CCSDS Historical Document

This document's Historical status indicates that it is no longer current. It has either been replaced by a newer issue or withdrawn because it was deemed obsolete. Current CCSDS publications are maintained at the following location:

http://public.ccsds.org/publications/

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

TC SPACE DATA LINK PROTOCOL

RECOMMENDED STANDARD

CCSDS 232.0-B-3

BLUE BOOK

September 2015

CCSDS HISTORICAL DOCUMENT

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

TC SPACE DATA LINK PROTOCOL

RECOMMENDED STANDARD

CCSDS 232.0-B-3

BLUE BOOK

September 2015

AUTHORITY

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

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

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

STATEMENT OF INTENT

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

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

- 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.
- 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.
- Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

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

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

FOREWORD

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

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

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

http://www.ccsds.org/

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

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

Member Agencies

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

Observer Agencies

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

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 232.0-B-1	TC Space Data Link Protocol, Recommended Standard, Issue 1	September 2003	Original Issue, superseded
CCSDS 232.0-B-2	TC Space Data Link Protocol, Recommended Standard, Issue 2	September 2010	Issue 2, superseded
CCSDS 232.0-B-3	TC Space Data Link Protocol, Recommended Standard, Issue 3	September 2015	Current issue: – adds specifications to support the Space Data Link Security Protocol; – removes obsolete informative annex detailing changes from Historical Recommendation CCSDS 202.0-B-3-S (1987–2005).

NOTE – Substantive changes from the previous issue are marked by change bars in the inside margin.

CONTENTS

Section Page

1 INTRODUCTION 1-1

1.1 PURPOSE 1-1
1.2 SCOPE 1-1
1.3 APPLICABILITY 1-1
1.4 RATIONALE 1-2
1.5 DOCUMENT STRUCTURE 1-2
1.6 CONVENTIONS AND DEFINITIONS 1-2
1.7 REFERENCES 1-5

2 OVERVIEW 2-1

2.1 CONCEPT OF TC SPACE DATA LINK PROTOCOL 2-1
2.2 OVERVIEW OF SERVICES 2-5
2.3 OVERVIEW OF FUNCTIONS 2-12
2.4 SERVICES ASSUMED FROM LOWER LAYERS 2-16

3 SERVICE DEFINITION 3-1

3.1 OVERVIEW 3-1
3.2 SOURCE DATA 3-1
3.3 MAP PACKET SERVICE 3-3
3.4 VIRTUAL CHANNEL PACKET SERVICE 3-9
3.5 MAP ACCESS SERVICE 3-15
3.6 VIRTUAL CHANNEL ACCESS SERVICE 3-20
3.7 VIRTUAL CHANNEL FRAME SERVICE 3-25
3.8 MASTER CHANNEL FRAME SERVICE 3-28
3.9 COP MANAGEMENT SERVICE 3-31

4 PROTOCOL SPECIFICATION WITHOUT SDLS OPTION 4-1

4.1 PROTOCOL DATA UNIT (TC TRANSFER FRAME) 4-1
4.2 PROTOCOL DATA UNIT (CLCW) 4-12
4.3 PROTOCOL PROCEDURES AT THE SENDING END 4-18
4.4 PROTOCOL PROCEDURES AT THE RECEIVING END 4-28

5 MANAGED PARAMETERS WITHOUT SDLS OPTION 5-1

5.1 MANAGED PARAMETERS FOR A PHYSICAL CHANNEL 5-1
5.2 MANAGED PARAMETERS FOR A MASTER CHANNEL 5-2
5.3 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL 5-2

CONTENTS (continued)

Section Page

5.4 MANAGED PARAMETERS FOR A MAP CHANNEL 5-4
5.5 MANAGED PARAMETERS FOR PACKET TRANSFER 5-4

6 PROTOCOL SPECIFICATION WITH SDLS OPTION 6-1

6.1 OVERVIEW 6-1
6.2 USE OF SDLS PROTOCOL 6-1
6.3 TC TRANSFER FRAME WITH SDLS 6-1
6.4 SENDING-END PROTOCOL PROCEDURES WITH SDLS 6-5
6.5 RECEIVING-END PROTOCOL PROCEDURES WITH SDLS 6-8
6.6 ADDITIONAL MANAGED PARAMETERS FOR SDLS 6-11

ANNEX A ACRONYMS (INFORMATIVE) A-1
ANNEX B INFORMATIVE REFERENCES (INFORMATIVE) B-1

Figure

1-1 Bit Numbering Convention 1-5
2-1 Relationship with OSI Layers 2-1
2-2 Relationships Between Channels 2-4
2-3 Internal Organization of Protocol Entity (Sending End) 2-13
2-4 Internal Organization of Protocol Entity (Receiving End) 2-13
2-5 TC Space Data Link Protocol Channel Tree 2-14
4-1 TC Transfer Frame Structural Components 4-1
4-2 Transfer Frame Primary Header 4-2
4-3 Segment Header 4-7
4-4 Logic Diagram of the Encoder 4-10
4-5 Logic Diagram of the Decoder 4-11
4-6 Communications Link Control Word 4-13
4-7 Internal Organization of Protocol Entity (Sending End) 4-18
4-8 Abstract Model of MAP Packet Processing Function 4-20
4-9 Example of MAP Packet Processing Procedures 4-20
4-10 Abstract Model of MAP Generation Function 4-21
4-11 Example of MAP Generation Procedures 4-21
4-12 Abstract Model of MAP Multiplexing Function 4-22
4-13 Abstract Model of VC Packet Processing Function 4-23
4-14 Example of VC Packet Processing Procedures 4-23
4-15 Abstract Model of Virtual Channel Generation Function 4-24
4-16 Abstract Model of Virtual Channel Multiplexing Function 4-25
4-17 Abstract Model of Master Channel Multiplexing Function 4-26

CONTENTS (continued)

Figure Page

4-18 Abstract Model of All Frames Generation Function 4-27
4-19 Internal Organization of Protocol Entity (Receiving End) 4-28
4-20 Abstract Model of MAP Packet Extraction Function 4-29
4-21 Abstract Model of MAP Reception Function 4-30
4-22 Abstract Model of MAP Demultiplexing Function 4-30
4-23 Abstract Model of VC Packet Extraction Function 4-31
4-24 Abstract Model of Virtual Channel Reception Function 4-32
4-25 Abstract Model of Virtual Channel Demultiplexing Function 4-33
4-26 Abstract Model of Master Channel Demultiplexing Function 4-34
4-27 Abstract Model of All Frames Reception Function 4-35
6-1 SDLS Fields in a Type-D Transfer Frame with a Segment Header 6-2
6-2 SDLS Fields in a Type-D Transfer Frame without a Segment Header 6-2
6-3 Order of Processing between TC, COP-1, and SDLS Functions 6-6

Table

2-1 Summary of Services Provided by TC Space Data Link Protocol 2-8
4-1 Interpretation of the Bypass and Control Command Flags 4-4
4-2 Interpretation of the Sequence Flags 4-8
5-1 Managed Parameters for a Physical Channel 5-1
5-2 Managed Parameters for a Master Channel 5-2
5-3 Managed Parameters for a Virtual Channel 5-3
5-4 Managed Parameters for a MAP Channel 5-4
5-5 Managed Parameters for Packet Transfer 5-4
6-1 Additional Managed Parameters for a Virtual Channel without Segment Headers When TC Space Data Link Protocol Supports SDLS 6-11
6-2 Additional Managed Parameters for a MAP When TC Space Data Link Protocol Supports SDLS 6-11

1 INTRODUCTION

1.1 PURPOSE

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

1.2 SCOPE

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

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

b) the protocol data units employed by the protocol; and

c) the procedures performed by the protocol.

It does not specify:

a) individual implementations or products;

b) the implementation of service interfaces within real systems;

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

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

1.3 APPLICABILITY

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

The Recommended Standard specified in this document is to be invoked through the normal standards programs of each CCSDS Agency and is applicable to those missions for which cross support based on capabilities described in this Recommended Standard is anticipated. Where mandatory capabilities are clearly indicated in sections of the Recommended Standard, they must be implemented when this document is used as a basis for cross support. Where options are allowed or implied, implementation of these options is subject to specific bilateral cross support agreements between the Agencies involved.

1.4 RATIONALE

The CCSDS believes it is important to document the rationale underlying the recommendations chosen, so that future evaluations of proposed changes or improvements will not lose sight of previous decisions.

1.5 DOCUMENT STRUCTURE

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

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

b) section 2 provides an overview of the TC Space Data Link Protocol;

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

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

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

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

g) annex A lists all acronyms used within this document;

h) annex B provides a list of informative references.

1.6 CONVENTIONS AND DEFINITIONS

1.6.1 DEFINITIONS

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

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

a) blocking;

b) connection;

c) Data Link Layer;

d) entity;

e) flow control;

f) Network Layer;

g) peer entities;

h) Physical Layer;

i) protocol control information;

j) protocol data unit;

k) real system;

l) segmenting;

m) service;

n) Service Access Point (SAP);

o) SAP address;

p) service data unit.

1.6.1.2 Definitions from OSI Service Definition Conventions

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

a) confirmation;

b) indication;

c) primitive;

d) request;

e) response;

f) service provider;

g) service user.

1.6.1.3 Terms Defined in this Recommended Standard

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

asynchronous: not *synchronous* (see below).

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

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.

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.

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

1.6.2 NOMENCLATURE

1.6.2.1 Normative Text

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

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

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

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

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

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

1.6.2.2 Informative Text

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

– Overview;

– Background;

– Rationale;

– Discussion.

1.6.3 CONVENTIONS

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

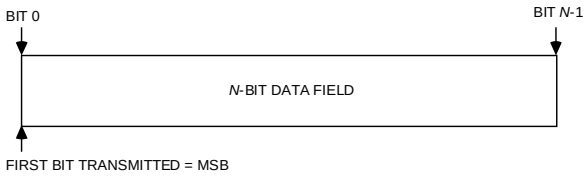

Figure 1-1: Bit Numbering Convention

In accordance with standard data-communications practice, data fields are often grouped into eight-bit 'words' which conform to the above convention. Throughout this Recommended Standard, such an eight-bit word is called an 'octet'.

The numbering for octets within a data structure starts with zero.

By CCSDS convention, all 'spare' bits shall be permanently set to '0'.

1.7 REFERENCES

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

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

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

[3] *TC Synchronization and Channel Coding*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-2. Washington, D.C.: CCSDS, September 2010.

[4] *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.

[5] "Registries." Space Assigned Number Authority. http://sanaregistry.org/r/.

[6] *CCSDS Global Spacecraft Identification Field Code Assignment Control Procedures*. Issue 6. Recommendation for Space Data System Standards (Blue Book), CCSDS 320.0-B-6. Washington, D.C.: CCSDS, October 2013.

[7] *Space Data Link Security Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.0-B-1. Washington, D.C.: CCSDS, September 2015.

NOTE – Informative references are listed in annex B.

2 OVERVIEW

2.1 CONCEPT OF TC SPACE DATA LINK PROTOCOL

2.1.1 ARCHITECTURE

The TC 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 ground-to-space or space-to-space communications links (hereafter called space links).

Figure 2-1 illustrates the relationship of this protocol to the Open Systems Interconnection (OSI) reference model (reference [1]). Two sublayers of the Data Link Layer are defined for CCSDS space link protocols as shown in reference [B2]. The TC Space Data Link Protocol corresponds to the Logical Link Sublayer and provides functions of transferring various data using a variable-length protocol data unit called the Transfer Frame. The optional Space Data Link Layer Security Protocol (reference [7]) is provided within the Data Link Protocol Sublayer, as illustrated below. The Channel Coding Sublayer provides some additional functions necessary for transferring Transfer Frames over a space link. These functions are error-correction coding/decoding, the delimiting/synchronizing of codeblocks (consisting of one or more Transfer Frames), and bit transition generation/removal (optional). For the Channel Coding Sublayer, the Channel Coding and Synchronization Recommended Standard (reference [3]) must be used with the TC Space Data Link Protocol. How the TC Space Data Link Protocol is used in overall space data systems is shown in references [B2], [B3], and [B4].

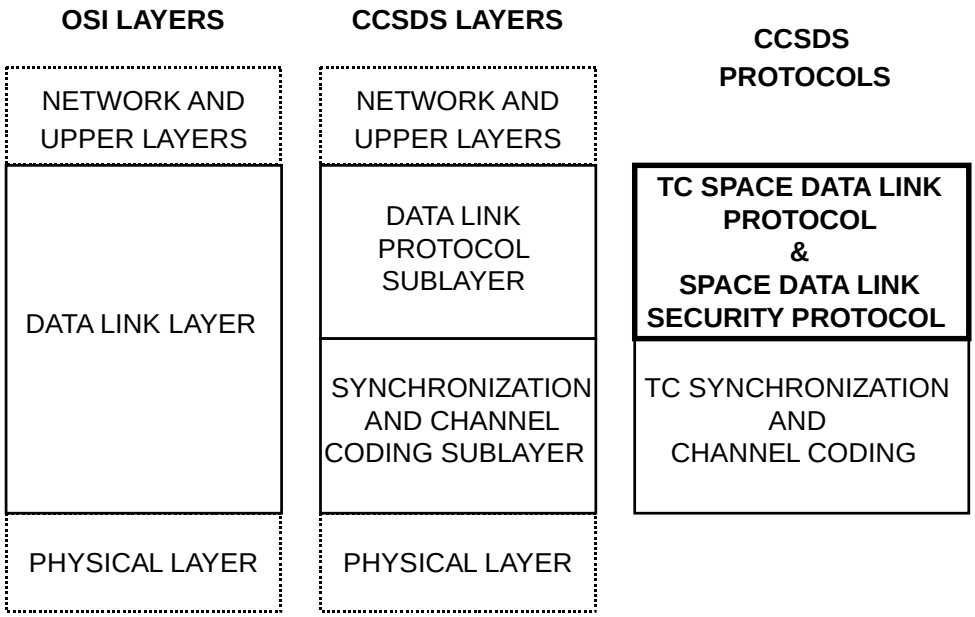

Figure 2-1: Relationship with OSI Layers

2.1.2 PROTOCOL FEATURES

2.1.2.1 Efficient Data Transfer

The TC Space Data Link Protocol provides the users with several services to transfer service data units over a space link. The major functions performed by this protocol are (1) segmentation and blocking of service data units and (2) transmission control of service data units.

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

The TC Space Data Link Protocol controls the transmission of service data units through the space link performing retransmissions needed to ensure delivery of service data units in sequence and without gaps or duplication. This function is provided by an automatic retransmission control mechanism called the Communications Operation Procedure (COP). The specification of the COP is given in reference [4]. In addition, a systematic retransmission mechanism for use on deep space links can optionally be provided by the Synchronization and Channel Coding Sublayer as specified in reference [3].

2.1.2.2 Sharing the Physical Channel

The protocol data units employed by this protocol are the TC Transfer Frame (unless otherwise stated, the terms ‘Transfer Frame’ and ‘Frame’ in this document refer to the TC Transfer Frame) and the Communications Link Control Word (CLCW). Each Transfer Frame contains a header, which provides protocol control information, and a variable-length data field, within which higher-layer service data units are carried. Transfer Frames are sent in the direction of the flow of service data units. Each CLCW contains a report that describes the status of acceptance of Transfer Frames. CLCWs are sent from the receiver to the sender of Transfer Frames.

A key feature of the TC Space Data Link Protocol is the concept of ‘Virtual Channels’. The Virtual Channel facility allows one Physical Channel to be shared among multiple higher-layer data streams, each of which may have different service requirements. A single Physical Channel may therefore be divided into several separate logical data channels, each known as a ‘Virtual Channel’ (VC). Each Transfer Frame transferred over a Physical Channel belongs to one of the Virtual Channels of the Physical Channel.

Optionally, this protocol enables service data units from different sources to be multiplexed together in one Virtual Channel using 'Multiplexer Access Points' (MAPs). If MAPs are used, service data units arriving at the service access point for a MAP at the sending end are transferred to the corresponding MAP at the receiving end.

2.1.2.3 Optional Space Data Link Security Protocol

The Data Link Protocol Sublayer includes the Space Data Link Security (SDLS) protocol specified in reference [7]. The SDLS protocol can provide security, such as authentication and confidentiality, for TC Transfer Frames. Support for the SDLS protocol is an optional feature of the TC Space Data Link Protocol.

NOTE – The introduction of the SDLS protocol in this Recommended Standard makes no changes to any requirements that apply to a TC Space Data Link Protocol that does not support the SDLS protocol.

The security provided by the SDLS protocol can vary between Virtual Channels and between MAPs within a Virtual Channel. So, for example, there can be some Virtual Channels with security and some without. The type of security can vary from one Virtual Channel to another and from one MAP to another.

2.1.3 ADDRESSING

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

MCID = TFVN + SCID;

GVCID = MCID + VCID
= TFVN + SCID + VCID.

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

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

In the optional Segment Header, there is a field called Multiplexer Access Point Identifier (MAP ID). All Transfer Frames with the same GVCID and MAP ID constitute a MAP Channel. If the Segment Header is used, a Virtual Channel consists of one or multiple MAP Channels. The concatenation of a GVCID and a MAP ID is known as a Global MAP ID (GMAP ID). Therefore

GMAP ID = GVCID + MAP ID
= MCID + VCID + MAP ID
= TFVN + SCID + VCID + MAP ID.

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

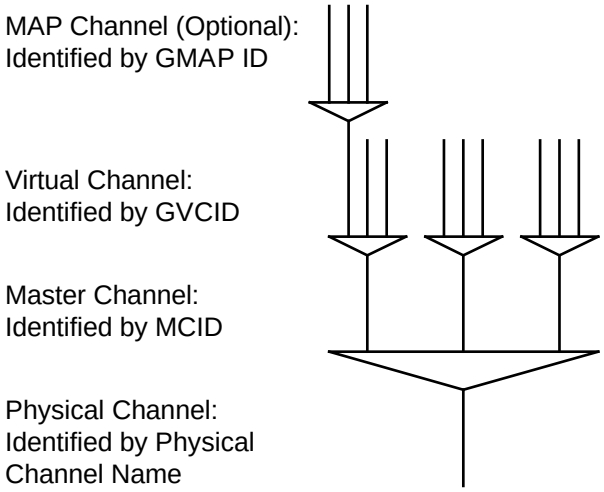

Figure 2-2: Relationships Between Channels

2.1.4 PROTOCOL DESCRIPTION

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

a) the services provided to the users;

b) the protocol data units; and

c) the procedures performed by the protocol.

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

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

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

2.2 OVERVIEW OF SERVICES

2.2.1 COMMON FEATURES OF SERVICES

The TC Space Data Link Protocol provides users with data transfer services. The point at which a service is provided by a protocol entity to a user is called a Service Access Point (SAP) (see reference [1]). Each service user is identified by a SAP address. At a SAP, two ports may be provided, each of which is for one of the service types described in 2.2.2.

Service data units of the same type submitted to a SAP (or a port if implemented) are processed in the order of submission. No processing order is maintained for service data units submitted to different SAPs (or ports).

NOTE – Implementations may be required to perform flow control at a SAP (or a port if present) between the service user and the service provider. However, CCSDS does not make any recommendations for a scheme for flow control between the user and the provider.

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

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

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

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

NOTE – This Recommended Standard assumes that these services are provided at the end points of a space link. However, this Recommended Standard makes no assumptions concerning how these end points are composed or configured either on-board a spacecraft or in a ground system. In a ground system, the services defined by this Recommended Standard may be extended or enhanced with Space Link Extension Services (reference [B5]).

2.2.2 SERVICE TYPES

2.2.2.1 Overview

The TC Space Data Link Protocol provides two service types (Sequence-Controlled and Expedited) that determine how reliably service data units supplied by the sending user are delivered to the receiving user.

