpretation	Version-3 Segment Header Sequence Flags	Version-4 TFDZ Construction Rule Equivalent	Version-4 TFDZ Construction Rule Interpretation
‘00’	Packets (integer number of unsegmented packets)	N/A	‘111’	No Segmentation
‘01’	Segment Data (a complete or segmented packet or user-defined SDU)	‘00’ ‘01’ ‘10’ ‘11’	‘100’ ‘101’ ‘110’ ‘111’	Starting Segment Continuing Segment Last Segment No Segmentation
‘10’	Reserved for future CCSDS definition	N/A	N/A	N/A
‘11’	User Defined Data (octet stream)	N/A	‘011’	Octet Stream

C1.6 SPACECRAFT IDENTIFIER

The SCID in bits 4–19 of the Version-4 Transfer Frame Primary Header shall replace the SCID in bits 6–15 of the Version-3 Transfer Frame.

NOTE – The SCID field in the Version-4 Frame is specified as a 16 bit field to accommodate the increase in SCID requests for future missions.

C1.7 PHYSICAL CHANNEL ID

The MSB of the VCID field (bit 21) of the Version-4 Transfer Frame Primary Header shall be used for the PCID in the Version-3 Frame header.

NOTES

1 The VCID in the Version-4 Transfer Frame is a 6-bit field, but only the first bit of this field is utilized for the PCID.

2 The use of a separate VCID for Expedited Frames eliminates the need to have two separate VCF Counters for the PCID.

C1.8 PORT IDENTIFIER (PORT ID)

The MAP ID in bits 28–30 of the Version-4 Transfer Frame Primary Header shall replace the Port ID in bits 17–19 of the Version-3 Frame.

C1.9 SOURCE/DESTINATION ID

The source or destination ID in bit 20 of the Version-4 Transfer Frame Primary Header shall be equivalent to the Source/Destination ID in bit 20 of the Version-3 Frame.

C1.10 DISCUSSION—FRAME LENGTH

The Frame Length Field in the Version-4 Transfer Frame Primary Header is 16 bits in length and is located in bits 32–47. The Frame Length Field in the Version-3 Frame is 11 bits in length and located in bits 21–31. The Version-4 Frame can support Frame lengths up to 65536 octets.

C1.11 VIRTUAL CHANNEL FRAME COUNT LENGTH

For Proximity-1 operations, the value of the VCF Count Length Field in the Version-4 Transfer Frame Primary Header shall be set to ‘001’, that is, one octet.

C1.12 FRAME SEQUENCE NUMBER

The VCF Count in bits 56–63 of the Version-4 Transfer Frame Primary Header shall replace the Frame sequence counter in bits 32–39 of the Version-3 Frame.

C1.13 TRUNCATED TRANSFER FRAME PRIMARY HEADER

The truncated USLP Transfer Frame shall be created and signaled by setting the End of Frame Primary Header Flag (bit 31) to ‘1’.

NOTE – This option is provided mainly for transmitting short length telecommands. Since the USLP Primary Transfer Frame Header is truncated and contains no length field, the frame length is defined by the VC managed parameter ‘Truncated Transfer Frame Length’.

C2 DISCUSSION—INSERT ZONE

The Insert Zone is specified for use with fixed-length USLP Frames of constant length that are aligned with fixed-length codeblocks. Thus the use of the Insert Zone in proximity operations as well as in truncated Transfer Frames is not allowed (see annex D).

C3 TRANSFER FRAME DATA FIELD

C3.1 OVERVIEW

The TFDF Header for Proximity operations utilizes a subset of the fields defined in USLP. Because Proximity-1 is defined to use variable-length USLP Frames exclusively, the optional USLP First Header/Last Valid Octet Pointer Field is not used in the TFDF Header. Thus the TFDF Header reduces to a header composed of two fields immediately followed by the TFDZ.

TFDZ Construction Rules	USLP Protocol Identifier
3 bits	5 bits

Figure C-1: Transfer Frame Data Field Header for Proximity Operations

NOTE – The First Header/Last Valid Octet Pointer Field (see figure 4-4) is not applicable for Proximity operations.

