The Consultative Committee for Space Data Systems

Draft Recommendation for
Space Data System Standards

UNIFIED SPACE DATA LINK PROTOCOL

DRAFT RECOMMENDED STANDARD

CCSDS 732.1-P-2.1

PINK SHEETS

August 2023

The Consultative Committee for Space Data Systems

Draft Recommendation for Space Data System Standards

UNIFIED SPACE DATA LINK PROTOCOL

DRAFT RECOMMENDED STANDARD

CCSDS 732.1-P-2.1

PINK SHEETS

August 2023

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 732.1-B-1	Unified Space Data Link Protocol, Recommended Standard, Issue 1	October 2018	Original issue, superseded
CCSDS 732.1-B-2	Unified Space Data Link Protocol, Recommended Standard, Issue 2	October 2021	Current issue
CCSDS 732.1-P-2.1	Unified Space Data Link Protocol, Draft Recommended Standard, Issue 2.1	August 2023	Current draft: – adds VC Packet and VC Access services; – changes maximum number of frames per CLTU from one to greater than one to match Telecommand specification; – adds security section table, diagram, and text concerning the order of COP to SDLS, similar to Telecommand specification.

1.7 REFERENCES

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

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

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

[3] *TM Synchronization and Channel Coding*. Issue ~~3~~4. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.0-B-~~3~~4. Washington, D.C.: CCSDS, ~~September 2017~~April 2022.

[4] *Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications*. Issue ~~1~~2. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.2-B- ~~1~~2. Washington, D.C.: CCSDS, ~~March 2012~~February 2023.

[5] *CCSDS Space Link Protocols over ETSI DVB-S2 Standard*. Issue ~~1~~2. Recommendation for Space Data System Standards (Blue Book), CCSDS 131.3-B-~~1~~2. Washington, D.C.: CCSDS, ~~March 2013~~April 2022.

[6] *TC Synchronization and Channel Coding*. Issue ~~3~~4. Recommendation for Space Data System Standards (Blue Book), CCSDS 231.0-B-~~3~~4. Washington, D.C.: CCSDS, ~~September 2017~~July 2021.

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

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

[9] *Communications Operation Procedure-1*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.1-B-2. Washington, D.C.: CCSDS, September 2010.

[10] *Proximity-1 Space Link Protocol—Data Link Layer*. Issue 6. Recommendation for Space Data System Standards (Blue Book), CCSDS 211.0-B-6. Washington, D.C.: CCSDS, July 2020.

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

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

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

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

[15] *Space Data Link Security Protocol*. Issue ~~1~~2. Recommendation for Space Data System Standards (Blue Book), CCSDS 355.0-B-~~1~~2. Washington, D.C.: CCSDS, ~~September 2015~~July 2022.

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

NOTE – Informative references are listed in annex F.

2.1.2 PROTOCOL FEATURES

2.1.2.1 Transfer Frames, Virtual Channels, and Multiplexer Access Points

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

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

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

2.1.2.2 Additional USLP Features

USLP has a larger maximum Transfer Frame size than previous CCSDS space data link protocols in order to reduce the operational frame handling process for high-rate missions. The protocol also has increased the capability for identifying (using a larger addressing space) more spacecraft than previous CCSDS space data link protocols. In order to accommodate a much larger frame-size range than previous space data link protocols, USLP provides a configurable-sized sequence counter in the Frame Primary Header.

2.1.2.3 Efficient Data Transfer

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

2.2.2 COMMON FEATURES OF SERVICES

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

Each service user is identified by a SAP address that is associated with a specific ID for that specific service e.g., for MAP services a GMAP ID is required to identify that specific SAP.

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

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

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

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

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

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

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

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

NOTE – This Recommended Standard assumes that these services are provided at the end points of a space link. However, this Recommended Standard makes no assumptions concerning how these end points are composed or configured either on board a spacecraft or in a ground system. In a ground system, the services defined by this Recommended Standard can be extended or enhanced with Space Link Extension Services (reference [F5]).