Both of these two service types are provided at any Service Access Point except for the Virtual Channel Frame, Master Channel Frame, and COP Management Services. The user requests with a parameter of the service request primitive whether the Sequence-Controlled or Expedited Service Type should be applied to each service data unit. Alternatively, the service provider may provide two separate ports (one for the Sequence-Controlled Service Type and the other for the Expedited Service Type) within a Service Access Point.

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

2.2.2.2 Sequence-Controlled Service (Type-A Service)

The Sequence-Controlled Service (Type-A Service) utilizes an Automatic Repeat Request (ARQ) procedure of the 'go-back-*n*' type with sequence-control mechanisms at both sending and receiving ends and a standard report returned from the receiving end to the sending end.

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

a) no service data unit is lost;

b) no service data unit is duplicated;

c) no service data unit is delivered out of sequence.

The Type-A Service guarantees, with a high probability, complete in-sequence delivery of service data units supplied by a user on a single MAP or Virtual Channel. Because retransmission is performed independently on each Virtual Channel, there is no guarantee that Type-A service data units transmitted on separate Virtual Channels will be delivered to the receiving users in the order initially presented. Further, because MAP multiplexing is performed before the sequence-control mechanisms are applied, there is no guarantee that Type-A service data units transmitted on separate MAP Channels will be delivered to the receiving users in the order initially presented.

NOTE – Some implementations of this protocol may not distinguish service data units transferred with Type-A Service from service data units transferred with Type-B Service at the receiving end. In this case, if both Type-A Service and Type-B Service are used simultaneously on one MAP Channel, the receiving end may not be able to reconstruct service data units transferred with Type-A Service even though the ARQ procedure has been applied to them (because, for example, the Type-A Transfer Frames derived from a single service data unit may be interlaced, at the receiving end, with Type-B Transfer Frames derived from a different service data unit). For this implementation, the sending end should terminate any ongoing Type-A Service before starting a Type-B Service on the same Virtual Channel.

2.2.2.3 Expedited Service (Type-B Service)

The Expedited Service (Type-B Service) is normally used either in exceptional operational circumstances, typically during spacecraft recovery operations, or when a higher layer protocol provides a retransmission capability.

For Type-B Service, service data units supplied by the sending user are transmitted only once (i.e., no retransmission). There is no guarantee that all Type-B service data units are delivered to the receiving user.

NOTES

1 Although Type-B Service carries the name ‘Expedited’, it is neither a required method nor a faster method for sending urgent data to the receiving end. If the service provider is supporting a reliable Type-A Service, then Type-A Service should be used exclusively.

2 For frames carrying service data units on the Type-B Service, the TC Space Data Link Protocol does not use the systematic retransmission mechanism that is optionally provided by the Synchronization and Channel Coding Sublayer (see 2.4.2).

2.2.3 SUMMARY OF SERVICES

2.2.3.1 Overview

Seven services are provided by the TC Space Data Link Protocol. Two of them (MAP Packet and MAP Access) are provided for a MAP Channel. Four of them (VC Packet, Virtual Channel Access, Virtual Channel Frame, and COP Management) are provided for a Virtual Channel. One of them (Master Channel Frame) is provided for a Master Channel.

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

– encryption, to provide confidentiality by hiding data content;

– authentication, to confirm the source and integrity of the data.

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

Service	Service Type	Service Data Unit	SAP Address	SDLS Security Features
MAP Packet (MAPP)	Type-A and Type-B	Packet	GMAP ID + Packet Version Number	All
Virtual Channel Packet (VCP)	Type-A and Type-B	Packet	GVCID + Packet Version Number	All
MAP Access (MAPA)	Type-A and Type-B	MAP_SDU	GMAP ID	All
Virtual Channel Access (VCA)	Type-A and Type-B	VCA_SDU	GVCID	All
Virtual Channel Frame (VCF)	N/A	Transfer Frame	GVCID	None
Master Channel Frame (MCF)	N/A	Transfer Frame	MCID	None
COP Management	N/A	N/A	GVCID	N/A

2.2.3.2 MAP Packet (MAPP) Service

The MAP Packet (MAPP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link on a specified MAP Channel. The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS. PVNs presently authorized by CCSDS are defined in reference [5].

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

In the context of a given GMAP ID (i.e., a GVCID and a MAP ID), 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 GMAP ID. Different users (i.e., Packets with different versions) can 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.

2.2.3.3 Virtual Channel Packet (VCP) Service

The Virtual Channel Packet (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 [5].

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, or uses one port for Type-A Packets and another port for Type-B Packets.

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.

2.2.3.4 MAP Access (MAPA) Service

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

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the MAPA 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, or uses one port for Type-A service data units and another port for Type-B service data units.

For a given service instance, only one user, identified with the GMAP ID (i.e., GVCID and MAP ID) of the MAP Channel, can use this service on a MAP Channel. Service data units from different users are not multiplexed together within one MAP Channel.

2.2.3.5 Virtual Channel Access (VCA) Service

The Virtual Channel Access (VCA) Service provides transfer of a sequence of privately formatted service data units of variable length across a space link. The length of the service data units transferred by this service can not exceed the maximum length of the Data Field of the Transfer Frame.

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, or uses one port for Type-A service data units and another port for Type-B service data units.

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.

2.2.3.6 Virtual Channel Frame (VCF) Service

The Virtual Channel Frame (VCF) Service provides transfer of a sequence of TC Transfer Frames, created by an independent protocol entity, on a Virtual Channel across a space link. The service does not guarantee completeness nor does it make any distinction between Sequence-Controlled and Expedited service types applicable to service data units 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 Virtual Channel, can use this service on a Virtual Channel. Service data units from different users are not multiplexed together within one Virtual Channel.

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

2.2.3.7 Master Channel Frame (MCF) Service

The Master Channel Frame (MCF) Service provides transfer of a sequence of TC Transfer Frames of a Master Channel, created by an independent protocol entity, across a space link. The service does not guarantee completeness nor does it make any distinction between Sequence-Controlled and Expedited service types applicable to service data units 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 MCID of the Master Channel, can use this service on a Master Channel. Service data units from different users are not multiplexed together within one Master Channel.

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

2.2.3.8 COP Management Service

The COP Management Service is used by a user at the sending end for managing the operations of the COP for a particular Virtual Channel. The user manages the operations of the COP by invoking Directives defined in reference [4]. 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 COP for a particular Virtual Channel. For a given service instance, only one user, identified with the GVCID of the Virtual Channel, is allowed to use this service on a Virtual Channel.

2.2.4 RESTRICTIONS ON SERVICES

There are some restrictions on the services provided on a Physical Channel.

a) If the Master Channel Frame Service exists on a Master Channel, other services shall not exist simultaneously on that Master Channel.

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

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

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

e) On one MAP Channel, the MAP Access Service shall not exist simultaneously with the MAP Packet Service.

2.3 OVERVIEW OF FUNCTIONS

2.3.1 GENERAL FUNCTIONS

Using services of lower layers, the TC Space Data Link Protocol transfers various service data units, supplied by sending users, encapsulated in a sequence of protocol data units. The protocol data units, known as TC Transfer Frames, have variable lengths and are 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 service data units to transfer service data units of various sizes in protocol data units suitable for efficient transfer;

c) multiplexing/demultiplexing in order for various service users to share a single Physical Channel;

d) retransmission of missing protocol data units, rejection of out-of-sequence and duplicated protocol data units, and control of sequence-control mechanisms at sending and receiving ends to guarantee complete and in-order delivery (for Type-A Service only);

e) flow control (for Type-A Service only).

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

The protocol entity does not perform the following protocol functions:

a) connection establishment and release;

b) management or configuration of the SDLS protocol.

2.3.2 INTERNAL ORGANIZATION OF PROTOCOL ENTITY

Figures 2-3 and 2-4 show the internal organization of the protocol entity of the sending and receiving ends, respectively. Data flow from top to bottom in figure 2-3 and from bottom to top in figure 2-4. The four functions in the upper part of these figures are collectively called the Segmentation Sublayer, and the other four functions in the lower part are collectively called the Transfer Sublayer.

These figures identify data-handling functions performed by the protocol entity. The purpose of these figures is to show logical relationships among the functions of the protocol entity. 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.

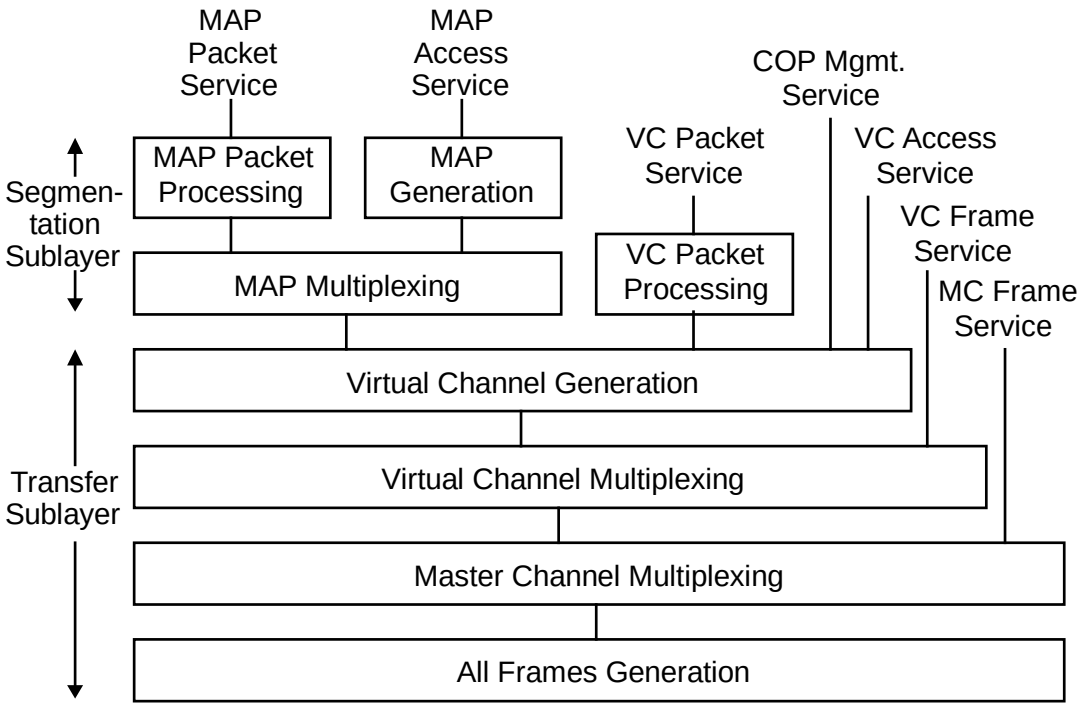

Figure 2-3: Internal Organization of Protocol Entity (Sending End)

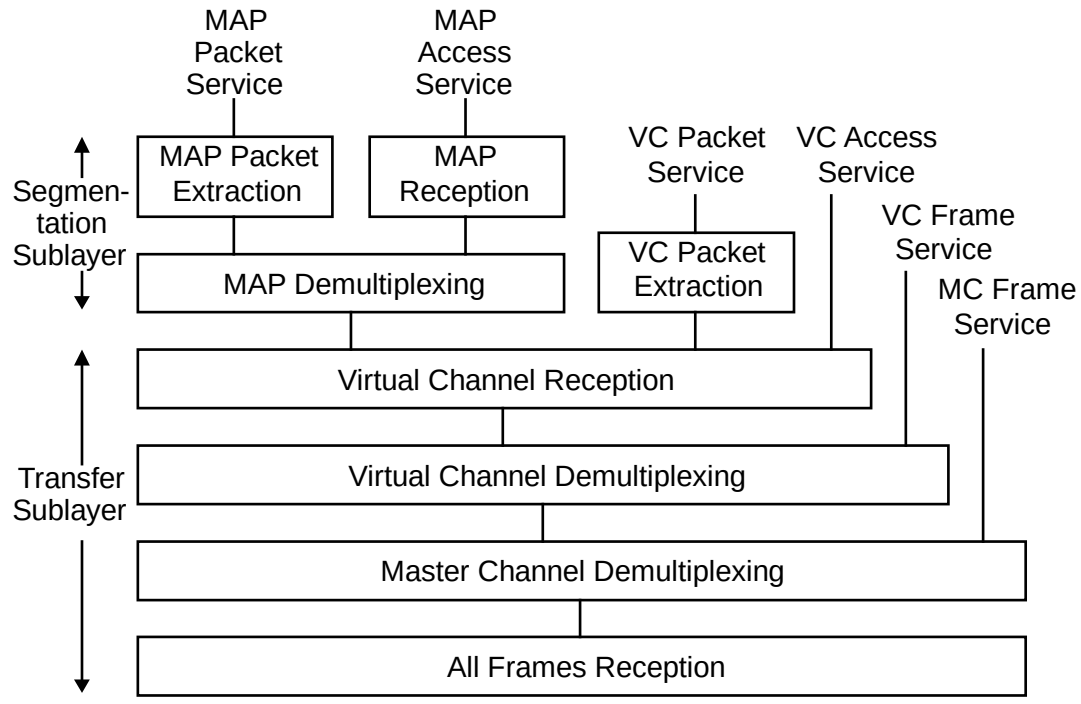

Figure 2-4: Internal Organization of Protocol Entity (Receiving End)

By extracting multiplexing and demultiplexing functions from figures 2-3 and 2-4, the relationship among various data units can be shown as figure 2-5, which is known as the Channel Tree of the TC Space Data Link Protocol.

In figure 2-5, 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, and generating a single stream of data units.

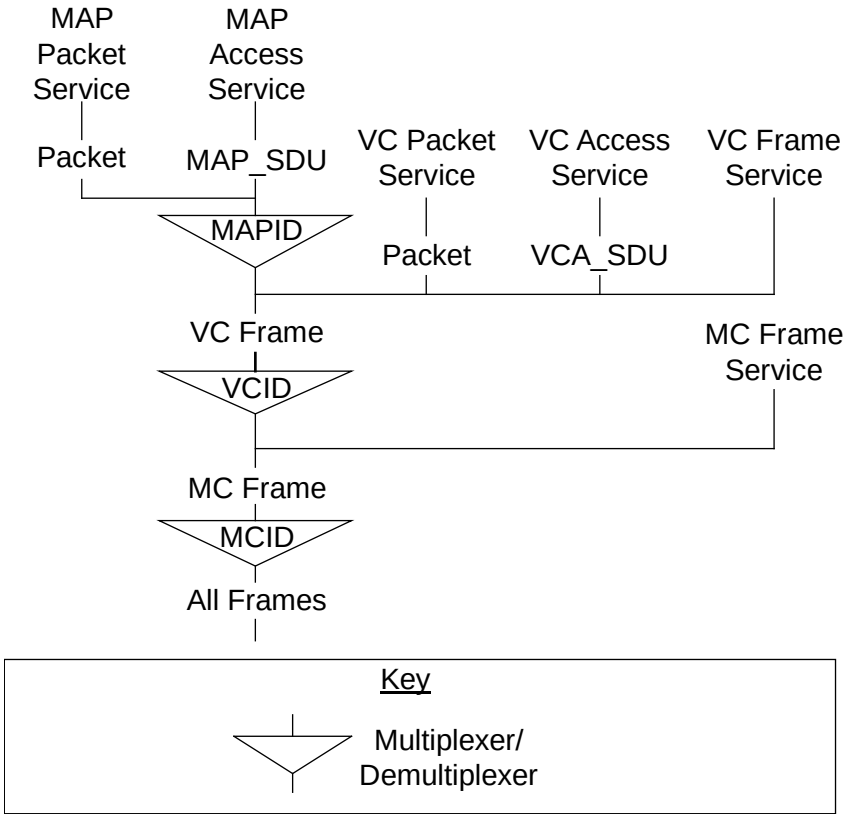

Figure 2-5: TC Space Data Link Protocol Channel Tree

2.3.3 COMMUNICATIONS OPERATION PROCEDURE (COP)

The Communications Operation Procedure (COP) fully specifies the closed-loop procedures executed by the sending and receiving ends of the TC Space Data Link Protocol. The COP, which exists wholly within this protocol, consists of a pair of synchronized procedures for each Virtual Channel: a Frame Operation Procedure (FOP) that executes within the sending entity; and a Frame Acceptance and Reporting Mechanism (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 Communications Link Control Words (CLCWs) and thus closes the loop.

Within this protocol, the COP provides a basic Quality of Service (QoS), i.e., the delivery of service data units to the layer above at the receiving end, correct and without omission or

duplication, and in the same sequential order in which they were received from the layer above at the sending end.

Underlying this protocol is a space link, which interconnects its sending and receiving ends. If a perfect link existed, the QoS would be assured since the exact duplicate of the series of service data units input at the sending end would appear at the receiving end. However, space links are noisy and may introduce errors or discontinuities into transmitted data streams. The job of the COP within this protocol is therefore to ensure, in the presence of such errors or discontinuities introduced by the space link, the correctness, completeness, and sequentiality of the delivered service data units.

Correctness of the delivered service data units is guaranteed (within known error probabilities) by the error-protection encoding applied by the Channel Coding Sublayer, and by the Frame Validation Checks performed in this protocol. However, validation of the completeness, sequentiality, and non-duplication of the delivered service data units on a particular Virtual Channel requires that an accounting (i.e., numbering) scheme for Transfer Frames be implemented by the COP.

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

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

Only one COP, which is called COP-1, is defined in this Recommended Standard. The detailed specification of the COP-1 is given in reference [4].

CAUTION – The controlling specifications for the logical operations which must be executed to perform the COP-1 are contained in a more detailed CCSDS Recommended Standard (reference [4]). In the event of any conflict between the descriptive text contained in this Recommended Standard and the text of reference [4], the more detailed specifications contained in reference [4] are normative.

2.4 SERVICES ASSUMED FROM LOWER LAYERS

2.4.1 SERVICES ASSUMED FROM THE CHANNEL CODING SUBLAYER

As described in 2.1.1, the TC Synchronization and Channel Coding Recommended Standard (reference [3]) must be used with the TC Space Data Link Protocol as the Synchronization and Channel Coding Sublayer specification. The functions provided by the TC Synchronization and Channel Coding Recommended Standard are as follows:

a) error control encoding and decoding functions;

b) bit transition generation and removal functions (optional);

c) delimiting and synchronizing functions.

The Synchronization and Channel Coding Sublayer, then, transfers variable-length, delimited protocol data units as an intermittent stream of bits over a space link using the services of the underlying Physical Layer.

2.4.2 SYSTEMATIC RETRANSMISSIONS

In addition, the TC Space Data Link Protocol can request the Synchronization and Channel Coding Sublayer to perform systematic retransmissions of the data units submitted to it. The retransmissions can improve the efficiency of the protocol for deep space missions on links with long light time delays.

The definition of the service interface to the Synchronization and Channel Coding Sublayer specified in reference [3] 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 data unit is transferred once.

The TC Space Data Link Protocol requests the systematic retransmissions in accordance with parameters set by management. For each Virtual Channel, management sets the value to be used for the Repetitions parameter when requesting the transfer of frames carrying service data units on the Type-A Service. For each Virtual Channel, management sets a similar parameter for frames carrying COP control commands. For a Physical Channel, management sets an upper limit for the value of the Repetitions parameter specified in reference [3].

When requesting the transfer of frames carrying service data units on the Type-B Service, the TC Space Data Link Protocol always sets the value of the Repetitions parameter to one.

2.4.3 PERFORMANCE REQUIREMENTS TO LOWER LAYERS

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

a) the probability of misidentifying the MCID, VCID and MAP ID shall be less than a mission-specified value;

b) the probability of rejection of Transfer Frames by the Channel Coding Sublayer due to channel errors shall be less than a mission-specified value.