C3.2 TFDF HEADER

C3.2.1 General

For proximity operations, the TFDF Header shall utilize the first two fields of the TFDF defined in USLP:

a) the TFDZ Construction Rules, 3 bits;

b) UPID, 5 bits.

C3.2.2 TFDZ Construction Rules

The TFDZ Construction Rules that apply to Proximity-1 operations shall be those specified in table C-2 above.

NOTE – The TFDZ Construction Rules for proximity operations utilizes a subset of the eight rules defined in USLP. Three of these rules are defined exclusively for fixed-length USLP Frames and thus are not applicable for Proximity-1 operations. Segmentation for Proximity operations is accomplished by using the TFDZ Construction Rules. Therefore the Segment Header as defined in Proximity-1 as the first octet of the TFDF is not used.

C3.2.3 USLP Protocol Identifier

The UPID applicable to Proximity-1 operations shall be a value contained in the SANA UPID registry (reference [14]).

NOTE – Values ‘00001’ and ‘11111’ are not applicable to Proximity-1 operations.

C4 DISCUSSION—SECURITY HEADER AND TRAILER

The presence of the Security Header and Security Trailer is controlled by the USLP VC managed parameters. It is anticipated that these managed parameters will be fixed for a mission. Because USLP defines the VCID field, and because the MSB of the VCID field for Proximity-1 operations is defined as the PCID equivalent (see C1.7), there are only 32 VCIDs defined for Proximity Link operations over USLP.

NOTE – The VCID associated with a specific spacecraft can be set for the mission and different spacecraft could be assigned different VCIDs so that they could utilize very different SAs as defined in SDLS.

C5 OCF

The OCF may be used to transfer either a PLCW or an SDLS FSR.

NOTES

1 The OCF Flag in bit 52 of the Version-4 Transfer Frame Primary Header signals the presence or absence of the OCF field in the USLP Frame. The use of the OCF is optional.

2 For Proximity-1 operations using the Version-4 Frame, a PLCW or FSR can be transferred either

- as an SPDU sent in a separate P-frame (Proximity-1 protocol control frame); or
- as user data within the OCF.

3 The PLCW or FSR can be sent in the OCF field of a Transfer Frame that does not contain a TFDF.

C6 FECF

Since Proximity-1 Synchronization and Channel Coding (reference [7]) appends a CRC-32 to the PLTU, the functionality of FECF is not strictly needed. When the FECF is omitted, the error protection is mandatorily supplied by the C&S Sublayer, which appends the single CRC-32 to the Proximity-1 PLTU. When Proximity-1 coding (reference [7]) is used, the FECF may still be present but no check is required by the C&S Sublayer.

ANNEX D

TRUNCATED TRANSFER FRAME

(NORMATIVE)

D1 TRUNCATED TRANSFER FRAME PDU

D1.1 OVERVIEW

The truncated Transfer Frame Protocol Data Unit is provided for transmitting short-length forward-link (reference [10]) or direct-from-Earth telecommands (reference [F6]). There is no current expectation that these frames will be used on the Direct to Earth link for cross-support.

D1.2 GENERAL REQUIREMENTS

D1.2.1 The truncated Transfer Frame shall only be allowed when the managed parameter ‘VC Transfer Frame Type’ is set to ‘Variable Length’.

NOTE – All truncated Transfer Frames for a given Virtual Channel ID will have the same length.

D1.2.2 The Truncated USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

a) truncated Transfer Frame Primary Header (4 octets; mandatory);

b) Transfer Frame Data Field (integer number of octets; mandatory).