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

a) no SDU is lost;

b) no SDU is duplicated; and

c) no SDU is delivered out of sequence.

2.2.3.3 Expedited Service—Best Effort Delivery

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

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

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

2.2.3.4 Security Service

The optional use of the SDLS protocol (references [15] and [16]) provides all its functions (authentication, encryption, authenticated encryption) for the data in the Transfer Frame Data Zone (TFDZ) of a USLP Frame. It provides full protection for the service data of the MAP Packet (MAPP) service, the MAP Octet Stream Service, ~~and~~ the MAP Access (MAPA) Service, Virtual Channel Packet (VCP) Service, and Virtual Channel Access (VCA) Service (see section 6, Protocol Specification with SDLS option).

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

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

SDLS provides no protection for the Master Channel Operational Control Field (USLP_MC_OCF) Service or for the Insert Service. It also provides no protection for the frames supplied to USLP by external sources such as by the VCF and the MCF services.

2.2.3.5 Asynchronous Service

In asynchronous service, there are no timing relationships between the transfer of SDUs supplied by the service user and the transmission of Transfer Frames generated by the service

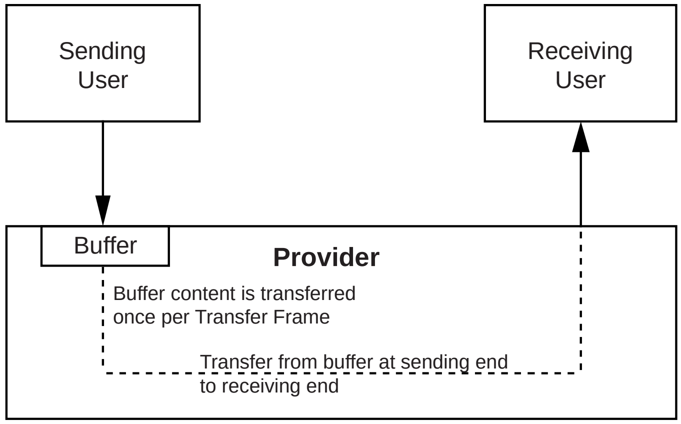


Figure 2-5: Synchronous Service Model

2.2.3.7 Periodic Service

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

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

2.2.4 SUMMARY OF SERVICES

2.2.4.1 Introduction

The following data transfer services are provided by USLP:

– MAPP, MAPA, and MAP Octet Stream are provided for a MAP Channel;

– VCP, VCA, and VCF is provided for a VC;

– USLP_MC_OCF and MCF are provided for an MC;

– Insert is provided for all Transfer Frames on a Physical Channel.

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

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

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

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

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

2.2.4.2 MAP Packet Service

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

The service is unidirectional and asynchronous. If a COP is used, then both Sequence-Controlled and Expedited services are provided for the MAPP Service. When the related managed parameter states that there is no COP in Effect, only Best Effort Delivery is provided for a MAP Channel.

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

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

2.2.4.3 Virtual Channel Packet Service

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

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

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

2.2.4.4 MAP Access Service

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

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

2.2.4.5 Virtual Channel Access Service

The VCA Service provides transfer of a sequence of privately formatted service data units of variable length across a space link. The length of the service data units transferred by this service cannot 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.