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

3 SERVICE DEFINITION

3.1 OVERVIEW

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

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

3.2 SOURCE DATA

3.2.1 SOURCE DATA OVERVIEW

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

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

a) Packet;

b) MAP Access Service Data Unit (MAP_SDU);

c) Virtual Channel Access Service Data Unit (VCA_SDU);

d) TC Transfer Frame.

3.2.2 PACKET

3.2.2.1 Packets shall be transferred over a space link via the MAP Packet and VC Packet Services.

3.2.2.2 The Packets transferred by this service must have a PVN authorized by CCSDS.

NOTES

1 Packets are variable-length, delimited, octet-aligned data units, and are usually the protocol data unit of a Network Layer protocol.

2 PVNs presently authorized by CCSDS are defined in reference [5].

3.2.2.3 If blocking of Packets is performed by the service provider, 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.

3.2.3 MAP CHANNEL ACCESS SERVICE DATA UNIT (MAP_SDU)

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

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

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

3.2.4 VIRTUAL CHANNEL ACCESS SERVICE DATA UNIT (VCA_SDU)

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

3.2.4.2 A single VCA_SDU shall be transmitted in the Data Field of a single Transfer Frame, and therefore the length of VCA_SDUs shall not exceed the maximum length of the Transfer Frame Data Field.

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

3.2.5 TC TRANSFER FRAME

If it is present on the Physical Channel, the Frame Error Control Field of Transfer Frames submitted to the Virtual Channel Frame and Master Channel Frame Services shall be included and shall be set to ‘all zeroes’.

NOTE – The TC Transfer Frame is the variable-length protocol data unit of the TC Space Data Link Protocol, but it can also be used as the service data unit of the Virtual Channel Frame and Master Channel Frame Services. Its format is defined in 4.1 of this Recommended Standard. Transfer Frames submitted to the Virtual Channel Frame and Master Channel Frame Services are actually partially formatted TC Transfer Frames having ‘empty’ Frame Error Control Fields.

3.3 MAP PACKET SERVICE

3.3.1 OVERVIEW

The MAP Packet (MAPP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link on a specified MAP Channel. The Packets transferred by this service must be assigned a Packet Version Number (PVN) by CCSDS. Packet Version Numbers presently authorized by CCSDS are defined in reference [5].

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

A user of this service is a protocol entity identified with the PVN and a GMAP ID (i.e., a GVCID and a MAP ID) that sends or receives Packets with a single PVN. Different users (i.e., Packets with different versions) can 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 MAPP SERVICE PARAMETERS

3.3.2.1 General

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

3.3.2.2 Packet

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

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

3.3.2.3 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 MAPP Service.

3.3.2.4 MAP ID

The MAP ID parameter shall contain a MAP ID that indicates the MAP Channel (within the Virtual Channel specified by the GVCID) through which the Packet is to be transferred.

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

3.3.2.5 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 MAPP Service and identifies the upper-layer protocol entity that uses the MAPP Service.

3.3.2.6 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.

3.3.2.7 Service Type

3.3.2.7.1 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).

3.3.2.7.2 When separate ports are provided for Type-A and Type-B Services (see 2.2.2), the Service Type parameter is not used.

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

3.3.2.8 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 [4].

3.3.2.9 Packet Quality Indicator

The Packet Quality Indicator is an optional parameter that may be used to notify the user at the receiving end of the Packet Service whether the Packet delivered by the primitive is complete or partial. This parameter shall be used when the service provider is required to deliver incomplete Packets to the user at the receiving end.

3.3.2.10 Verification Status Code

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

3.3.2.10.2 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.

3.3.2.10.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [7].

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

3.3.3 MAPP SERVICE PRIMITIVES

3.3.3.1 General

The service primitives associated with the MAPP service are:

a) MAPP.request;

b) MAPP_Notify.indication;

c) MAPP.indication.

3.3.3.2 MAPP.request

3.3.3.2.1 Function

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

3.3.3.2.2 Semantics

The MAPP.request primitive shall provide parameters as follows:

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

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

3.3.3.2.3 When Generated

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

3.3.3.2.4 Effect On Receipt

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

3.3.3.3 MAPP_Notify.indication

3.3.3.3.1 Function

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

3.3.3.3.2 Semantics

The MAPP_Notify.indication primitive shall provide parameters as follows:

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

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

3.3.3.3.3 When Generated

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

3.3.3.3.4 Effect On Receipt

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

3.3.3.4 MAPP.indication

3.3.3.4.1 Function

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

3.3.3.4.2 Semantics

The MAPP.indication primitive shall provide parameters as follows:

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

3.3.3.4.3 When Generated

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

3.3.3.4.4 Effect On Receipt

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

3.4 VIRTUAL CHANNEL PACKET SERVICE

3.4.1 OVERVIEW

The Virtual Channel Packet (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 Packet Version Number (PVN) by CCSDS. Packet Version Numbers presently authorized by CCSDS are defined in reference [5].

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, or uses one port for Type-A Packets and another port for Type-B Packets.

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 When separate ports are provided for Type-A and Type-B Services (see 2.2.2), the Service Type parameter is not used.

3.4.2.6.3 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 [4].

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 [7].

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)

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

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)

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

3.4.3.3.3 When Generated

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

The service is unidirectional and asynchronous. Both Sequence-Controlled (Type-A) and Expedited (Type-B) service types are provided for the MAPA 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, or uses one port for Type-A service data units and another port for Type-B service data units.

Only one user, identified with the GMAP ID (i.e., GVCID and MAP ID) of the MAP Channel, can use this service on a MAP Channel. Service data units from different users are not multiplexed together within one MAP Channel.

3.5.2 MAPA SERVICE PARAMETERS

3.5.2.1 General

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

3.5.2.2 MAP_SDU

The MAP_SDU parameter shall contain a MAP_SDU to be transferred over the MAP channel identified by MAP ID.

NOTE – The MAP_SDU is the service data unit transferred by the MAPA Service. Restrictions on the MAP_SDUs transferred by the MAPA Service are stated in 3.2.3.

3.5.2.3 GVCID

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

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

3.5.2.4 MAP ID

The MAP ID parameter shall contain the MAP ID of the MAP Channel (within the Virtual Channel specified by GVCID) through which the MAP_SDU is to be transferred.

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

3.5.2.5 SDU ID

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

3.5.2.6 Service Type

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

3.5.2.6.2 When separate ports are provided for Type-A and Type-B Services (see 2.2.2), the Service Type parameter is not used.

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

3.5.2.7 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with the transfer of a MAP_SDU. The values taken by this parameter are defined in reference [4].

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 Verification Status Code 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 MAP.

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 [7].

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

3.5.3 MAPA SERVICE PRIMITIVES

3.5.3.1 General

The service primitives associated with this service are:

a) MAPA.request;

b) MAPA_Notify.indication;

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 MAP_SDU be transferred to the user at the receiving end through the specified MAP Channel.

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

3.5.3.2.2 Semantics

The MAPA.request primitive shall provide parameters as follows:

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

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

3.5.3.2.3 When Generated

The sending-end service user shall generate a MAPA.request primitive when a MAP_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 MAP_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 MAP_SDU.

3.5.3.3.2 Semantics

The MAPA.indication primitive shall provide parameters as follows:

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

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

3.5.3.3.3 When Generated

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

3.5.3.3.4 Effect On Receipt

The effect of 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 MAP_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	(MAP_SDU, GVCID, MAP ID, Service Type (optional), Verification Status Code (optional))

3.5.3.4.3 When Generated

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

3.5.3.4.4 Effect On Receipt

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

3.6 VIRTUAL CHANNEL ACCESS SERVICE

3.6.1 OVERVIEW

The Virtual Channel Access (VCA) Service provides transfer of a sequence of privately formatted service data units of variable length across a space link. The length of the service data units transferred by this service should not exceed the maximum length of the Data Field of the Transfer Frame.

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, or uses one port for Type-A service data units and another port for Type-B service data units.

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.

3.6.2 VCA SERVICE PARAMETERS

3.6.2.1 General

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

3.6.2.2 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.

3.6.2.3 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.

3.6.2.4 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.

3.6.2.5 Service Type

3.6.2.5.1 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).

3.6.2.5.2 When separate ports are provided for Type-A and Type-B Services (see 2.2.2), the Service Type parameter is not used.

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

3.6.2.6 Notification Type

In notifications to the user, the Notification Type parameter shall contain information about an event associated with the transfer of a VCA_SDU. The values taken by this parameter are defined in reference [4].

3.6.2.7 Verification Status Code

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

3.6.2.7.2 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.

3.6.2.7.3 A non-zero value shall indicate that the SDLS protocol has detected an error: the values taken by this parameter are defined in reference [7].

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.

3.6.3 VCA SERVICE PRIMITIVES

3.6.3.1 General

The service primitives associated with this service are:

a) VCA.request;

b) VCA_Notify.indication;

c) VCA.indication.

3.6.3.2 VCA.request

3.6.3.2.1 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.

3.6.3.2.2 Semantics

The VCA.request primitive shall provide parameters as follows:

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

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

3.6.3.2.3 When Generated

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

3.6.3.2.4 Effect On Receipt

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

3.6.3.3 VCA_Notify.indication

3.6.3.3.1 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.

3.6.3.3.2 Semantics

The VCA.indication primitive shall provide parameters as follows:

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

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

3.6.3.3.3 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.

3.6.3.3.4 Effect On Receipt

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

3.6.3.4 VCA.indication

3.6.3.4.1 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.

3.6.3.4.2 Semantics

The VCA.indication primitive shall provide parameters as follows:

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

3.6.3.4.3 When Generated

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

3.6.3.4.4 Effect On Receipt

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

3.7 VIRTUAL CHANNEL FRAME SERVICE

3.7.1 OVERVIEW

The Virtual Channel Frame (VCF) Service provides transfer of a sequence of TC Transfer Frames of a Virtual Channel, created by an independent protocol entity, across a space link. The service does not guarantee completeness nor does it make any distinction between Sequence-Controlled and Expedited service types against service data units supplied by the user. The user should perform necessary procedures to provide Sequence-Controlled and Expedited service types.

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.

3.7.2 VCF SERVICE PARAMETERS

3.7.2.1 General

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

3.7.2.2 Frame

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

NOTE – The TC Transfer Frame is the service data unit transferred by the VCF Service. The format of the TC Transfer Frame is defined in 4.1. Restrictions on the TC Transfer Frames transferred by the VCF Service are stated in 3.2.5.

3.7.2.3 GVCID

The GVCID parameter shall contain the GVCID of the Virtual Channel on which the Frame is to be transferred.

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

3.7.3 VCF SERVICE PRIMITIVES

3.7.3.1 General

The service primitives associated with VCF Service are:

a) VCF.request;

b) VCF.indication.

3.7.3.2 VCF.request

3.7.3.2.1 Function

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

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

3.7.3.2.2 Semantics

The VCF.request primitive shall provide parameters as follows:

VCF.request	(Frame, GVCID)

3.7.3.2.3 When Generated

The sending-end service user shall generate a VCF.request primitive when a Frame is ready to be transferred.

3.7.3.2.4 Effect On Receipt

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

3.7.3.3 VCF.indication

3.7.3.3.1 Function

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

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

3.7.3.3.2 Semantics

The VCF.indication primitive shall provide parameters as follows:

VCF.indication	(Frame, GVCID)

3.7.3.3.3 When Generated

The receiving-end service provider shall generate a VCF.indication primitive when a Frame is ready to be delivered to the user.

3.7.3.3.4 Effect On Receipt

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

3.8 MASTER CHANNEL FRAME SERVICE

3.8.1 OVERVIEW

The Master Channel Frame (MCF) Service provides transfer of a sequence of TC Transfer Frames of a Master Channel, created by an independent protocol entity, across a space link. The service does not guarantee completeness nor does it make any distinction between Sequence-Controlled and Expedited service types against service data units supplied by the user. The user should perform necessary procedures to provide Sequence-Controlled and Expedited service types.

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

3.8.2 MCF SERVICE PARAMETERS

3.8.2.1 General

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

3.8.2.2 Frame

The Frame parameter shall contain a TC Transfer Frame to be transferred on the Master Channel identified by MCID.

NOTE – The TC Transfer Frame is the service data unit transferred by the MCF Service. The format of the TC Transfer Frame is defined in 4.1. Restrictions on the TC Transfer Frames transferred by the MCF Service are stated in 3.2.5.

3.8.2.3 MCID

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

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

3.8.3 MCF SERVICE PRIMITIVES

3.8.3.1 General

The service primitives associated with the MCF service are:

a) MCF.request;

b) MCF.indication.

3.8.3.2 MCF.request

3.8.3.2.1 Function

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

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

3.8.3.2.2 Semantics

The MCF.request primitive shall provide parameters as follows:

MCF.request	(Frame, MCID)

3.8.3.2.3 When Generated

The sending-end service user shall generate an MCF.request primitive when a Frame is ready to be transferred.

3.8.3.2.4 Effect On Receipt

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

3.8.3.3 MCF.indication

3.8.3.3.1 Function

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

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

3.8.3.3.2 Semantics

The MCF.indication primitive shall provide parameters as follows:

MCF.indication	(Frame, MCID)

3.8.3.3.3 When Generated

The receiving-end service provider shall generate an MCF.indication primitive when a Frame is ready to be delivered.

3.8.3.3.4 Effect On Receipt

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

3.9 COP MANAGEMENT SERVICE

3.9.1 OVERVIEW

The COP Management Service is used by a user at the sending end for managing the operations of COP for a particular Virtual Channel. The user manages the operations of COP by invoking Directives defined in reference [4]. 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 for a particular Virtual Channel. Only one user, identified with the GVCID of the Virtual Channel, is allowed to use this service on a Virtual Channel.

3.9.2 COP MANAGEMENT SERVICE PARAMETERS

3.9.2.1 General

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

3.9.2.2 GVCID

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

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

3.9.2.3 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.

3.9.2.4 Directive Type

The Directive Type parameter shall contain the type of Directive. The values taken by this parameter are defined in reference [4].

3.9.2.5 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 [4].

3.9.2.6 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 [4].

3.9.2.7 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 [4].

3.9.3 COP MANAGEMENT SERVICE PRIMITIVES

3.9.3.1 General

The service primitives associated with the COP Management Service are:

a) Directive.request;

b) Directive_Notify.indication;

c) Async_Notify.indication.

3.9.3.2 Directive.request

3.9.3.2.1 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 [4].

3.9.3.2.2 Semantics

The Directive.request primitive shall provide parameters as follows:

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

3.9.3.2.3 When Generated

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

3.9.3.2.4 Effect On Receipt

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

3.9.3.3 Directive_Notify.indication

3.9.3.3.1 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.

3.9.3.3.2 Semantics

The Directive_Notify.indication primitive shall provide parameters as follows:

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

3.9.3.3.3 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.

3.9.3.3.4 Effect On Receipt

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

3.9.3.4 Async_Notify.indication

3.9.3.4.1 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.

3.9.3.4.2 Semantics

The Async_Notify.indication primitive shall provide parameters as follows:

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

3.9.3.4.3 When Generated

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

3.9.3.4.4 Effect On Receipt

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

4 PROTOCOL SPECIFICATION WITHOUT SDLS OPTION

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

4.1 PROTOCOL DATA UNIT (TC TRANSFER FRAME)

4.1.1 TC TRANSFER FRAME

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

a) Transfer Frame Header (5 octets, mandatory);

b) Transfer Frame Data Field (up to 1019 or 1017 octets, mandatory);

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

NOTES

1 The TC Transfer Frame is the protocol data unit transmitted from the sending end to the receiving end by the TC Space Data Link Protocol. In this Recommended Standard, the TC Transfer Frame is also called the Transfer Frame or Frame for simplicity.

2 The maximum Transfer Frame length allowed by a particular spacecraft or ground implementation on a particular Virtual Channel may be less than the maximum specified here.

3 The structural components of the TC Transfer Frame are shown in figure 4-1.

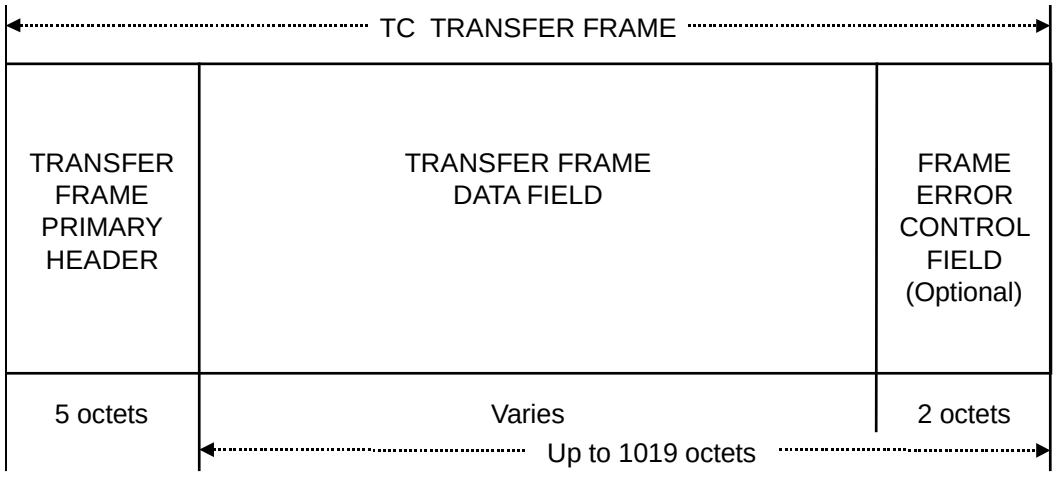

Figure 4-1: TC Transfer Frame Structural Components

4.1.2 TRANSFER FRAME PRIMARY HEADER

4.1.2.1 General

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

a) Transfer Frame Version Number (2 bits, mandatory);

b) Bypass Flag (1 bit, mandatory);

c) Control Command Flag (1 bit, mandatory);

d) Reserved Spare (2 bits, mandatory);

e) Spacecraft Identifier (10 bits, mandatory);

f) Virtual Channel Identifier (6 bits, mandatory);

g) Frame Length (10 bits, mandatory);

h) Frame Sequence Number (8 bits, mandatory).

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

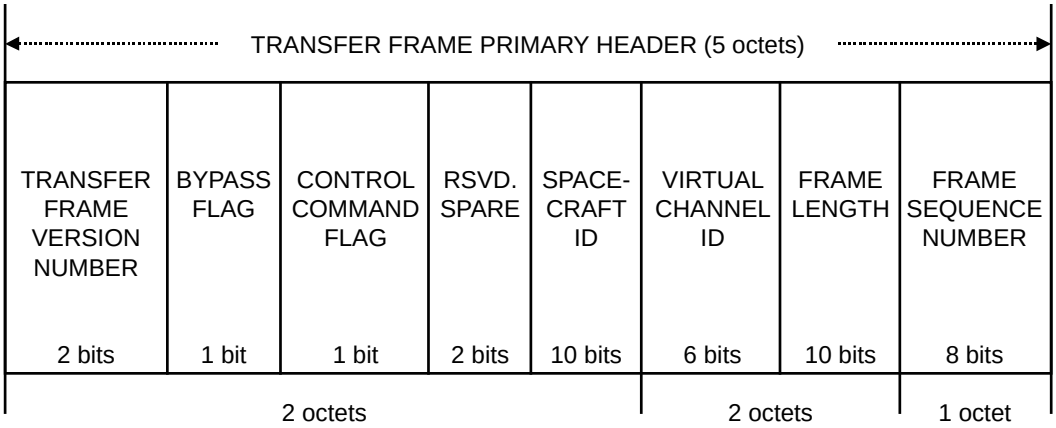

Figure 4-2: Transfer Frame Primary Header

4.1.2.2 Transfer Frame Version Number

4.1.2.2.1 Bits 0–1 of the Transfer Frame Primary Header shall contain the (binary encoded) Transfer Frame Version Number.