NOTES

1 The truncated Transfer Frame does not contain an Insert Zone, OCF, or FECF.

2 The format of the truncated Transfer Frame without SDLS and with SDLS is shown in figure D-1.

D1.2.3 When SDLS is in use, the Truncated USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

a) truncated Transfer Frame Primary Header (4 octets; mandatory);

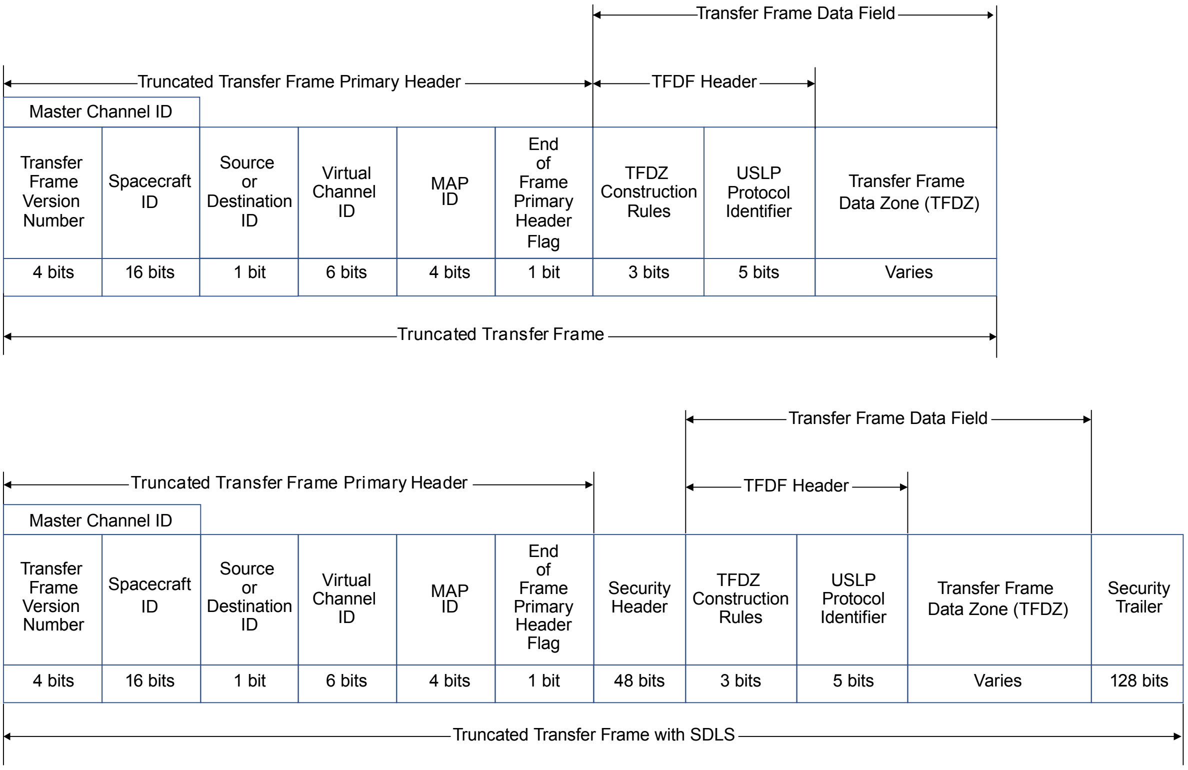
b) SDLS Security Header (6 octets; mandatory);

c) Transfer Frame Data Field (integer number of octets; mandatory);

d) SDLS Security Trailer (16 octets; mandatory).

D1.2.4 SDLS over Truncated USLP Transfer Frames shall only be allowed when the VC managed parameter 'Truncated Transfer Frame Length' is at least 28 octets.

NOTE – SDLS Security Association(s) for use with Virtual Channels that contain truncated Transfer Frames should use a different Security Association (see reference [15]) than that used for Virtual Channels containing non-truncated Transfer Frames.


Transfer Frame Data Field
Truncated Transfer Frame Primary Header
TFDF Header
Master Channel ID
Truncated Transfer Frame
Transfer Frame Data Field
Truncated Transfer Frame Primary Header
TFDF Header
Master Channel ID
Truncated Transfer Frame with SDLS

Transfer Frame Version Number	Spacecraft ID	Source or Destination ID	Virtual Channel ID	MAP ID	End of Frame Primary Header Flag	TFDZ Construction Rules	USLP Protocol Identifier	Transfer Frame Data Zone (TFDZ)
4 bits	16 bits	1 bit	6 bits	4 bits	1 bit	3 bits	5 bits	Varies