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

2.2.4.6 MAP Octet Stream Service

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

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

2.2.4.7 USLP Master Channel Operational Control Field Service

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

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

2.2.4.8 Virtual Channel Frame Service

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

2.2.4.11 COPs Management Service

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

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

2.2.5 RESTRICTIONS ON SERVICES

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

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

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

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

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

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

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

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

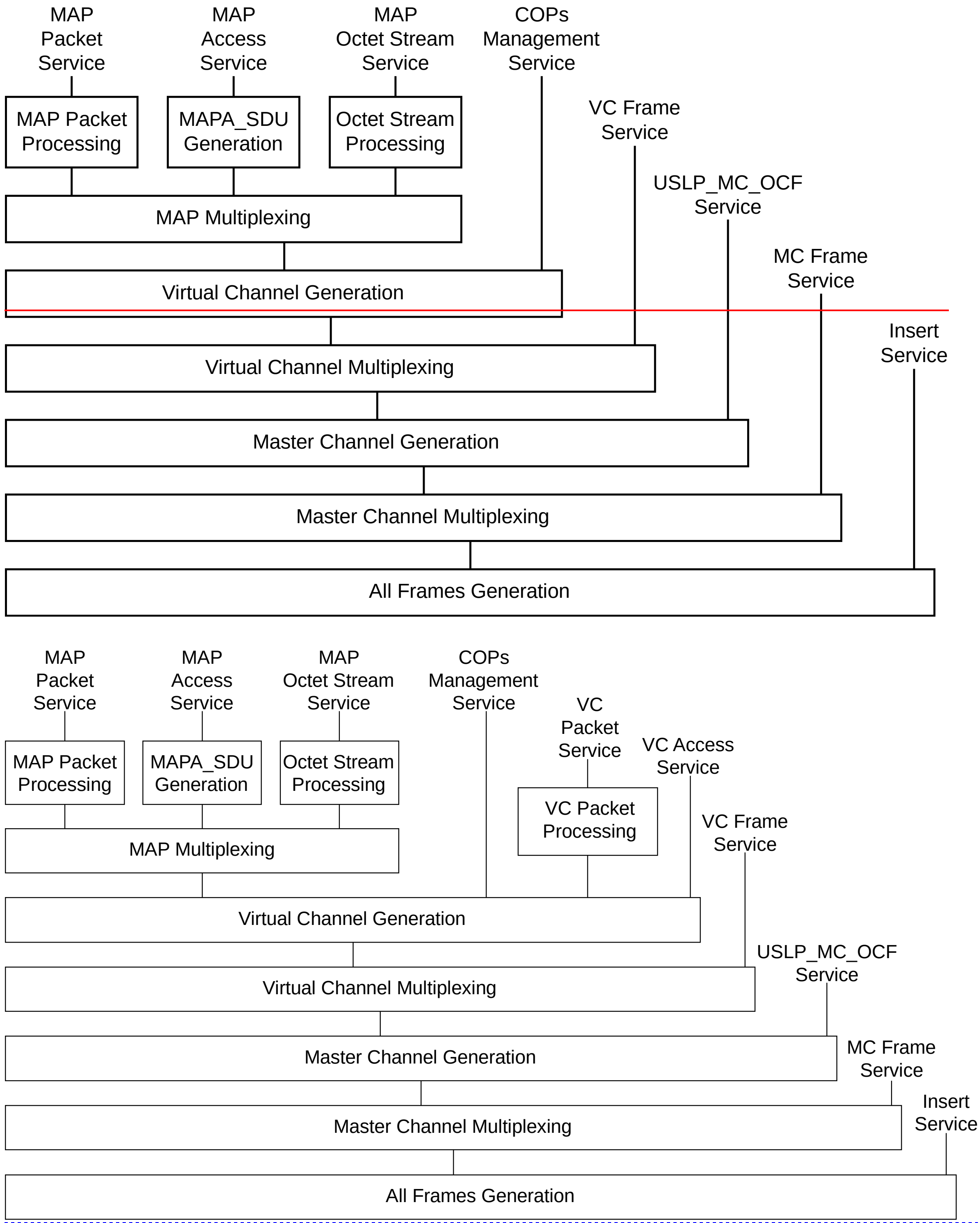


Figure 2-6: Internal Organization of Protocol Entity (Sending End)