4.1.2.2.2 This 2-bit field shall identify the data unit as a Transfer Frame defined by this Recommended Standard and shall be set to ‘00’.

NOTE – This Recommended Standard defines the TC Version 1 Asynchronous Transfer Frame, whose binary encoded Version Number is ‘00’.

4.1.2.3 Bypass and Control Command Flags

4.1.2.3.1 Bypass Flag

4.1.2.3.1.1 Bit 2 of the Transfer Frame Primary Header shall contain the Bypass Flag.

4.1.2.3.1.2 The single-bit Bypass Flag shall be used to control the application of Frame Acceptance Checks by the receiving end:

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

b) setting the Bypass Flag to value '1' shall specify that this Transfer Frame is a Type-B Transfer Frame, and the Frame Acceptance Checks of the FARM by the receiving end shall be bypassed.

NOTES

1 The Frame Acceptance and Reporting Mechanism (FARM) associated with the COP can be made to operate in a normal 'Acceptance' mode (for Type-A Transfer Frames) or a 'Bypass' mode (for Type-B Transfer Frames), according to the setting of the Bypass Flag.

2 All Transfer Frames received by the receiving end first undergo a basic standard set of Frame Validation Checks, which are applied regardless of the setting of the Bypass Flag (see 4.4.8.3).

4.1.2.3.2 Control Command Flag

4.1.2.3.2.1 Bit 3 of the Transfer Frame Primary Header shall contain the Control Command Flag.

4.1.2.3.2.2 The single-bit Control Command Flag shall be used to specify whether the Data Field of the Transfer Frame is conveying Control Commands (Type-C) or Data (Type-D):

a) setting the Control Command Flag to value '0' shall indicate that the Transfer Frame Data Field contains a Frame Data Unit (FDU) (Type-D);

b) setting the Control Command Flag to value '1' shall indicate that the Transfer Frame Data Field contains control information (Type-C).

NOTE – In the C mode, the Transfer Frame Data Field conveys control information used to set the parameters of the FARM to the proper configuration to accept data. In the D mode, the Transfer Frame Data Field contains a Frame Data Unit.

4.1.2.3.3 Combined States of the Bypass Flag and Control Command Flag

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

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

Bypass Flag	Control Command Flag	Interpretation
0	0	Type-AD. Transfer Frame Data Field carries a Frame Data Unit, subject to Frame Acceptance Checks under control of the FARM. These Frames use the Sequence-Controlled (or AD) Service of the COP.
0	1	Reserved for future application.
1	0	Type-BD. Transfer Frame Data Field carries a Frame Data Unit, with Frame Acceptance Checks bypassed under control of the FARM. These Frames use the Expedited (or BD) Service of the COP.
1	1	Type-BC. Transfer Frame Data Field carries Control Commands, with Frame Acceptance Checks bypassed under control of the FARM. These Frames control the Sequence-Controlled Service of the COP.

4.1.2.4 Reserved Spare

4.1.2.4.1 Bits 4–5 of the Transfer Frame Primary Header shall contain the Reserved Spare.

4.1.2.4.2 These two bits are reserved for future application and shall be set to ‘00’.

4.1.2.5 Spacecraft Identifier

4.1.2.5.1 Bits 6–15 of the Transfer Frame Primary Header shall contain the Spacecraft Identifier (SCID).

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

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

NOTE – The Secretariat of the CCSDS assigns Spacecraft Identifiers according to the procedures in reference [6].

4.1.2.6 Virtual Channel Identifier

4.1.2.6.1 Bits 16–21 of the Transfer Frame Primary Header shall contain the Virtual Channel Identifier (VCID).

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

NOTE – There are no restrictions on the selection of Virtual Channel Identifiers. In particular, Virtual Channels are not required to be numbered consecutively.

4.1.2.7 Frame Length

4.1.2.7.1 Bits 22–31 of the Transfer Frame Primary Header shall contain the Frame Length.

4.1.2.7.2 This 10-bit field shall contain a length count *C* which equals one fewer than the total octets in the Transfer Frame.

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

NOTES

1 The length count *C* is expressed as:

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

2 The size of this field limits the maximum length of a Transfer Frame to 1024 octets.

4.1.2.8 Frame Sequence Number

Bits 32–39 of the Transfer Frame Primary Header shall contain the Frame Sequence Number, N(S).

NOTES

1 The procedure for assigning the Frame Sequence Number to Transfer Frames is defined in reference [4].

2 The Frame Sequence Number enables the FARM to check the sequentiality of incoming Type-A Transfer Frames. The Frame Sequence Number is Virtual Channel dependent; i.e., this protocol maintains a separate Frame Sequence Number for each of the Virtual Channels.

3 The COP does not use this field of Type-B Transfer Frames; in this case the contents of the Frame Sequence Number field is set to ‘all zeroes’ by COP.

4.1.3 TRANSFER FRAME DATA FIELD

4.1.3.1 General

4.1.3.1.1 The Transfer Frame Data Field shall follow, without gap, the Transfer Frame Primary Header.

4.1.3.1.2 The Transfer Frame Data Field, which shall contain an integral number of octets, may vary in length up to a maximum of 1019 octets (1017 octets if the Frame Error Control Field is present).

4.1.3.1.3 The Transfer Frame Data Field shall contain either an integral number of octets of data corresponding to one Frame Data Unit (for a Type-D Transfer Frame) or an integral number of octets of Control Command information (for a Type-C Transfer Frame).

4.1.3.2 Frame Data Unit

4.1.3.2.1 General

4.1.3.2.1.1 A Frame Data Unit shall consist of either

a) an integral number of User Data octets; or

b) a Segment Header followed by an integral number of User Data octets, if the Segment Header is used on the Virtual Channel.

4.1.3.2.1.2 If the Segment Header is present, the User Data contained in a Frame Data Unit shall consist of one of the following:

a) a complete Packet;

b) multiple Packets;

c) a portion of a Packet;

d) a complete MAP_SDU;

e) a portion of a MAP_SDU.

4.1.3.2.1.3 If the Segment Header is not present, the User Data contained in a Frame Data Unit shall consist of one of the following:

a) a complete Packet;

b) multiple Packets;

c) a complete VCA_SDU.

4.1.3.2.1.4 The Segment Header is required for any Virtual Channel which has more than one MAP or which transfers service data units larger than permitted in a single Transfer

Frame. Its use is optional otherwise, except that if it is used in any Transfer Frame carried on a Virtual Channel, it must be used for all Transfer Frames carrying Frame Data Units (not Control Commands) on that Virtual Channel.

NOTE – A Frame Data Unit that has a Segment Header is called a Segment.

4.1.3.2.2 Segment Header

4.1.3.2.2.1 General

4.1.3.2.2.1.1 If present, the Segment Header shall follow, without gap, the Transfer Frame Primary Header.

4.1.3.2.2.1.2 The Segment Header is optional; its presence or absence shall be established by management for each Virtual Channel.

4.1.3.2.2.1.3 The Segment Header must not be present in Transfer Frames carrying Control Commands.

4.1.3.2.2.1.4 The Segment Header shall contain the following fields:

a) Sequence Flags (2 bits, mandatory);

b) Multiplexer Access Point (MAP) Identifier (6 bits, mandatory).

NOTE – The format of the Segment Header is shown in figure 4-3.

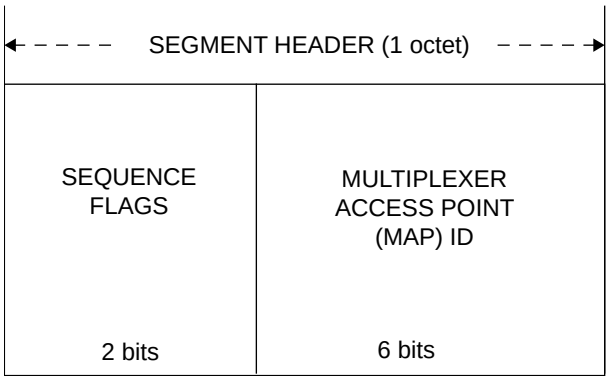

Figure 4-3: Segment Header

4.1.3.2.2.2 Sequence Flags

4.1.3.2.2.2.1 Bits 0–1 of the Segment Header shall contain the Sequence Flags.

NOTE – This two-bit field delimits the service data unit provided by the user by indicating the sequential position of the User Data contained in the Frame Data Unit relative to the service data unit of which the User Data is a part.

4.1.3.2.2.2.2 The Sequence Flags shall be interpreted as shown in table 4-2.

Table 4-2: Interpretation of the Sequence Flags

Bit 0	Bit 1	Interpretation
0	1	First portion of service data unit on one MAP
0	0	Continuing portion of service data unit on one MAP
1	0	Last portion of service data unit on one MAP
1	1	No segmentation (one complete service data unit or multiple Packets)

4.1.3.2.2.3 Multiplexer Access Point Identifier

4.1.3.2.2.3.1 Bits 2–7 of the Segment Header shall contain the Multiplexer Access Point Identifier (MAP ID).

NOTE – The MAP Identifier provides the identification of the MAP Channel within a Virtual Channel.

4.1.3.2.2.3.2 If multiple MAPs are not used on a particular Virtual Channel, but the Segment Header is otherwise required to be present, the MAP Identifier shall be set to a constant value for all Frame Data Units which are placed on that Virtual Channel.

NOTE – There are no restrictions on the selection of MAP Identifiers. In particular, MAPs are not required to be numbered consecutively.

4.1.3.3 Control Commands

4.1.3.3.1 General

Two Control Commands are defined: Unlock and Set V(R); the action to be taken when the FARM receives one of these Control Commands is defined in reference [4].

4.1.3.3.2 Unlock

The Unlock Control Command shall consist of a single octet containing ‘all zeroes’.

4.1.3.3.3 Set V(R)

The Set V(R) Control Command shall consist of three octets with the following values:

10000010 00000000 XXXXXXXX

where XXXXXXXX is the value to which the FARM should set the Receiver_Frame_Sequence_Number, V(R). This octet should therefore be set to the Sequence Number that will be put into the Header of the next Type-A Transfer Frame to be transmitted on that Virtual Channel.

4.1.3.3.4 Other Bit Combinations

All other bit combinations for Control Commands are reserved by the CCSDS for future application.

4.1.4 FRAME ERROR CONTROL FIELD

4.1.4.1 General

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

4.1.4.1.2 If present, the Frame Error Control Field shall occupy the two octets following, without gap, the Transfer Frame Data Field.

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

NOTES

1 The purpose of this field is to provide a capability for detecting errors which may have been introduced into the Transfer Frame during the transmission and data handling process.

2 Whether this field should be used on a particular Physical Channel will be determined based on the mission requirements for data quality and the selected options for the underlying Channel Coding Sublayer.

4.1.4.2 Frame Error Control Field Encoding Procedure

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

NOTE – The Bit Numbering Convention as specified in 1.6.2 is applicable below.

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

$$\text{FECF} = [(X^{16} \cdot \text{M}(X)) + (X^{(n-16)} \cdot \text{L}(X))] \text{ modulo } \text{G}(X)$$

$$= \text{P}_0 \cdot X^{15} + \text{P}_1 \cdot X^{14} + \text{P}_2 \cdot X^{13} + \ldots + \text{P}_{14} \cdot X^1 + \text{P}_{15} \cdot X^0$$

where

all arithmetic is modulo 2;

FECF is the 16-bit Frame Error Control Field with the first bit transferred being the most significant bit P_0 taken as the coefficient of the highest power of X;

n is the number of bits in the encoded message;

$\text{M}(X)$ is the (n-16)-bit information message to be encoded expressed as a polynomial with binary coefficients, with the first bit transferred being the most significant bit M_0 taken as the coefficient of the highest power of X;

$\text{L}(X)$ is the presetting polynomial given by

$$\text{L}(X) = \sum_{i=0}^{15} X^i ;$$

$\text{G}(X)$ 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 4-4. For each frame, the shift register cells are initialized to '1'. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the sixteen FECF bits.

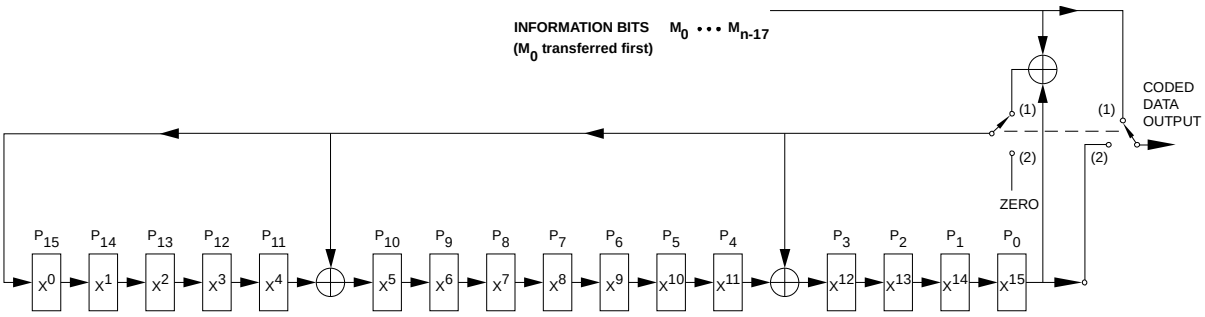

Figure 4-4: Logic Diagram of the Encoder

4.1.4.3 Frame Error Control Field Decoding Procedure

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

$$S(X) = [(X^{16} \cdot C^*(X)) + (X^n \cdot L(X))] \text{ modulo } G(X)$$

where

C*(X) is the received block, including the Frame Error Control Field, in polynomial form, with the first bit transferred being the most significant bit C_0* taken as the coefficient of the highest power of X; and

S(X) is the syndrome polynomial which will be zero if no error is detected and non-zero if an error is detected, with the most significant bit S_0 taken as the coefficient of the highest power of X.

The received block C*(X) equals the transmitted codeblock C(X) plus (modulo two) the n-bit error block E(X), C*(X) = C(X) + E(X), where both are expressed as polynomials of the same form, i.e., with the most significant bit 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 4-5. For each frame, the shift register cells are initialized to '1'. The frame includes n-bits, i.e., (n-16) information message bits plus the 16 bits of the FECF. All the n bits of the frame are clocked into the input and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be 'zero'. A non-zero content indicates an erroneous block.

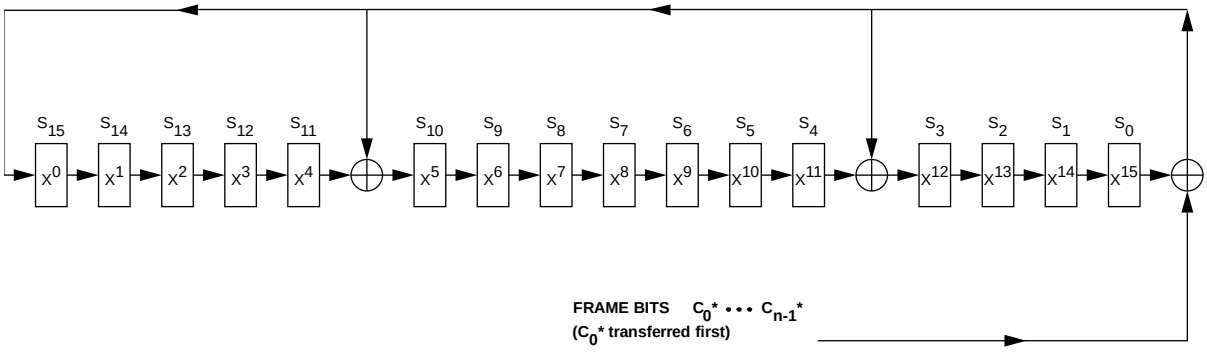

Figure 4-5: Logic Diagram of the Decoder

4.2 PROTOCOL DATA UNIT (CLCW)

4.2.1 COMMUNICATIONS LINK CONTROL WORD

4.2.1.1 General

4.2.1.1.1 The Communications Link Control Word (CLCW), which is the protocol data unit transmitted from the receiving end to the sending end, shall provide the mechanism by which the FARM at the receiving end reports the status of frame acceptance to the Frame Operation Procedure (FOP) at the sending end.

NOTES

1 The controlling specification for how the CLCW is used within the COP is contained in reference [4].

2 CLCWs are usually carried in the Operational Control Field of TM or AOS Transfer Frames (references [B6] and [B7]) using the MC_OCF or VC_OCF Service.

3 The CLCW is the only reporting mechanism for this protocol. Although it is not necessary that the CLCW reporting rate (from the receiving end to the sending end) match the Transfer Frame transfer rate (from the sending end to the receiving end), some minimum CLCW sampling rate is necessary for the proper operation of the COP.

4.2.1.1.2 The CLCW shall consist of ten fields, positioned contiguously, in the following sequence:

a) Control Word Type (1 bit, mandatory);

b) CLCW Version Number (2 bits, mandatory);

c) Status Field (3 bits, mandatory);

d) COP in Effect (2 bits, mandatory);

e) Virtual Channel Identification (6 bits, mandatory);

f) Reserved Spare (2 bits, mandatory);

g) Flags (5 bits, mandatory);

h) FARM-B Counter (2 bits, mandatory);

i) Reserved Spare (1 bit, mandatory);

j) Report Value (8 bits, mandatory).

NOTE – The structural components of the CLCW are shown in figure 4-6.

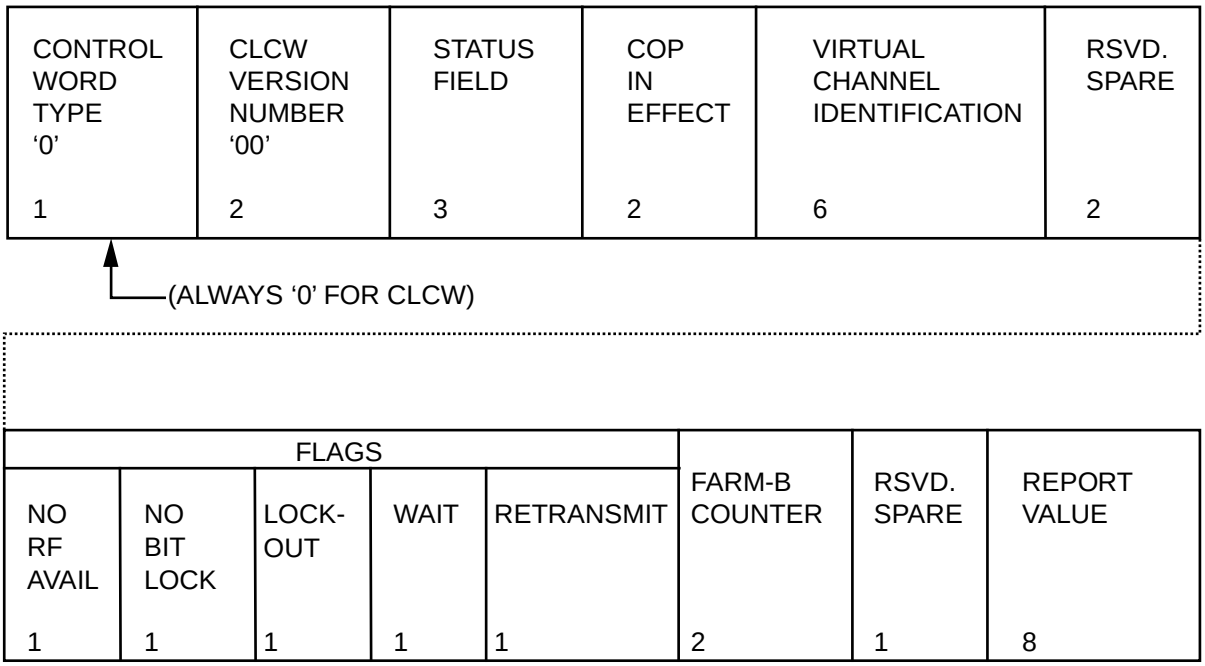

Figure 4-6: Communications Link Control Word

4.2.1.2 Control Word Type

4.2.1.2.1 Bit 0 of the CLCW shall contain the Control Word Type.

NOTE – This field is used to distinguish the CLCW from another type of report that may be alternatively contained in the field that carries the CLCW (e.g., the Operational Control Field of TM or AOS Transfer Frames (references [B6] and [B7]).