Transfer Frame Version Number	Spacecraft ID	Source or Destination ID	Virtual Channel ID	MAP ID	End of Frame Primary Header Flag	Security Header	TFDZ Construction Rules	USLP Protocol Identifier	Transfer Frame Data Zone (TFDZ)	Security Trailer
4 bits	16 bits	1 bit	6 bits	4 bits	1 bit	48 bits	3 bits	5 bits	Varies	128 bits

Figure D-1: USLP Truncated Transfer Frame without SDLS and with SDLS

D1.2.5 Truncated and non-truncated USLP Transfer Frames shall not be mixed together within the same CLTU.

D1.3 TRUNCATED TRANSFER FRAME PRIMARY HEADER

D1.3.1 The truncated Transfer Frame Primary Header is mandatory and shall consist of 6 fields, positioned contiguously, in the following sequence:

a) TFVN (4 bits);

b) SCID (16 bits);

c) Source or Destination Identifier (1 bit);

d) VCID (6 bits);

e) MAP ID (4 bits);

f) End of Frame Primary Header Flag (1 bit).

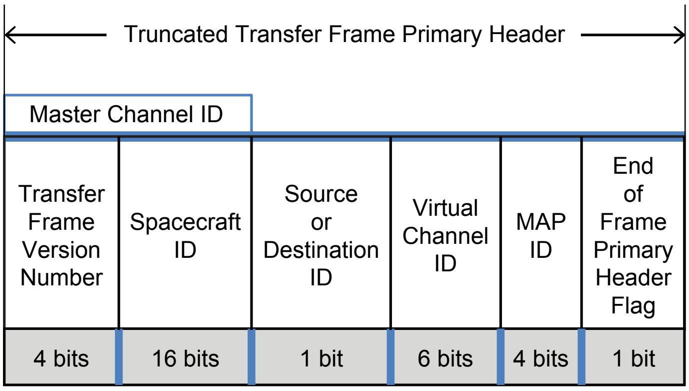
NOTES

1 For backward compatibility with the lengths of the Transfer Frame primary headers of both the Telecommand Space Data Link Protocol (reference [F6]) and the Proximity-1 Space Data Link Protocol (reference [10]), the sum of the USLP Primary and Data Field Headers is set to 5 octets.

2 The fields of the truncated Transfer Frame Primary Header are identical to the first 6 fields of the non-truncated Transfer Frame Primary Header (see 4.1.2 for definition of the fields).

3 Since the truncated Transfer Frame does not contain either a Bypass/Sequence Control Flag or a Protocol Control Flag, these Transfer Frames are to be processed as expedited user data frames. The TFDFs of these truncated Transfer Frames carry user data, with Frame Acceptance Checks bypassed under control of the FARM. These truncated frames use the BD (Expedited) Service of the COP(1/ P). (See table 4-1.)

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



Transfer Frame Version Number	Spacecraft ID	Source or Destination ID	Virtual Channel ID	MAP ID	End of Frame Primary Header Flag
4 bits	16 bits	1 bit	6 bits	4 bits	1 bit

Figure D-2: Truncated Transfer Frame Primary Header

D1.3.2 For each VCID, the length of the truncated Transfer Frame shall be defined by the VC managed parameter ‘Truncated Transfer Frame Length’.

D1.3.2.1 This integer value shall contain a length count that equals the total number of octets in the truncated Transfer Frame.

D1.3.2.2 The count shall be measured from the first octet of the truncated Transfer Frame Primary Header to the last octet of the Transfer Frame Data Field.

NOTES

1 The truncated Transfer Frame itself does not contain a length field.

2 The minimum length of the truncated Transfer Frame is 6 octets and is derived from the length of the truncated Transfer Frame headers and a single Transfer Frame data zone octet.

3 The maximum length truncated Transfer Frame is 32 octets and is derived from the complete frame fitting within one LDPC(512, 256) codeword as defined in reference [6].