4.2.1.2.2 This one-bit field shall be set to '0'.

4.2.1.3 CLCW Version Number

4.2.1.3.1 Bits 1-2 of the CLCW shall contain the (Binary Encoded) CLCW Version Number.

4.2.1.3.2 This two-bit field shall be set to '00'.

NOTE – The CLCW Version Number is included to provide future growth flexibility. At present a single 'Version-1' CLCW, whose binary encoded Version Number is '00', is defined in this Recommended Standard.

4.2.1.4 Status Field

Bits 3-5 of the CLCW shall contain the Status Field.

NOTES

1 Application of the Status Field is mission-specified.

2 The Status Field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

4.2.1.5 COP in Effect

4.2.1.5.1 Bits 6-7 of the CLCW shall contain the COP in Effect parameter and shall be used to indicate the COP that is being used.

4.2.1.5.2 For COP-1, this two-bit field shall be set to '01'.

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

4.2.1.6 Virtual Channel Identification

Bits 8-13 of the CLCW shall contain the Virtual Channel Identifier of the Virtual Channel with which this report is associated.

NOTE – Each Virtual Channel in use has its own CLCW reporting activated.

4.2.1.7 Reserved Spare

4.2.1.7.1 Bits 14-15 of the CLCW shall contain the Reserved Spare.

4.2.1.7.2 These two bits are reserved by CCSDS for future application and shall be set to '00'.

4.2.1.8 Flags

4.2.1.8.1 General

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

4.2.1.8.2 No RF Available Flag

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

4.2.1.8.2.2 The No RF Available Flag shall provide a logical indication of the 'ready' status of the radio frequency (RF) elements within the space link provided by the Physical Layer.

NOTE – Precise definition of the set of physical states which must each be in the 'ready' condition before communication is possible is mission-specified. For example, the flag can represent a logical sum of the overall ready status of components such as the RF transponder and the demodulator.

4.2.1.8.2.3 A setting of '0' in the No RF Available Flag shall indicate that the Physical Layer is Available (i.e., any Transfer Frame will be received and processed by the Physical Layer and passed on to this protocol if correct).

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

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

NOTE – This field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

4.2.1.8.3 No Bit Lock Flag

4.2.1.8.3.1 Bit 17 of the CLCW shall contain the No Bit Lock Flag.

NOTES

1 The No Bit Lock Flag is an optional, mission-specific engineering measurement that provides a performance quality indicator that indicates specifically whether the Physical Layer is working normally by having enough signal energy to achieve bit synchronization with the received data stream.

2 Failure to achieve bit lock may indicate that the Physical Layer is operating at a non-nominal performance level and that the Transfer Frame rejection rate may be correspondingly abnormally high.

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

a) '0' shall indicate bit lock has been achieved;

b) '1' shall indicate bit lock has not been achieved.

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

4.2.1.8.3.4 If the No Bit Lock Flag is not used, it shall be set permanently to '0'.

NOTE – This field may be used by Agencies for local enhancements to operations of this protocol and is not part of the COP.

4.2.1.8.4 Lockout Flag

4.2.1.8.4.1 Bit 18 of the CLCW shall contain the Lockout Flag.

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

4.2.1.8.4.3 A setting of '1' in the Lockout Flag shall indicate Lockout.

NOTE – Lockout occurs whenever a Type-A Transfer Frame that violates certain Frame Acceptance Checks is received on a particular Virtual Channel. Once the FARM is in Lockout, all subsequent Type-A Transfer Frames will be rejected by the FARM until the condition is cleared.

4.2.1.8.4.4 A setting of '0' in the Lockout Flag shall indicate that the FARM is not in Lockout.

4.2.1.8.4.5 Separate Lockout Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Lockout Flag are contained in reference [4].

4.2.1.8.5 Wait Flag

4.2.1.8.5.1 Bit 19 of the CLCW shall contain the Wait Flag.

4.2.1.8.5.2 The Wait Flag shall be used to indicate that the receiving end is unable to accept data for processing on a particular Virtual Channel.

NOTE – An inability to accept data could be caused by temporary lack of storage and/or processing resources in the receiving end of this protocol or higher layers.

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

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

4.2.1.8.5.5 Separate Wait Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Wait Flag are contained in reference [4].

4.2.1.8.6 Retransmit Flag

4.2.1.8.6.1 Bit 20 of the CLCW shall contain the Retransmit Flag.

NOTE – The Retransmit Flag is used to speed the operation of the COP by providing immediate indication to the FOP at the sending end that retransmission is necessary.

4.2.1.8.6.2 A setting of '1' in the Retransmit Flag shall indicate that one or more Type-A Transfer Frames on a particular Virtual Channel have been rejected or found missing by the FARM and therefore retransmission is required.

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

4.2.1.8.6.4 Separate Retransmit Flags shall be maintained for each Virtual Channel.

NOTE – The precise specifications for use of the Retransmit Flag are contained in reference [4].

4.2.1.9 FARM-B Counter

4.2.1.9.1 Bits 21-22 of the CLCW shall contain the FARM-B Counter.

4.2.1.9.2 Separate FARM-B Counters shall be maintained for each Virtual Channel.

NOTE – This 2-bit field contains the two least significant bits of a FARM-B Counter. This counter is maintained within the FARM and increments once each time a Type-B Transfer Frame is accepted in Bypass mode on a particular Virtual Channel. The field supports the verification that Type-B Transfer Frames (Control or User Data) were accepted by the receiving end.

4.2.1.10 Reserved Spare

4.2.1.10.1 Bit 23 of the CLCW shall contain the Reserved Spare.

4.2.1.10.2 This bit is reserved by CCSDS for future application and shall be set to '0'.

4.2.1.11 Report Value

4.2.1.11.1 Bits 24-31 of the CLCW shall contain the Report Value.

4.2.1.11.2 Separate Report Values shall be maintained for each Virtual Channel.

NOTE – This 8-bit field contains the value of the Next Expected Frame Sequence Number, N(R), which is equal to the value of FARM's Receiver_Frame_Sequence_Number, V(R). The FARM V(R) counter increments once each time a Type-AD Transfer Frame is accepted on a particular Virtual Channel. The precise specifications for use of the Report Value are contained in reference [4].

4.3 PROTOCOL PROCEDURES AT THE SENDING END

NOTE – This subsection describes procedures at the sending end associated with each of the functions shown in figure 4-7. In the figure, data flows 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. This figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all of the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.

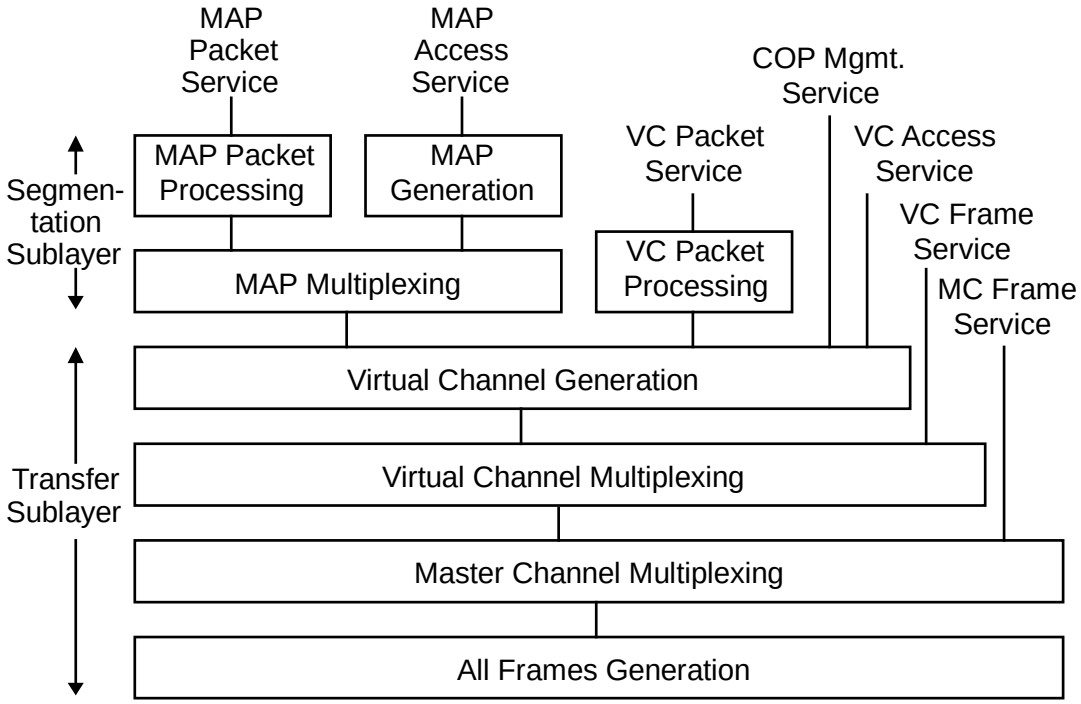

Figure 4-7: Internal Organization of Protocol Entity (Sending End)

4.3.1 MAP PACKET PROCESSING FUNCTION

4.3.1.1 The MAP Packet Processing Function shall be used to transfer variable-length Packets in the Data Field of Transfer Frames (i.e., in Frame Data Units) of a MAP Channel.

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

4.3.1.2 If the Packet to be transferred exceeds a predetermined length, the MAP Packet Processing Function shall divide it into portions that are compatible with insertion into the Frame Data Unit and attach a Segment Header to each portion, forming Frame Data Units.

4.3.1.3 The first octet of the Packet shall appear, without gap, after the Segment Header of the first corresponding Frame Data Unit.

4.3.1.4 The Frame Data Units containing the first and continuing portions of the Packet may each have a length equal to the maximum allowable length of the Frame Data Unit on that particular MAP Channel.

4.3.1.5 The Frame Data Unit containing the last portion of the Packet shall contain the Segment Header and the residue of the Packet.

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

4.3.1.7 If the Packet to be transferred does not exceed a predetermined length, a Segment Header is generated and attached to the Packet, forming a Frame Data Unit.

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

a) multiple complete Packets may be placed into a Frame Data Unit with a single Segment Header preceding them;

b) the blocked Packets plus the Segment Header must fit within the maximum size Frame Data Unit permitted for the MAP Channel.

4.3.1.9 If Packets of multiple versions are to be transferred on a MAP Channel, Packets of these versions are multiplexed into a contiguous string of Packets before they are placed in Frame Data Units.

NOTES

1 An abstract model of the MAP Packet Processing Function is illustrated in figure 4-8.

2 Figure 4-9 shows how Packets assigned to one MAP are segmented or blocked to form Frame Data Units.

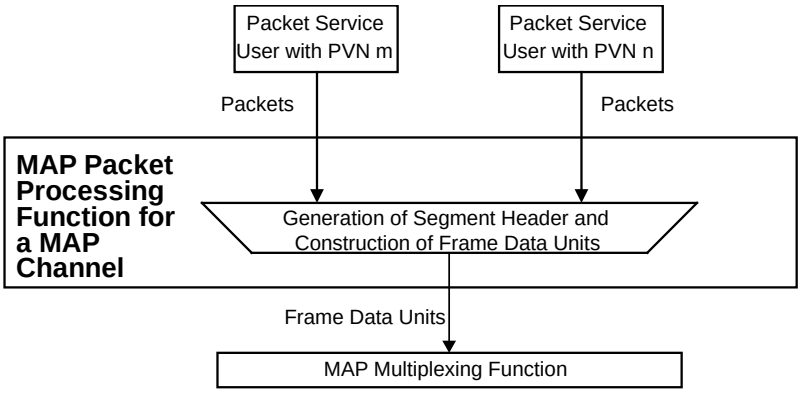

Figure 4-8: Abstract Model of MAP Packet Processing Function

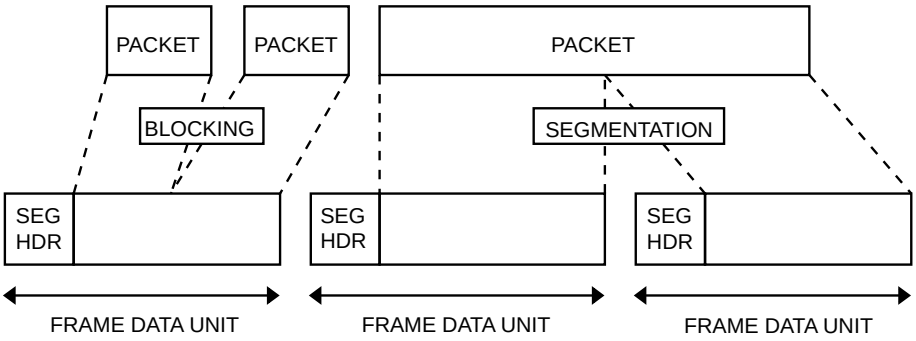

Figure 4-9: Example of MAP Packet Processing Procedures

4.3.2 MAP GENERATION FUNCTION

4.3.2.1 The MAP Generation Function shall be used to transfer variable-length user-defined service data units (MAP_SDUs) in the Data Field of Transfer Frames (i.e., in Frame Data Units) of a MAP Channel.

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

4.3.2.2 If the MAP_SDU to be transferred exceeds a predetermined length, the MAP Generation Function shall divide it into portions that are compatible with insertion into the Frame Data Unit and attach a Segment Header to each portion, forming Frame Data Units.

4.3.2.3 The first octet of the MAP_SDU shall appear, without gap, after the Segment Header of the first corresponding Frame Data Unit.

4.3.2.4 The Frame Data Units containing the first and continuing portions of the MAP_SDU may each have a length equal to the maximum allowable length of the Frame Data Unit on that particular MAP Channel.

4.3.2.5 The Frame Data Unit containing the last portion of the MAP_SDU shall contain the Segment Header and the residue of the MAP_SDU.

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

4.3.2.7 If the MAP_SDU to be transferred does not exceed a predetermined length, a Segment Header shall be generated and attached to the MAP_SDU, forming a Frame Data Unit.

NOTES

1 An abstract model of the MAP Generation Function is illustrated in figure 4-10.

2 Figure 4-11 shows how MAP_SDUs assigned to one MAP are processed to form Frame Data Units.

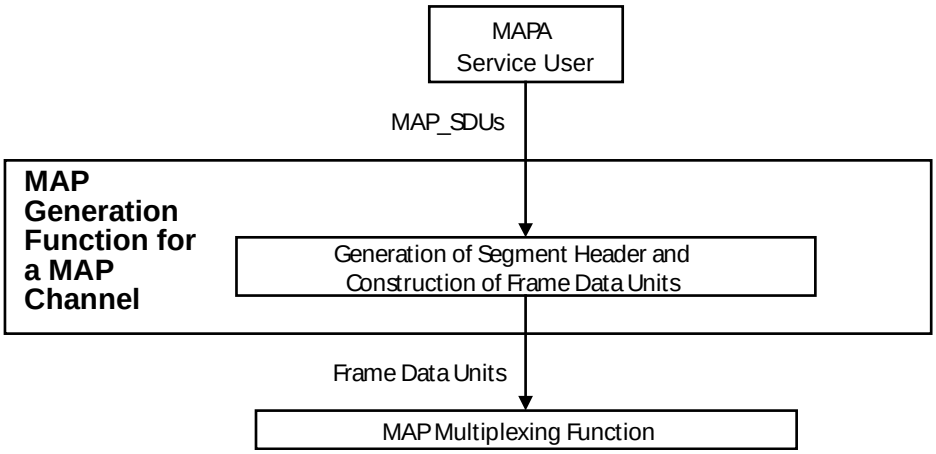

Figure 4-10: Abstract Model of MAP Generation Function

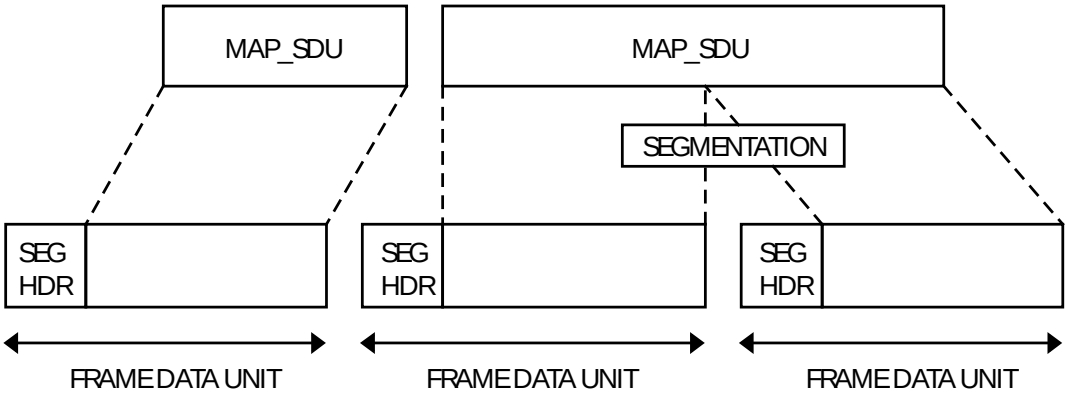

Figure 4-11: Example of MAP Generation Procedures

4.3.3 MAP MULTIPLEXING FUNCTION

4.3.3.1 The MAP Multiplexing Function shall be used to multiplex Frame Data Units (each containing a Segment Header) of different MAP Channels of a Virtual Channel.

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

4.3.3.2 The MAP Multiplexing Function shall multiplex Frame Data Units received from the instances of the MAP Packet Processing and/or MAP Generation Functions and put them into a queue of Frame Data Units in an appropriate order set by management.

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

NOTE – An abstract model of the MAP Multiplexing Function is illustrated in figure 4-12.

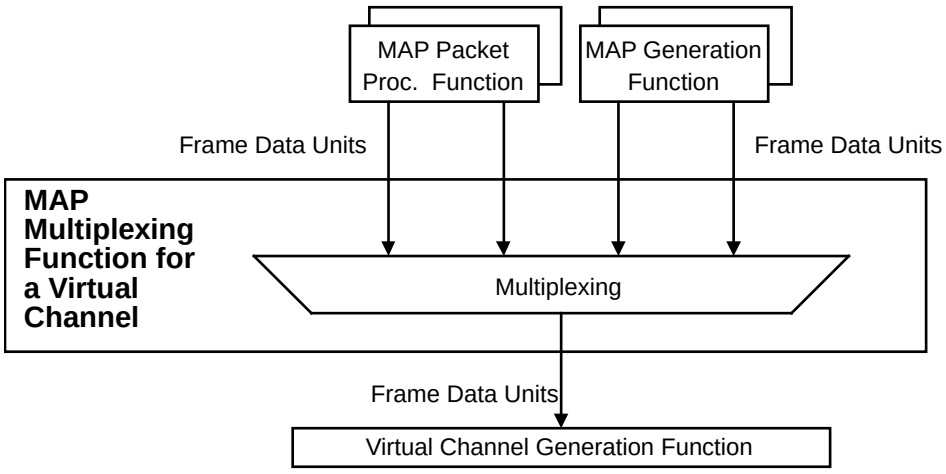

Figure 4-12: Abstract Model of MAP Multiplexing Function

4.3.4 VC PACKET PROCESSING FUNCTION

4.3.4.1 The VC Packet Processing Function shall be used to transfer variable-length Packets in the Data Field of Transfer Frames (i.e., in Frame Data Units) of a Virtual Channel.

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

4.3.4.2 The VC Packet Processing Function shall generate Frame Data Units, each containing one or multiple complete Packets (without a Segment Header).

4.3.4.3 If blocking of Packets is permitted on a particular Virtual Channel,

a) multiple complete Packets may be placed into a Frame Data Unit;

b) the blocked Packets must fit within the maximum size Frame Data Unit permitted for the Virtual Channel.

4.3.4.4 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 they are placed in Frame Data Units.

NOTES

1 An abstract model of the VC Packet Processing Function is illustrated in figure 4-13.

2 Figure 4-14 shows how Packets assigned to one Virtual Channel are processed to form Frame Data Units.

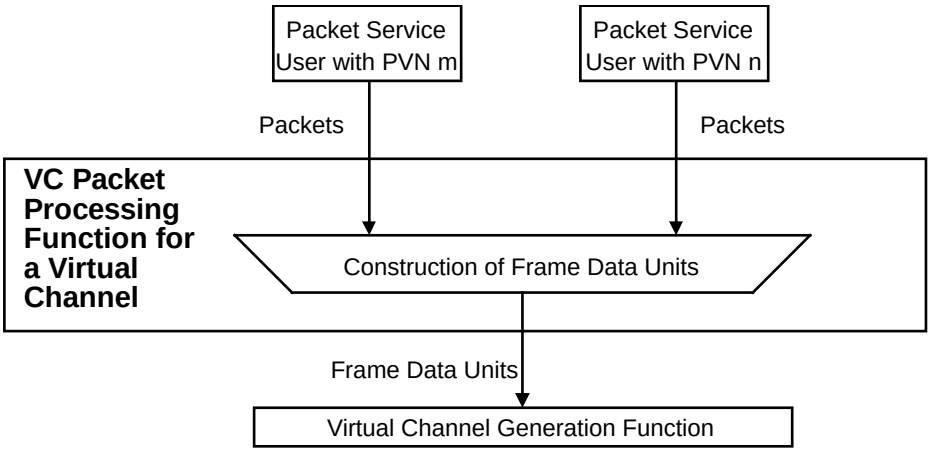

Figure 4-13: Abstract Model of VC Packet Processing Function

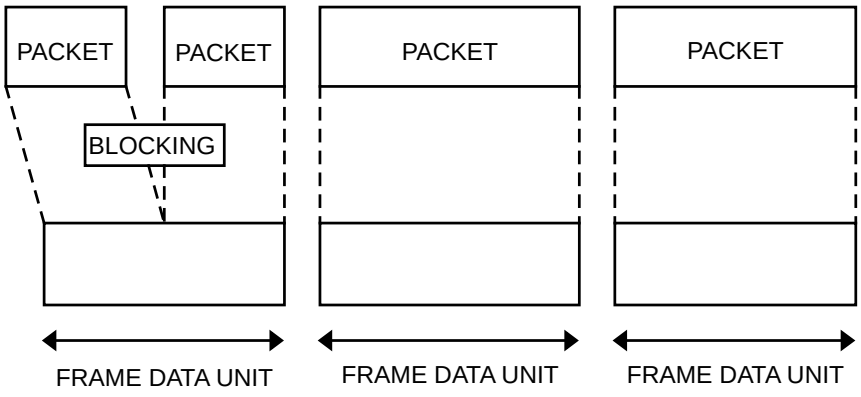

Figure 4-14: Example of VC Packet Processing Procedures

4.3.5 VIRTUAL CHANNEL GENERATION FUNCTION

NOTE – The Virtual Channel Generation Function is the 'heart' of this protocol. It builds Transfer Frames and performs most of the operations required to move service data units reliably from the sending end to the receiving end of the protocol. There is an instance of the Virtual Channel Generation Function for each Virtual Channel.

4.3.5.1 The Virtual Channel Generation Function shall perform the following two procedures in the following order:

a) the Frame Operation Procedure (FOP), which is a sub-procedure of the Communications Operation Procedure (COP); and

b) the Frame Generation Procedure in this order.

4.3.5.2 The FOP shall accept Frame Data Units (FDUs) from the MAP Multiplexing Function, the VC Packet Processing Function, or a VCA Service User (one VCA_SDU is treated as one FDU) and shall control transmission and retransmission of FDUs by examining the report contained in the CLCWs and generating Control Commands.

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

NOTE – The detailed specification of the FOP is given in reference [4].

4.3.5.4 The Frame Generation Procedure shall generate Transfer Frames by attaching a Transfer Frame Primary Header to each Frame Data Unit or Control Command delivered by the FOP.

NOTE – An abstract model of the Virtual Channel Generation Function is illustrated in figure 4-15.

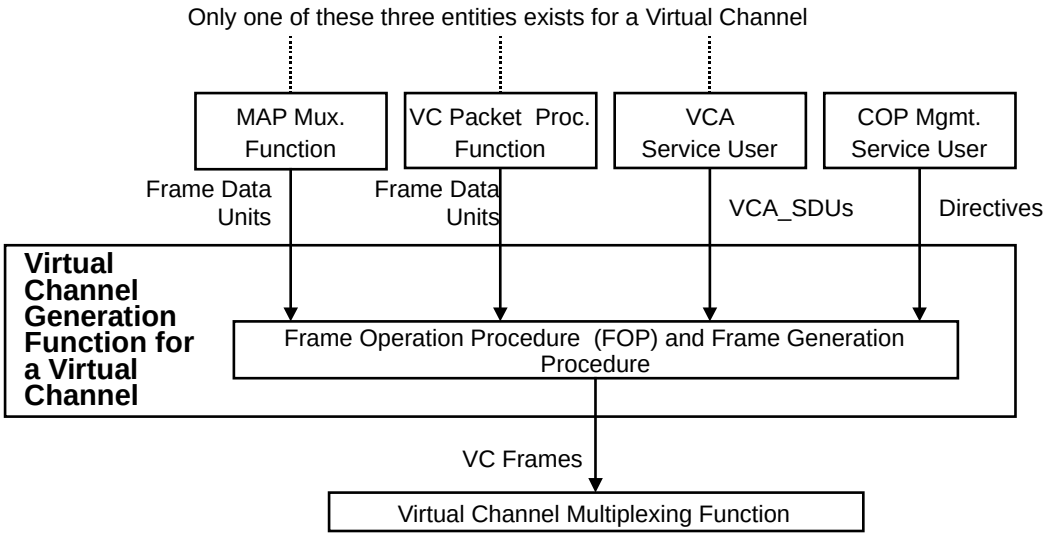

Figure 4-15: Abstract Model of Virtual Channel Generation Function

4.3.6 VIRTUAL CHANNEL MULTIPLEXING FUNCTION

4.3.6.1 The Virtual Channel Multiplexing Function shall be used to multiplex Transfer Frames from different Virtual Channels on a Master Channel.

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

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

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

4.3.6.3 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.

NOTE – An abstract model of the Virtual Channel Multiplexing Function is illustrated in figure 4-16.

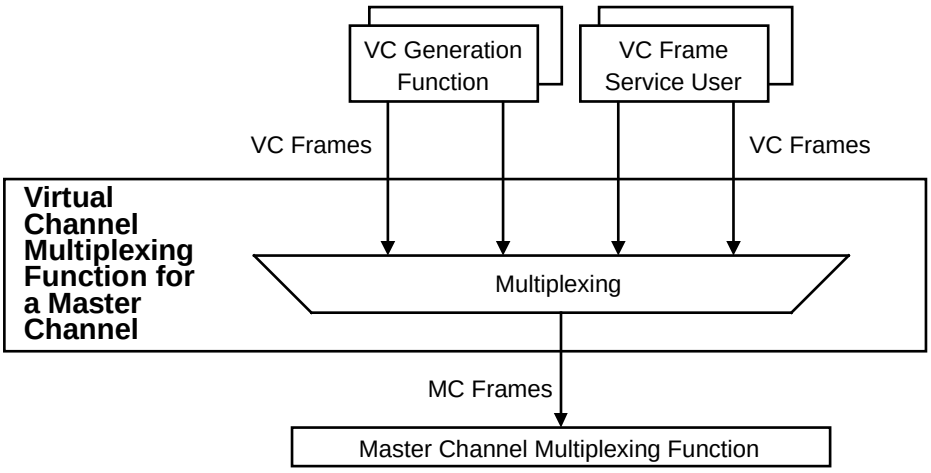

Figure 4-16: Abstract Model of Virtual Channel Multiplexing Function

4.3.7 MASTER CHANNEL MULTIPLEXING FUNCTION

4.3.7.1 The Master Channel Multiplexing Function shall be used to multiplex Transfer Frames from different Master Channels on a Physical Channel.

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

4.3.7.2 The Master Channel Multiplexing Function shall multiplex Transfer Frames received from the instances of the Master Channel Generation Function and, if present, the Master Channel Frame Service users, and shall put them into a queue of Transfer Frames in an appropriate order set by management.

4.3.7.3 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.

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

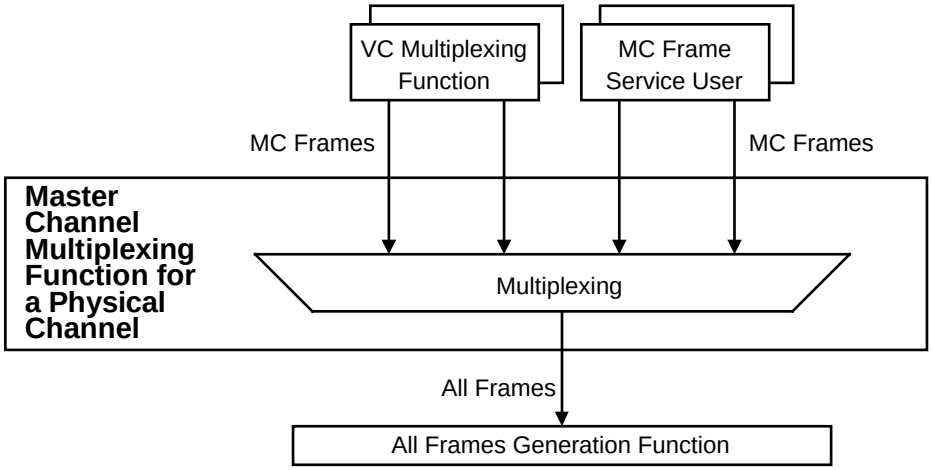

Figure 4-17: Abstract Model of Master Channel Multiplexing Function

4.3.8 ALL FRAMES GENERATION FUNCTION

4.3.8.1 The All Frames Generation Function shall be used to perform error control encoding defined by this Recommended Standard and to deliver Transfer Frames at an appropriate rate to the Channel Coding Sublayer.

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

4.3.8.2 If the Frame Error Control Field is present, then:

a) check bits shall be generated using the encoding procedure described in 4.1.4.2 and inserted into the Transfer Frame Trailer;

b) the Frame Error Control Field shall be present in all the Transfer Frames transmitted in a particular Physical Channel.

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

4.3.8.4 Each data unit delivered by the All Frames Generation Function shall consist of one or more Transfer Frames in accordance with parameters set by management.

4.3.8.5 In accordance with parameters set by management, the All Frames Generation Function may request the Synchronization and Channel Coding Sublayer to perform systematic retransmissions of a data unit as described in 2.4.2, unless the data unit contains one or more frames carrying service data on the Type-B Service.

4.3.8.6 The All Frames Generation Function shall deliver data units at a rate specified by management.

NOTES

1 When systematic retransmissions of a data unit are requested, the improved reception probabilities from the retransmissions are increased if the data unit consists of a single Transfer Frame.

2 When systematic retransmissions of a data unit are requested, the additional delay for the retransmissions can be taken into account when deciding the delivery time for the following data unit.

3 An abstract model of the All Frames Generation Function is illustrated in figure 4-18.

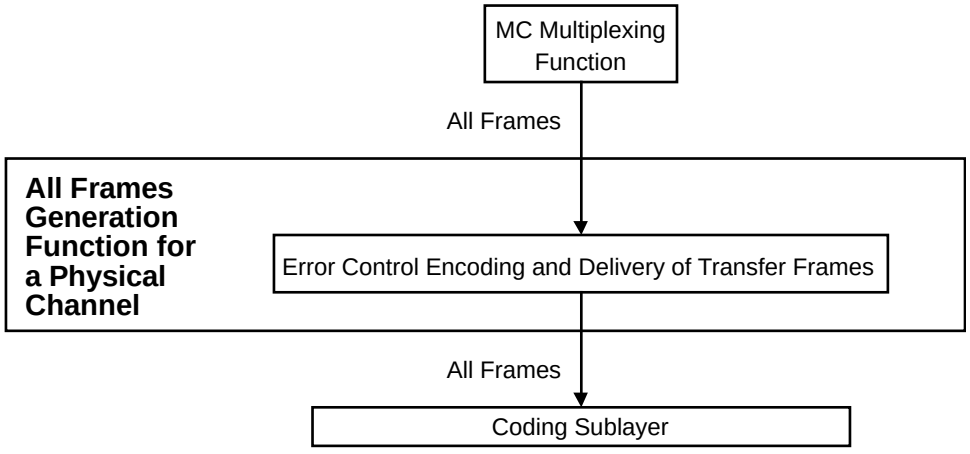

Figure 4-18: Abstract Model of All Frames Generation Function

4.4 PROTOCOL PROCEDURES AT THE RECEIVING END

NOTE – This subsection describes procedures at the receiving end associated with each of the functions shown in figure 4-19. 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. This figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all of the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.

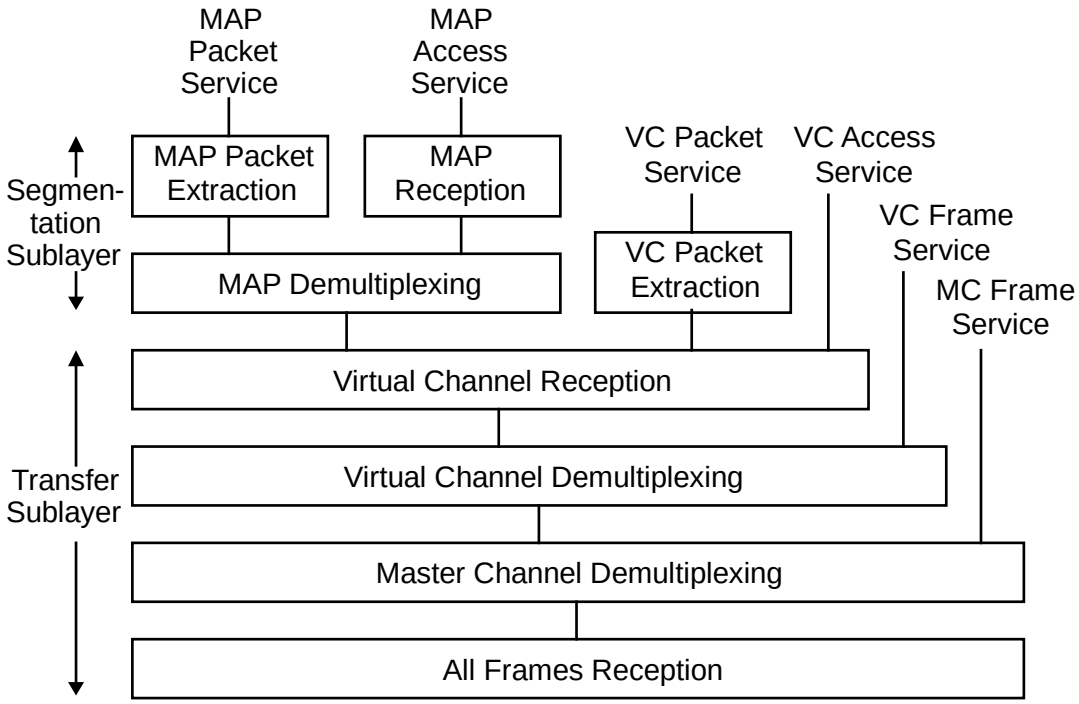

Figure 4-19: Internal Organization of Protocol Entity (Receiving End)

4.4.1 MAP PACKET EXTRACTION FUNCTION

4.4.1.1 The MAP Packet Extraction Function shall be used to extract variable-length Packets from Frame Data Units on a MAP Channel.

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

4.4.1.2 The MAP Packet Extraction Function shall extract Packets from Frame Data Units received from the MAP Demultiplexing Function.

4.4.1.3 Original Packets shall be extracted and reconstructed from Frame Data Units using the Sequence Flag of the Segment Header of each Frame Data Unit and, if blocking of Packets is permitted, the length field of each Packet.

4.4.1.4 Extracted Packets shall be delivered to the users on the basis of the PVN in their header.

4.4.1.5 Incomplete Packets are not required to be delivered in cross support situations.

NOTE – An abstract model of the MAP Packet Extraction Function is illustrated in figure 4-20.

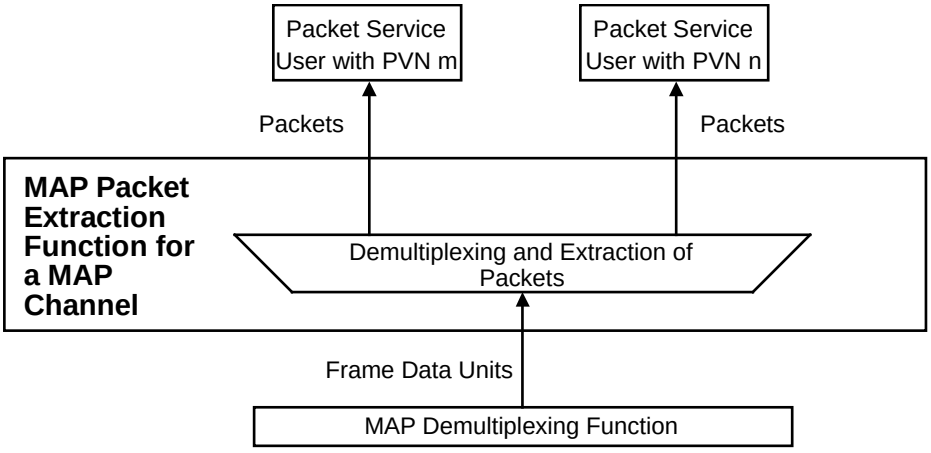

Figure 4-20: Abstract Model of MAP Packet Extraction Function

4.4.2 MAP RECEPTION FUNCTION

4.4.2.1 The MAP Reception Function shall be used to extract variable-length service data units (MAP_SDUs) from Frame Data Units on a MAP Channel.

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

4.4.2.2 The MAP Reception Function shall extract MAP_SDUs from Frame Data Units received from the MAP Demultiplexing Function.

4.4.2.3 Original MAP_SDUs shall be extracted and reconstructed from Frame Data Units using the Sequence Flag of the Segment Header of each Frame Data Unit.