D1.4 TRUNCATED TRANSFER FRAME DATA FIELD

D1.4.1 Transfer Frame Data Field Header

D1.4.1.1 The TFDF Header shall follow, without gap, the truncated Transfer Frame Primary Header (see figure D-1).

D1.4.1.2 The TFDF Header shall consist of 1 octet and contain the following fields:

a) TFDZ Construction Rules (3 bits, mandatory);

NOTE – Value equals '111' (No Segmentation).

b) UPID (5 bits, mandatory);

NOTE – Value equals 'Mission Specific Information-1' as one complete MAPA_SDU contained within the TFDZ (see SANA https://sanaregistry.org/r/uslp_protocol_id).

NOTE – The optional First Header/Last Valid Octet Pointer field is not present in truncated TFDF headers.)

D1.4.2 Transfer Frame Data Zone (TFDZ)

D1.4.2.1 The TFDZ shall follow, without gap, the TFDF Header.

D1.4.2.2 The data contained within any given TFDZ shall be associated with one and only one GMAP ID and UPID.

D1.4.2.3 The TFDZ shall contain the data defined by the UPID.

D1.4.2.4 The length of the TFDZ shall be an integral number of octets and may vary in length up to a maximum of 27 octets.

ANNEX E

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

E1 SECURITY CONSIDERATIONS

Security Considerations at the Data Link Layer are addressed by the *Space Data Link Security Protocol* (reference [15]).

E2 SANA CONSIDERATIONS

E2.1 GENERAL

The recommendations in this document have created or modified the following SANA registries located at http://sanaregistry.org.

E2.2 NEW REGISTRY

E2.2.1 USLP Protocol Identifier

UPID = ‘00000’ to ‘11111’

Status = Managed

Reference: CCSDS 732.1-B-3

E2.3 MODIFIED REGISTRIES

E2.3.1 CCSDS Abbreviations

Abbreviation: USLP – Unified Space Data Link Protocol

Reference: CCSDS 732.1-B-3

E2.3.2 CCSDS Terms

isochronous: Characterized by occurrence at equal intervals of time.

E2.3.3 MAP ID

For TFVN = 4, the MAP ID is constrained to values ‘0000’ – ‘1111’.

E2.3.4 Spacecraft Identifier

Version 4 (VN = 1100); SCID = 16 bits; GSCID = VN.SCID = 20 bits.

NOTE – The term GSCID used in reference [11] is actually the same as the MCID defined in the main body of this document.

Version 4 Range: SCID = [0x0000 – 0xFFFF]; GSCID = [C0000 – CFFFF].

E2.3.5 Transfer Frame Version Number

Version Number = ‘1100’; Transfer Frame = USLP Version 4; Reference = CCSDS 732.1-B-3

E2.3.6 VCID

Addition of USLP Protocol to the registry for VCID:

VCID = 000000 – 111111

Status: Unmanaged

Reference: CCSDS 732.1-B-3

E3 PATENT CONSIDERATIONS

At the time of publication, CCSDS was not aware of any claimed patent rights applicable to implementing the provisions of this Recommended Standard.

ANNEX F

INFORMATIVE REFERENCES

(INFORMATIVE)

[F1] *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.

[F2] *Overview of Space Communications Protocols*. Issue 4. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-4. Washington, D.C.: CCSDS, April 2023.

[F3] *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.

[F4] *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.

[F5] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2-S. Recommendation for Space Data System Standards (Historical), CCSDS 910.4-B-2-S. Washington, D.C.: CCSDS, (October 2005) December 2023.

[F6] *TC Space Data Link Protocol*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-4. Washington, D.C.: CCSDS, October 2021.

[F7] *The Application of Security to CCSDS Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-3. Washington, D.C.: CCSDS, March 2019.

[F8] *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.

[F9] *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.

[F10] *Performance Characteristics for Intermediate Data Rate Digital Carriers Using Convolutional Encoding/Viterbi Encoding*. Rev. 10. IESS 308. Washington, DC: INTELSAT, 2000.

[F11] *Overview of the Unified Space Data Link Protocol*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 700.1-G-1. Washington, D.C.: CCSDS, June 2020.