4.4.2.4 Extracted MAP_SDUs shall be delivered to the user.

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

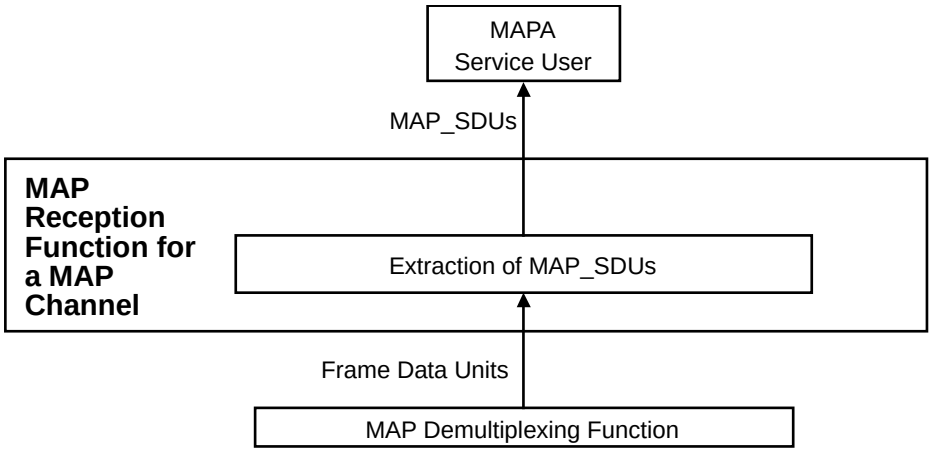

Figure 4-21: Abstract Model of MAP Reception Function

4.4.3 MAP DEMULTIPLEXING FUNCTION

4.4.3.1 The MAP Demultiplexing Function shall be used to demultiplex Frame Data Units from different MAP Channels on a Virtual Channel.

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

4.4.3.2 The MAP Demultiplexing Function shall examine the MAP ID in the Segment Header of the incoming Frame Data Units and shall route them to the instances of the MAP Packet Extraction and/or MAP Reception Functions.

4.4.3.3 Frame Data Units with an invalid MAP ID shall be discarded.

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

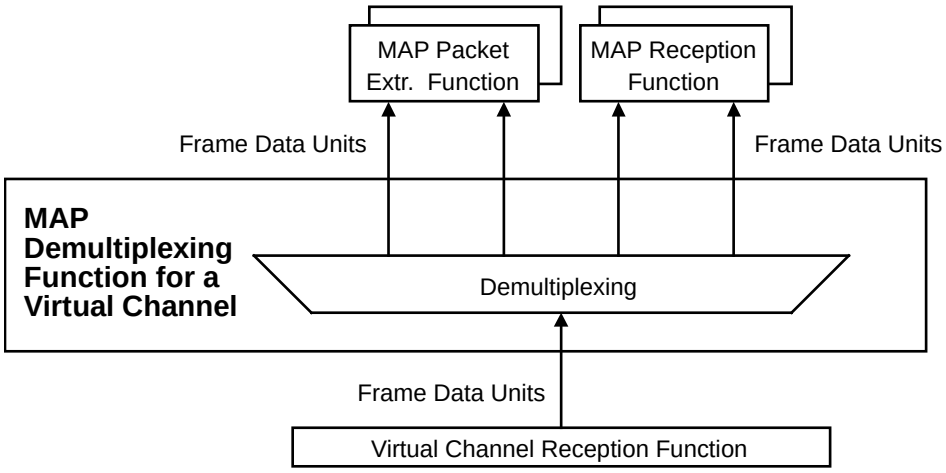

Figure 4-22: Abstract Model of MAP Demultiplexing Function

4.4.4 VC PACKET EXTRACTION FUNCTION

4.4.4.1 The VC Packet Extraction Function shall be used to extract variable-length Packets from the Frame Data Units on a Virtual Channel.

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

4.4.4.2 The VC Packet Extraction Function shall extract Packets from Frame Data Units received from the Virtual Channel Reception Function.

4.4.4.3 If blocking of Packets is permitted, Packets shall be extracted from Frame Data Units using the length field of each Packet.

4.4.4.4 Extracted Packets shall be delivered to the users on the basis of the PVN in their header.

NOTE – An abstract model of the VC Packet Extraction Function is illustrated in figure 4-23.

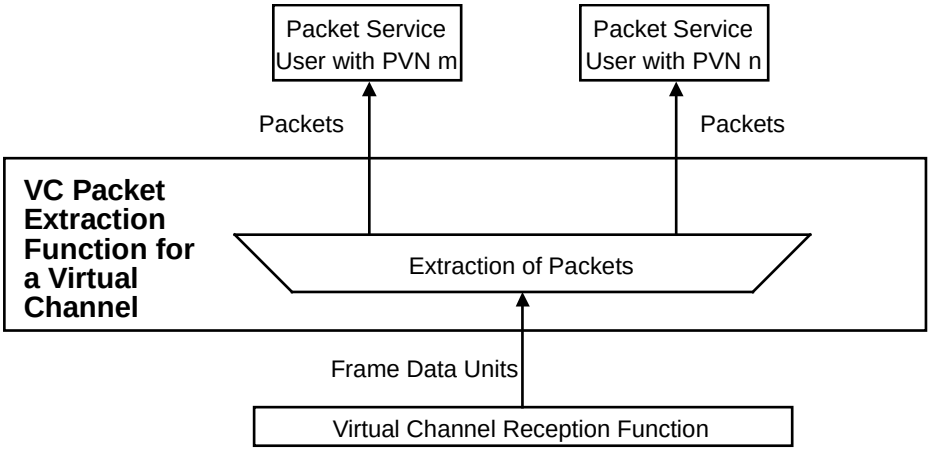

Figure 4-23: Abstract Model of VC Packet Extraction Function

4.4.5 VIRTUAL CHANNEL RECEPTION FUNCTION

4.4.5.1 The Virtual Channel Reception Function shall perform the Frame Acceptance and Reporting Mechanism (FARM), which is a sub-procedure of the Communications Operation Procedure (COP).

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

4.4.5.2 The FARM shall examine the Primary Header of the incoming Transfer Frames, perform Frame Acceptance Checks against Type-A Transfer Frames, execute Control

Commands, and generate some parameters to be transferred back to the sending end with CLCWs.

4.4.5.3 Frame Data Units extracted from Type-A Transfer Frames shall be delivered to their users (or Functions) only if they have passed the Frame Acceptance Checks.

4.4.5.4 Frame Data Units extracted from Type-B Transfer Frames shall be delivered to their users (or Functions) immediately.

NOTES

1 The detailed specification of the FARM is given in reference [4].

2 An abstract model of the Virtual Channel Reception Function is illustrated in figure 4-24.

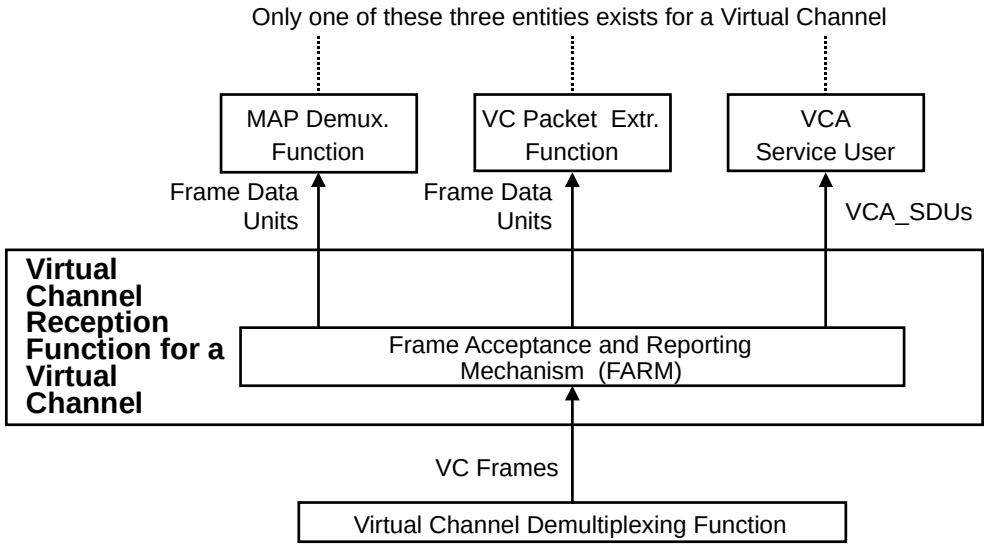

Figure 4-24: Abstract Model of Virtual Channel Reception Function

4.4.6 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION

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

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

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

4.4.6.3 Transfer Frames with an invalid VCID shall be discarded.

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

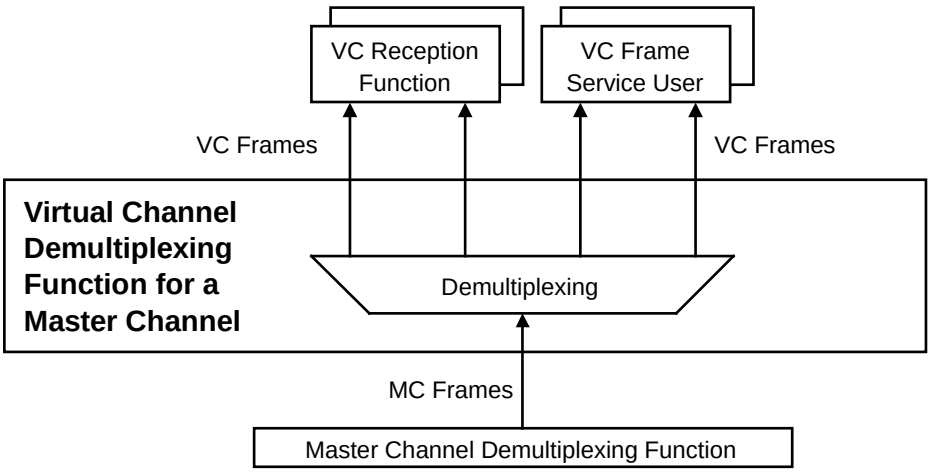

Figure 4-25: Abstract Model of Virtual Channel Demultiplexing Function

4.4.7 MASTER CHANNEL DEMULTIPLEXING FUNCTION

4.4.7.1 The Master Channel Demultiplexing Function shall be used to demultiplex Transfer Frames from different Master Channels on a Physical Channel.

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

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

NOTE – An abstract model of the Master Channel Demultiplexing Function is illustrated in figure 4-26.

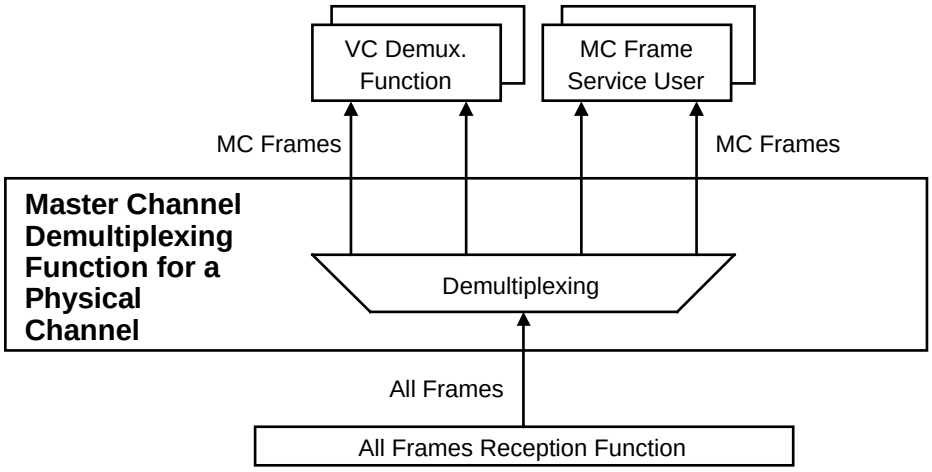

Figure 4-26: Abstract Model of Master Channel Demultiplexing Function

4.4.8 ALL FRAMES RECEPTION FUNCTION

4.4.8.1 General

4.4.8.1.1 The All Frames Reception Function shall be used to reconstitute Transfer Frames from the data stream provided by the Channel Coding Sublayer and to perform checks to determine whether the reconstituted Transfer Frames are valid or not.

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

4.4.8.1.2 The All Frames Reception Function shall perform two procedures:

a) Frame Delimiting and Fill Removal Procedure; and

b) Frame Validation Check Procedure, in this order.

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

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

NOTE – An abstract model of the All Frames Reception Function is illustrated in figure 4-27.

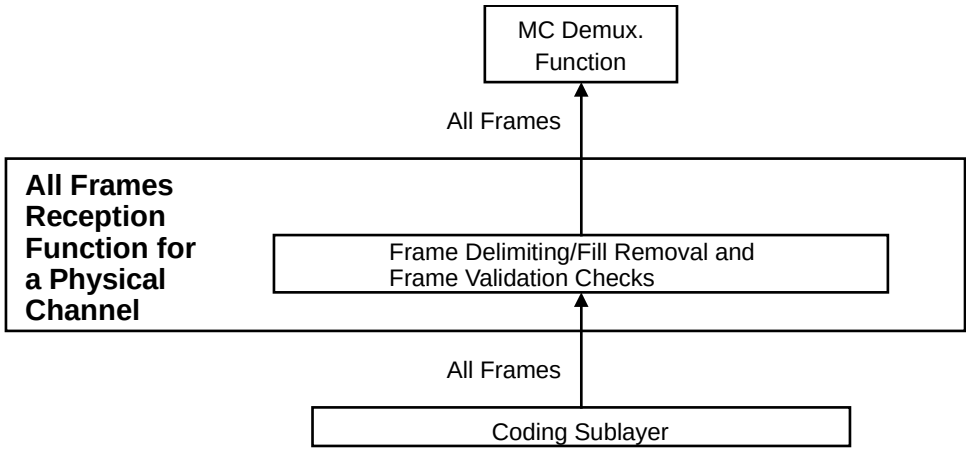

Figure 4-27: Abstract Model of All Frames Reception Function

4.4.8.2 Frame Delimiting and Fill Removal Procedure

4.4.8.2.1 The Channel Coding and Synchronization Recommended Standard (reference [3]) shall be used as the specification for the Channel Coding Sublayer immediately below this protocol.

4.4.8.2.2 At the sending end, the TC Space Data Link Protocol shall pass one or more Transfer Frames to the Channel Coding Sublayer.

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

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

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

4.4.8.2.4 If one or more valid Frame Length fields are detected by this Procedure and the number of octets received when the Data Stop condition occurs equals the number of octets specified by the Frame Length(s), then each Transfer Frame shall be passed on to the Frame Validation Check Procedure (see 4.4.8.3) as it is delimited.

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

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

4.4.8.2.6 If a valid Frame Length field is detected by this procedure but the number of octets received when the Data Stop condition occurs is greater than the number of octets specified by that Frame Length, the procedure shall

a) assume that the octets following the final expected octet of the frame are Fill Data appended by the sending end of the Channel Coding Sublayer to complete the last Codeblock (see reference [3]);

b) discard that Fill Data;

c) pass the Frame to the Frame Validation Check Procedure (see 4.4.8.3).

NOTES

1 Because the receiving end of the Channel Coding Sublayer cannot distinguish between valid data and Fill Data, the Fill Data must be stripped by this protocol.

2 The characteristics of the present Codeblock structure are such that no more than six octets of Fill Data can occur. If fewer than five trailing octets of Fill Data are present, then they cannot possibly form a Transfer Frame Header, and they will be immediately discarded by this Procedure. If five or six trailing octets of Fill Data exist, this procedure might attempt to interpret the Fill Data as a new Transfer Frame Header; however, the subsequent Frame Validation Checks (see 4.4.8.3) 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, thus failing a test and causing the trailing five or six octets to be discarded.

3 After each Transfer Frame is reconstituted by the Frame Delimiting and Fill Removal Procedure, it will next be subjected to a set of standard tests called Frame Validation Checks.

4.4.8.3 Frame Validation Check Procedure

4.4.8.3.1 The Frame Validation Checks shall be applied to all incoming Transfer Frames, regardless of whether they are Type-A or Type-B.

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

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

a) The Transfer Frame must have an expected Transfer Frame Version Number.

b) The Transfer Frame must have one of the expected MCIDs (Transfer Frame Version Number and Spacecraft IDs).

c) The Transfer Frame Header must not contain any values which are not consistent with the implemented features for that spacecraft.

d) The value of the Frame Length must be consistent with the number of octets that are present.

e) If the Frame Error Control Field is present, the recomputed CRC value for the Transfer Frame must match the content of the Frame Error Control Field.

5 MANAGED PARAMETERS WITHOUT SDLS OPTION

NOTES

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

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

3 This section specifies managed parameters for the TC Space Data Link Protocol without support for the SDLS protocol. Additional managed parameters for the TC Space Data Link Protocol with the SDLS option are specified in 6.6.

5.1 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
Maximum Transfer Frame Length (octets)	Integer
Transfer Frame Version Number	1
Valid Spacecraft IDs	Set of Integers
MC Multiplexing Scheme	Mission Specific
Presence of Frame Error Control	Present, Absent
Maximum Number of Transfer Frames Given to the Coding Sublayer as a Single Data Unit	Integer
Maximum Length of Data Unit Given to the Coding Sublayer	Integer
Maximum Bit Rate Accepted by the Coding Sublayer	Real number/second
Maximum value for the Repetitions parameter to the Coding Sublayer	Integer

5.2 MANAGED PARAMETERS FOR A MASTER CHANNEL

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

Table 5-2: Managed Parameters for a Master Channel

Managed Parameter	Allowed Values
Maximum Transfer Frame Length (octets)	Integer (up to 1024)
Spacecraft ID	Integer
Valid VCIDs	Set of Integers (from 0 to 63)
VC Multiplexing Scheme	Mission Specific
NOTE – The value of the Transfer Frame Version Number is the same for all Transfer Frames on a Physical Channel.	

5.3 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

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

NOTE – The managed parameters associated with the COP are listed in reference [4].

Table 5-3: Managed Parameters for a Virtual Channel

Managed Parameter	Allowed Values
Maximum Transfer Frame Length (octets)	Integer (up to 1024)
Spacecraft ID	Integer
VCID	0, 1, …, 63
COP in Effect	1
CLCW Version Number	1
CLCW Reporting Rate	Real number/Second
Presence of Segment Header	Present, Absent
Data Field Content (if Segment Header is absent)	Packets, VCA_SDU
Valid MAP IDs (if Segment Header is present)	Set of integers (from 0 to 63)
MAP Multiplexing Scheme (if Segment Header is present)	Mission Specific
Blocking (if Segment Header is absent and Data Field Content is Packets)	Permitted, Prohibited
Value for the Repetitions parameter to the Coding Sublayer when transferring frames carrying service data on the Type-A Service	Integer
Value for the Repetitions parameter to the Coding Sublayer when transferring frames carrying COP control commands	Integer
NOTE – The value of the Transfer Frame Version Number is the same for all Transfer Frames on a Physical Channel.	

5.4 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
Maximum Frame Data Unit Length (octets)	Integer (up to 1019)
Spacecraft ID	Integer
VCID	0, 1, …, 63
MAP ID	0, 1, …, 63
Data Field Content	Packets, MAP_SDU
Blocking (if Data Field Content is Packets)	Permitted, Prohibited
Segmentation	Permitted, Prohibited
Maximum MAP_SDU Length (octet) (if the MAP permits Segmentation)	Integer
NOTE – The value of the Transfer Frame Version Number is the same for all Transfer Frames on a Physical Channel.	

5.5 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 (see reference [5])
Maximum Packet Length (octets)	Integer
Whether incomplete Packets are required to be delivered to the user at the receiving end	Required, Not required

6 PROTOCOL SPECIFICATION WITH SDLS OPTION

6.1 OVERVIEW

This section specifies the protocol data unit and the procedures of the TC Space Data Link Protocol with support for the SDLS protocol (reference [7]). If the TC Space Data Link protocol entity supports SDLS, it has managed parameters for each Virtual Channel 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 [7] is required over the TC space data link, then the SDLS protocol shall be used.

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

6.3 TC TRANSFER FRAME WITH SDLS

6.3.1 OVERVIEW

Additional fields are defined for a TC Transfer Frame to support the use of the SDLS security features in the frame. This subsection specifies the TC Transfer Frames on a Virtual Channel or MAP that is using SDLS.

The use of SDLS can vary between Virtual Channels and between MAPs on a Virtual Channel, and the use is indicated by managed parameters (see 6.6). If a Virtual Channel or MAP is not using SDLS, then the frames are as specified in 4.1.