[F12] *Advanced Orbiting Systems, Networks and Data Links: Summary of Concept, Rationale and Performance*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 700.0-G-3. Washington, D.C.: CCSDS, November 1992.

[F13] *TC Synchronization and Channel Coding—Summary of Concept and Rationale*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 230.1-G-3. Washington, D.C.: CCSDS, October 2021.

[F14] *Proximity-1 Space Link Protocol—Rationale, Architecture, and Scenarios*. Issue 2. Report Concerning Space Data System Standards (Green Book), CCSDS 210.0-G-2. Washington, D.C.: CCSDS, December 2013.

[F15] *Telecommand Summary of Concept and Rationale*. Issue 6. Report Concerning Space Data System Standards (Green Book), CCSDS 200.0-G-6. Washington, D.C.: CCSDS, January 1987.

[F16] *TM Synchronization and Channel Coding—Summary of Concept and Rationale*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.1-G-3. Washington, D.C.: CCSDS, June 2020.

[F17] *Next Generation Uplink*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 230.2-G-1. Washington, D.C.: CCSDS, July 2014.

[F18] *Space Data Link Protocols—Summary of Concept and Rationale*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.2-G-3. Washington, D.C.: CCSDS, September 2015.

[F19] *TM Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-3. Washington, D.C.: CCSDS, October 2021.

[F20] *AOS Space Data Link Protocol*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-4. Washington, D.C.: CCSDS, October 2021.

[F21] *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.

[F22] *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 – Normative references are listed in 1.7.

ANNEX G

ABBREVIATIONS AND ACRONYMS

(INFORMATIVE)

This annex lists the abbreviations used in this Recommended Standard.

Term	Meaning
AOS	Advanced Orbiting Systems
APID	application process identifier
ARQ	automatic repeat queuing
ASM	attached sync marker
CCSDS	Consultative Committee for Space Data Systems
CLCW	communications link control word
CLTU	communications link transmission unit
COP-1	communications operation procedure 1
COP-P	communications operation procedure Proximity
DFC ID	data field construction ID
DVB-S2	Digital Video Broadcasting—Satellite—Second Generation
FARM	frame acceptance and reporting mechanism
FARM-B	frame acceptance and reporting mechanism expedited service
FARM-P	frame acceptance and reporting mechanism-Proximity
FECF	frame error control field
FHP	first header pointer
FOP	frame operation procedure
FSR	frame security report
GMAP	global multiplexer access point
GMAP ID	global multiplexer access point identifier
GSCID	global spacecraft identifier
GVCID	global virtual channel identifier
IESS	Intelsat Earth Station Standards
IN_SDU	insert service data unit
LFSR	linear feedback shift register

Term	Meaning
MAP	multiplexer access point
MAP ID	multiplexer access point identifier
MAPA	multiplexer access point access
MAPA_SDU	multiplexer access point access service data unit
MAPP	multiplexer access point packet
MC	master channel
USLP_MC_OCF	master channel operational control field
MCF	master channel frame
MCID	master channel identifier
MSB	most significant bit
OCF	operational control field
OCF_SDU	operational control field service data unit
OID	only idle data (Transfer Frame)
OSI	Open Systems Interconnection
PCID	Physical Channel identifier
PDU	protocol data unit
PICS	protocol implementation conformance statement
PLCW	Proximity link control word
PN	pseudo noise
PVN	packet version number
QoS	quality of service
RF	radio frequency
SA	security association
SANA	Space Assigned Numbers Authority
SAP	service access point
SCCC	Serially Concatenated Convolutional Code
SCID	spacecraft identifier
SDLS	Space Data Link Security
SDU	service data unit
SLP	space link protocol
SPDU	supervisory protocol data unit

Term	Meaning
Sync	synchronization
TC	telecommand
TFDF	Transfer Frame data field
TFDZ	Transfer Frame data zone
TFVN	Transfer Frame version number
TM	telemetry
UPID	USLP protocol identifier
USLP	Unified Space Data Link Protocol
VC	virtual channel
VCA	virtual channel access
VCF	virtual channel frame
VCID	virtual channel identifier
VCP	virtual channel packet
VN	version number

ANNEX H

OID FRAME RANDOMIZATION IMPLEMENTATION OPTIONS

(INFORMATIVE)

Two possible implementations of the OID Frame Randomizer are shown in figures H-1 and H-2. For each frame, the LFSRs are initialized at device start-up as follows:

a) with an all-one seed when the generator in figure H-1 (Fibonacci form) is used; and

b) with a ‘00000000001111111111111111111101’ seed when the generator in figure H-2 (Galois form) is used.