There are two fields for the SDLS protocol: a Security Header and a Security Trailer. In a Type-D frame, these fields are placed before and after the Transfer Frame Data Field, and they reduce the maximum length of the Transfer Frame Data Field compared to a frame without SDLS.

Type-C frames do not have the Security Header and Security Trailer. The Control Command Flag in the Transfer Frame Primary Header distinguishes between a Type-D frame carrying Data and a Type-C frame carrying Control Commands (see 4.1.2.3.2).

For a frame on a Virtual Channel where Segment Headers are present, the Security Header is placed between the Segment Header and the User Data field of the Segment. Figures 6-1 and 6-2 show the fields for frames with and without Segment Headers.

If the Segment Header is present then it is present in all frames for the Virtual Channel.

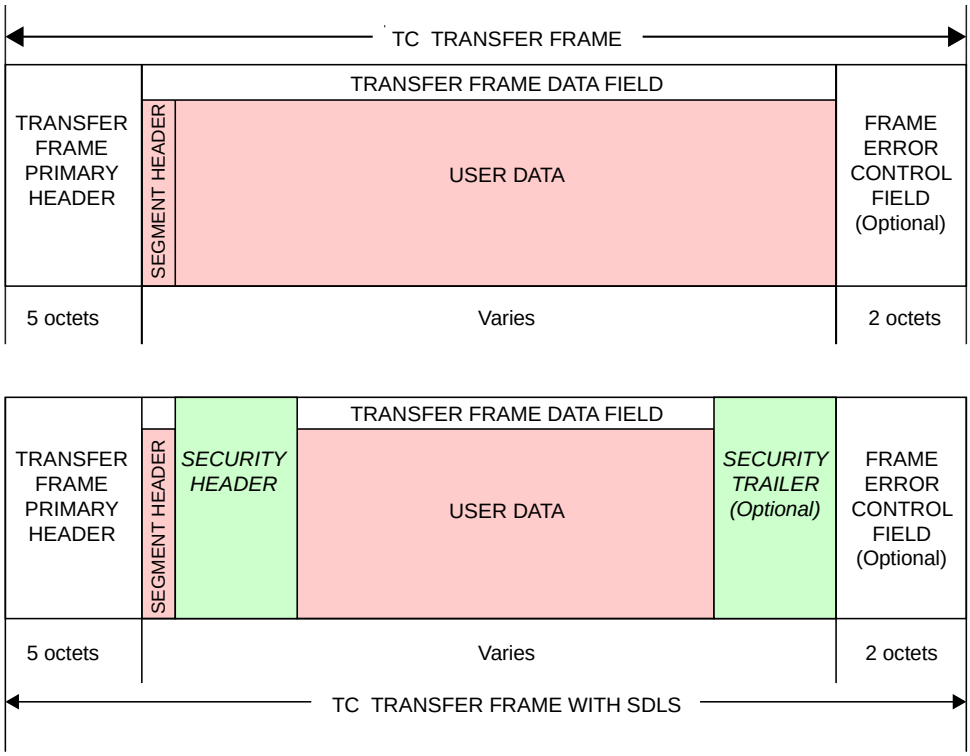

Figure 6-1: SDLS Fields in a Type-D Transfer Frame with a Segment Header

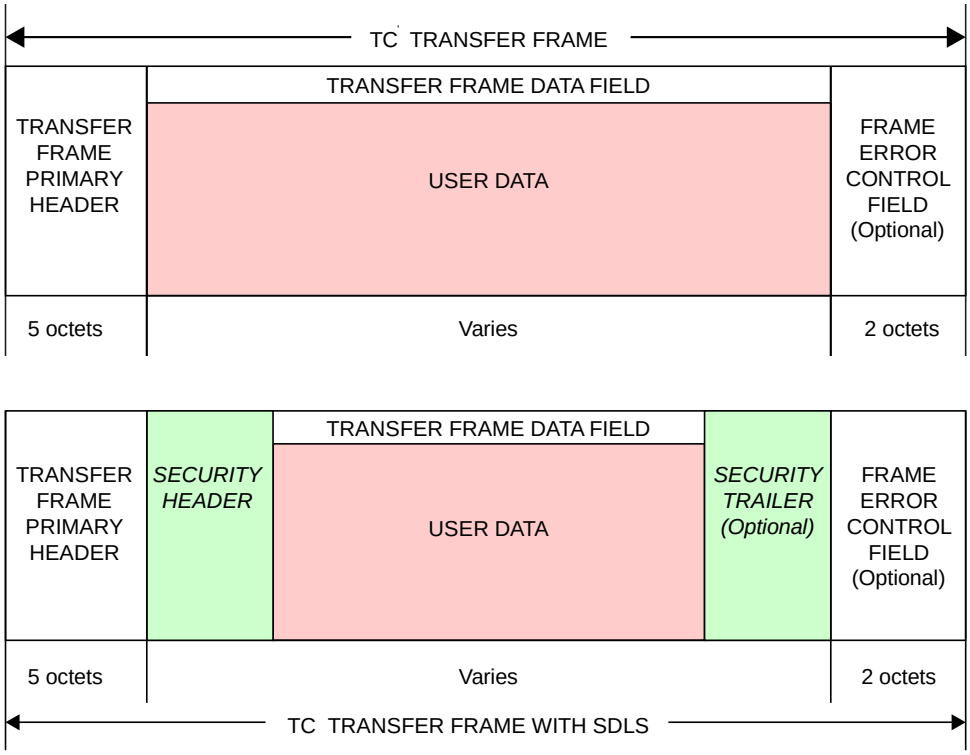

Figure 6-2: SDLS Fields in a Type-D Transfer Frame without a Segment Header

6.3.2 TRANSFER FRAME PRIMARY HEADER IN A FRAME WITH SDLS

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

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

6.3.3 SEGMENT HEADER IN A FRAME WITH SDLS

The Segment Header for a frame with SDLS shall conform to the specifications of 4.1.3.2.1.4 (requirements for the presence of the Segment Header) and 4.1.3.2.2 (requirements for the position and contents of the Segment Header).

NOTES

1 The Segment Header is the same for a frame without SDLS and a frame with SDLS.

2 The presence of the Segment Header is a managed parameter of the Virtual Channel (see 5.3).

6.3.4 SECURITY HEADER

6.3.4.1 If the Control Command Flag in the Transfer Frame Primary Header indicates that the Transfer Frame Data Field contains control information, the Security Header shall not be present in the Transfer Frame.

NOTE – SDLS is not used for frames carrying control information (Type-C frames).

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

NOTES

1 The requirements for the length and contents of the Security Header are in reference [7].

2 For a Transfer Frame with a Segment Header, the presence of the Security Header in Type-D frames is a managed parameter of the MAP. For a Transfer Frame without a Segment Header, the presence of the Security Header in Type-D frames is a managed parameter of the Virtual Channel. If the Security Header is not present, then SDLS is not in use on the MAP or Virtual Channel and the Transfer Frame has the format specified in 4.1.

3 The length of the Security Header is an integral number of octets and is a managed parameter of the MAP or Virtual Channel.

6.3.5 TRANSFER FRAME DATA FIELD IN A FRAME WITH SDLS

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

a) follow, without gap, the Security Header;

b) contain an integral number of octets, which may vary in length up to a maximum number of octets equal to 1024 minus

– the lengths of the Transfer Frame Primary Header and of the Security Header;

– the lengths of the Segment Header, of the Security Trailer, and of the Frame Error Control Field, if any of these are present.

6.3.5.2 In a Transfer Frame with SDLS and with a Segment Header, the Transfer Frame Data Field shall contain the User Data of the Segment whose Segment Header is in the Segment Header field of the frame.

NOTES

1 In this case, the Frame Data Unit consists of a Segment. The Security Header separates the Segment Header from the rest of the Segment as shown in figure 6-1.

2 Segment User Data is defined in 4.1.3.2.1.2.

6.3.5.3 In a Transfer Frame with SDLS and without a Segment Header, the Transfer Frame Data Field shall contain a Frame Data Unit consisting of User Data as defined in 4.1.3.2.1.3.

NOTE – This case is shown in figure 6-2.

6.3.6 SECURITY TRAILER

6.3.6.1 If the Control Command Flag in the Transfer Frame Primary Header indicates that the Transfer Frame Data Field contains control information, the Security Trailer shall not be present in the Transfer Frame.

NOTE – SDLS is not used for frames carrying control information (Type-C frames).

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

NOTES

1 The requirements for the length and contents of the Security Trailer are in reference [7].

2 The Security Trailer is optional in a TC Transfer Frame with SDLS. For a Transfer Frame with a Segment Header, the presence of the Security Trailer in Type-D frames is a managed parameter of the MAP. For a Transfer Frame without a Segment

Header, the presence of the Security Header in Type-D frames is a managed parameter of the Virtual Channel.

3 The length of the Security Trailer is an integral number of octets and is a managed parameter of the MAP or Virtual Channel.

6.3.7 FRAME ERROR CONTROL FIELD IN A FRAME WITH SDLS

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

The Frame Error Control Field of a frame with SDLS shall conform to the specifications of 4.1.4.1.1, 4.1.4.1.3, 4.1.4.2, and 4.1.4.3.

6.4 SENDING-END PROTOCOL PROCEDURES WITH SDLS

6.4.1 OVERVIEW

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

6.4.2 ORDER OF PROCESSING BETWEEN TC, COP-1, 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 TC, COP-1, and SDLS protocols shall occur as follows:

a) the Frame Generation Procedure (first step of processing by the function);

b) the SDLS ApplySecurity Function;

c) the Frame Operation Procedure (FOP-1), which is a sub-procedure of the Communications Operation Procedure (COP-1) and an integral part of the Virtual Channel Generation Function (final step of processing by the function).

6.4.2.2 Discussion

For completeness, figure 6-3 shows the order of processing between TC, COP-1, and SDLS functions at both the sending and receiving ends. The order of processing of the Frame Generation Function under the SDLS option differs from that of 4.3.5.1.

The sending side functional flow of the diagram proceeds as follows:

The Virtual Channel Generation Function accepts a Frame Data Unit (FDU) from the MAP Multiplexing Function, the VC Packet Processing Function, or a VCA Service User (one VCA_SDU is treated as one FDU). The Frame Generation Procedure generates a TC Transfer Frame for the FDU. The frame includes provision for the Security Header and optional Security Trailer.

The SDLS ApplySecurity Function specified in reference [7] provides the contents of these security fields as necessary and may modify the contents of the Transfer Frame Data Field by encrypting the data. The way that Transfer Frame data is passed between the Virtual Channel Generation Function and the SDLS ApplySecurity Function is implementation dependent.

The FOP-1 specified in reference [4] controls transmission and retransmission of FDUs, and it provides the value for the Frame Sequence Number in the Transfer Frame Primary Header.

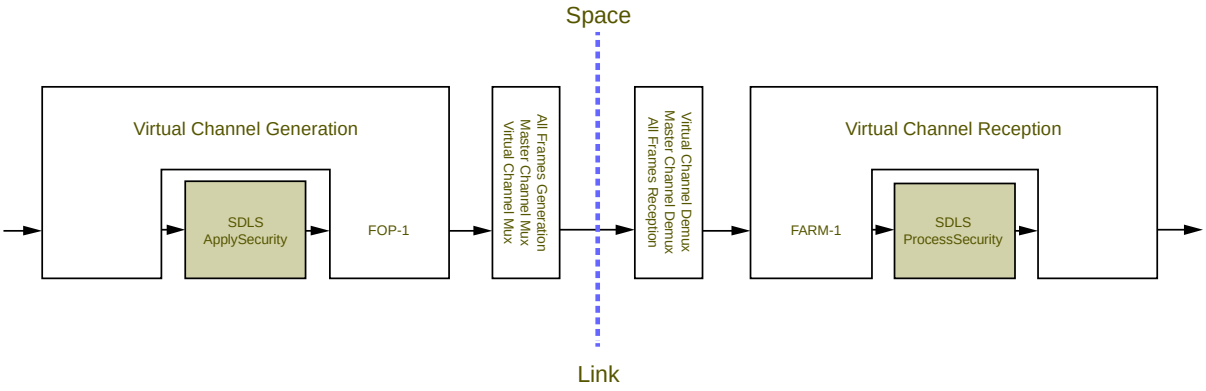

Figure 6-3: Order of Processing between TC, COP-1, and SDLS Functions

6.4.3 MAP PACKET PROCESSING FUNCTION WITH SDLS

The MAP Packet Processing Function of a TC protocol entity that supports SDLS shall

a) apply the Transfer Frame Data Field specification in 6.3.5 to determine the maximum allowable length of the User Data fields of the Segments that it generates;

b) conform to the specifications of 4.3.1.

6.4.4 MAP GENERATION FUNCTION WITH SDLS

The MAP Generation Function of a TC protocol entity that supports SDLS shall

a) apply the Transfer Frame Data Field specification in 6.3.5 to determine the maximum allowable length of the User Data fields of the Segments that it generates;

b) conform to the specifications of 4.3.2.

6.4.5 MAP MULTIPLEXING FUNCTION WITH SDLS

The MAP Multiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.3.3.

6.4.6 VC PACKET PROCESSING FUNCTION WITH SDLS

The VC Packet Processing Function of a TC protocol entity that supports SDLS shall

a) apply the Transfer Frame Data Field specification in 6.3.5 to determine the maximum allowable length of the Frame Data Units that it generates;

b) conform to the specifications of 4.3.4.

6.4.7 VIRTUAL CHANNEL GENERATION FUNCTION WITH SDLS

6.4.7.1 Discussion

There can be security configurations where, 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.

6.4.7.2 Field Lengths and Positions

When assembling a Type-D 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 4.3.5.

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 Type-D 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 TC protocol entity that supports SDLS is specified in 6.4.2.

6.4.8 VIRTUAL CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The Virtual Channel Multiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.3.6.

6.4.9 MASTER CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The Master Channel Multiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.3.7.

6.4.10 ALL FRAMES GENERATION FUNCTION WITH SDLS

The All Frames Generation Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.3.8.

6.5 RECEIVING-END PROTOCOL PROCEDURES WITH SDLS

6.5.1 OVERVIEW

When a secure TC link is required, the TC Space Data Link Protocol supports the use of the SDLS protocol. In this case, the TC Space Data Link Protocol contains differences in the receiving-end procedures compared to the procedures described in 4.4. This subsection defines those differences.

6.5.2 ORDER OF PROCESSING BETWEEN TC, COP-1, AND SDLS PROTOCOLS

6.5.2.1 Virtual Channel Reception Function

The order of processing between the functions of the TC, COP-1, and SDLS protocols shall occur as follows in the Virtual Channel Reception Function at the receiving end:

a) the Frame Acceptance and Reporting Mechanism (FARM-1), which is a sub-procedure of the Communications Operation Procedure (COP-1) and an integral part of the Virtual Channel Reception Function (first step of processing);

b) the SDLS ProcessSecurity Function;

c) the Virtual Channel Reception Function (final step of processing by the function).

6.5.2.2 Discussion

Figure 6-3 above shows the order of processing between TC, COP-1, and SDLS protocol functions at both the sending and receiving ends. The receiving side functional flow of the diagram proceeds as follows:

The FARM-1 specified in reference [4] 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-1 discards the frame.

Depending on the security features in use, the SDLS ProcessSecurity function specified in reference [7] 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 3.3.2.10, 3.4.2.9, 3.5.2.8, and 3.6.2.7.)

Before a frame reaches the Virtual Channel Reception Function it is checked by the Frame Delimiting and Fill Removal Procedure and the Frame Validation Check Procedure; both procedures are in the All Frames Reception Function. The Frame Validation Check Procedure includes the check of the Frame Error Control Field, if it is present. A frame that fails any of these checks is discarded.

6.5.3 MAP PACKET EXTRACTION FUNCTION WITH SDLS

The MAP Packet Extraction Function of a TC 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 User Data fields of the Segments that it receives;

b) conform to the specifications of 4.4.1.

6.5.4 MAP RECEPTION FUNCTION WITH SDLS

The MAP Reception Function of a TC 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 User Data fields of the Segments that it receives;

b) conform to the specifications of 4.4.2.

6.5.5 MAP DEMULTIPLEXING FUNCTION WITH SDLS

The MAP Demultiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.4.3.

6.5.6 VC PACKET EXTRACTION FUNCTION WITH SDLS

The VC Packet Extraction Function of a TC 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 4.4.4.

6.5.7 VIRTUAL CHANNEL RECEPTION FUNCTION WITH SDLS

When handling a Type-D 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 4.4.5.

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 Type-D Transfer Frames that it handles for Virtual Channels or MAPs that use SDLS.

6.5.8 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The Virtual Channel Demultiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.4.6.

6.5.9 MASTER CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The Master Channel Demultiplexing Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.4.7.

6.5.10 ALL FRAMES RECEPTION FUNCTION WITH SDLS

The All Frames Reception Function of a TC protocol entity that supports SDLS shall conform to the specifications of 4.4.8.

6.6 ADDITIONAL MANAGED PARAMETERS FOR SDLS

6.6.1 The managed parameters associated with a Virtual Channel without Segment Headers when the TC Space Data Link Protocol supports the SDLS protocol shall conform to the definitions in table 5-3 and the additional definitions in table 6-1.

Table 6-1: Additional Managed Parameters for a Virtual Channel without Segment Headers 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 Virtual Channel. 2 The valid lengths for the Security Header and Security Trailer are specified in reference [7].	

6.6.2 The managed parameters associated with a MAP when the TC Space Data Link Protocol supports the SDLS protocol shall conform to the definitions in table 5-4 and the additional definitions in table 6-2.

Table 6-2: 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 [7].	

ANNEX A

ACRONYMS

(INFORMATIVE)

APID	Application Process Identifier
ARQ	Automatic Repeat Request
CCSDS	Consultative Committee for Space Data Systems
CLCW	Communications Link Control Word
COP	Communications Operation Procedure
FARM	Frame Acceptance and Reporting Mechanism
FDU	Frame Data Unit
FOP	Frame Operation Procedure
GMAP ID	Global Multiplexer Access Point Identifier
GVCID	Global Virtual Channel Identifier
MAP ID	Multiplexer Access Point Identifier
MAP	Multiplexer Access Point
MAPA	Multiplexer Access Point Access
MAPP	Multiplexer Access Point Packet
MC	Master Channel
MCF	Master Channel Frame
MCID	Master Channel Identifier
MSB	Most Significant Bit
OSI	Open Systems Interconnection
PVN	Packet Version Number
QoS	Quality of Service

SAP	Service Access Point
SCID	Spacecraft Identifier
SDLS	Space Data Link Security
SDU	Service Data Unit
TC	Telecommand
TFVN	Transfer Frame Version Number
VC	Virtual Channel
VCA	Virtual Channel Access
VCF	Virtual Channel Frame
VCID	Virtual Channel Identifier
VCP	Virtual Channel Packet

ANNEX B

INFORMATIVE REFERENCES

(INFORMATIVE)

[B1] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 4. CCSDS Record (Yellow Book), CCSDS A02.1-Y-4. Washington, D.C.: CCSDS, April 2014.

[B2] *Overview of Space Communications Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-3. Washington, D.C.: CCSDS, July 2014.

[B3] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.

[B4] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

[B5] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 910.4-B-2. Washington, D.C.: CCSDS, October 2005.

[B6] *TM Space Data Link Protocol*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 132.0-B-2. Washington, D.C.: CCSDS, September 2015.

[B7] *AOS Space Data Link Protocol*. Issue 3. Recommendation for Space Data System Standards (Blue Book), CCSDS 732.0-B-3. Washington, D.C.: CCSDS, September 2015.

NOTE – Normative references are listed in 1.7.