In both cases LFSRs are not restarted for subsequent OID Transfer Frames.

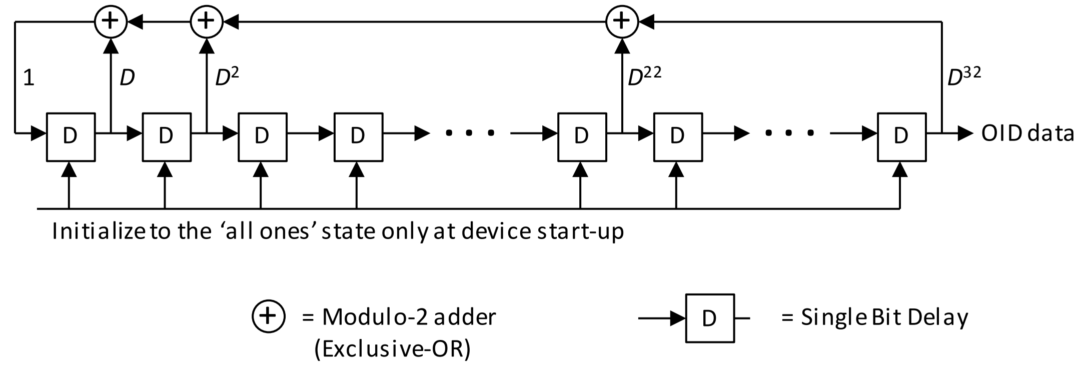

1
D
D2
D22
D32
D
OID data
Initialize to the ‘all ones’ state only at device start-up
= Modulo-2 adder (Exclusive-OR)
= Single Bit Delay

Figure H-1: OID Data Generation Logic Diagram (Fibonacci Form)

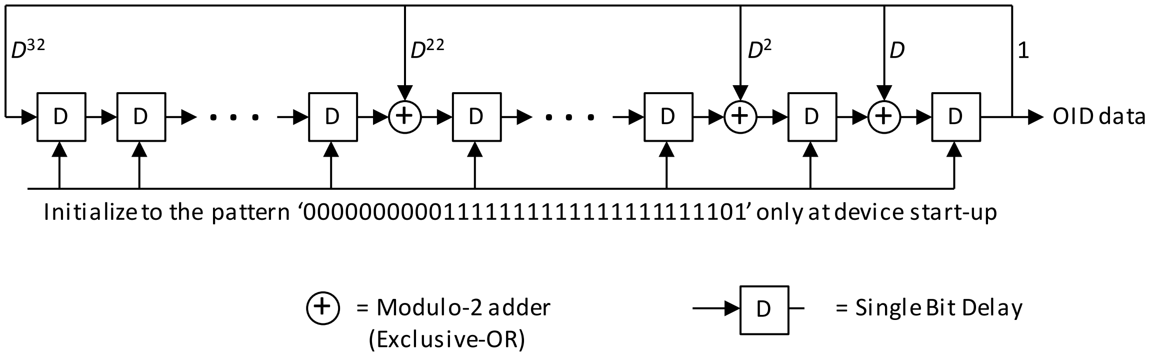

D32
D22
D2
D
1
D
OID data
Initialize to the pattern ‘00000000001111111111111111111101’ only at device start-up
= Modulo-2 adder (Exclusive-OR)
= Single Bit Delay

Figure H-2: OID Data Generation Logic Diagram (Galois Form)

Generated data pattern in both cases:

FF FF FF FF 6D B6 D8 61 45 1F 11 F1 97 16 72 3C BE 7E 00 B1 . . .