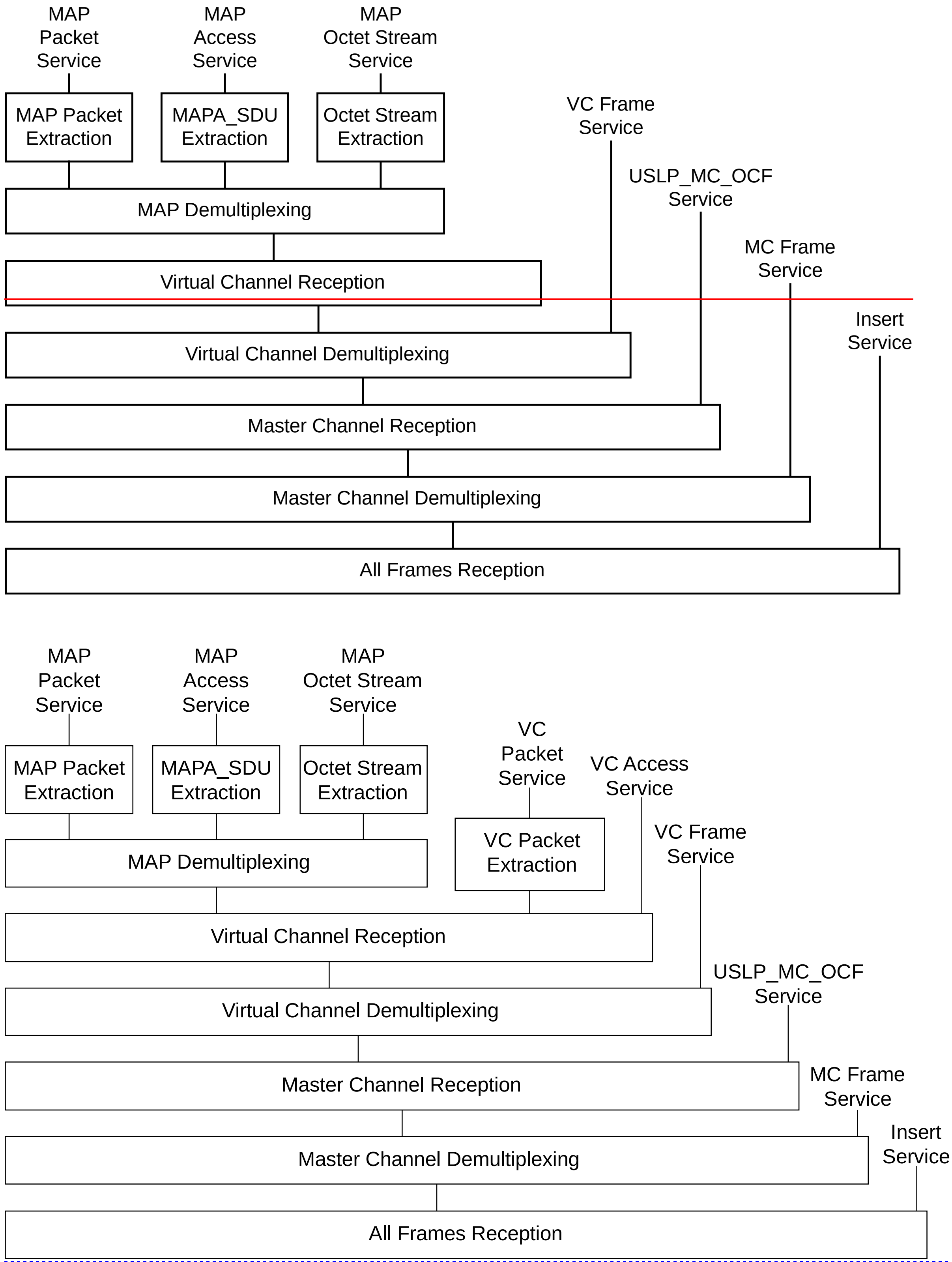
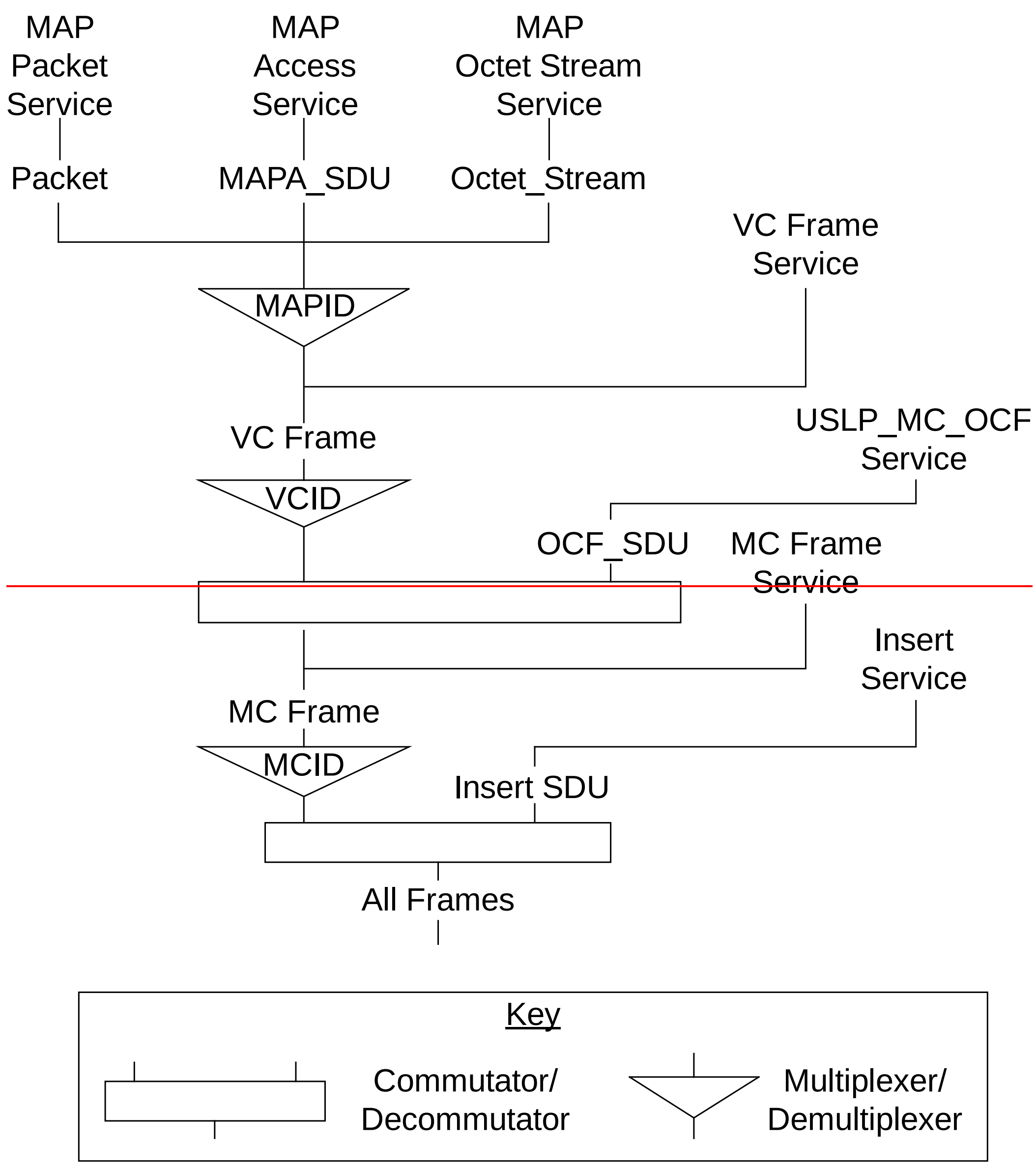


Figure 2-7: Internal Organization of Protocol Entity (Receiving End)

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

In figure 2-8, multiplexing (shown with a triangle) is a function of mixing, according to an algorithm established by the project, multiple streams of data units, each with a different identifier, to generate a single stream of data units. Commutation (shown with a box) is a function of concatenating (according to the formatting rule specified by the protocol definition) multiple data units, each from a different service, in a single PDU sharing the same identifier.



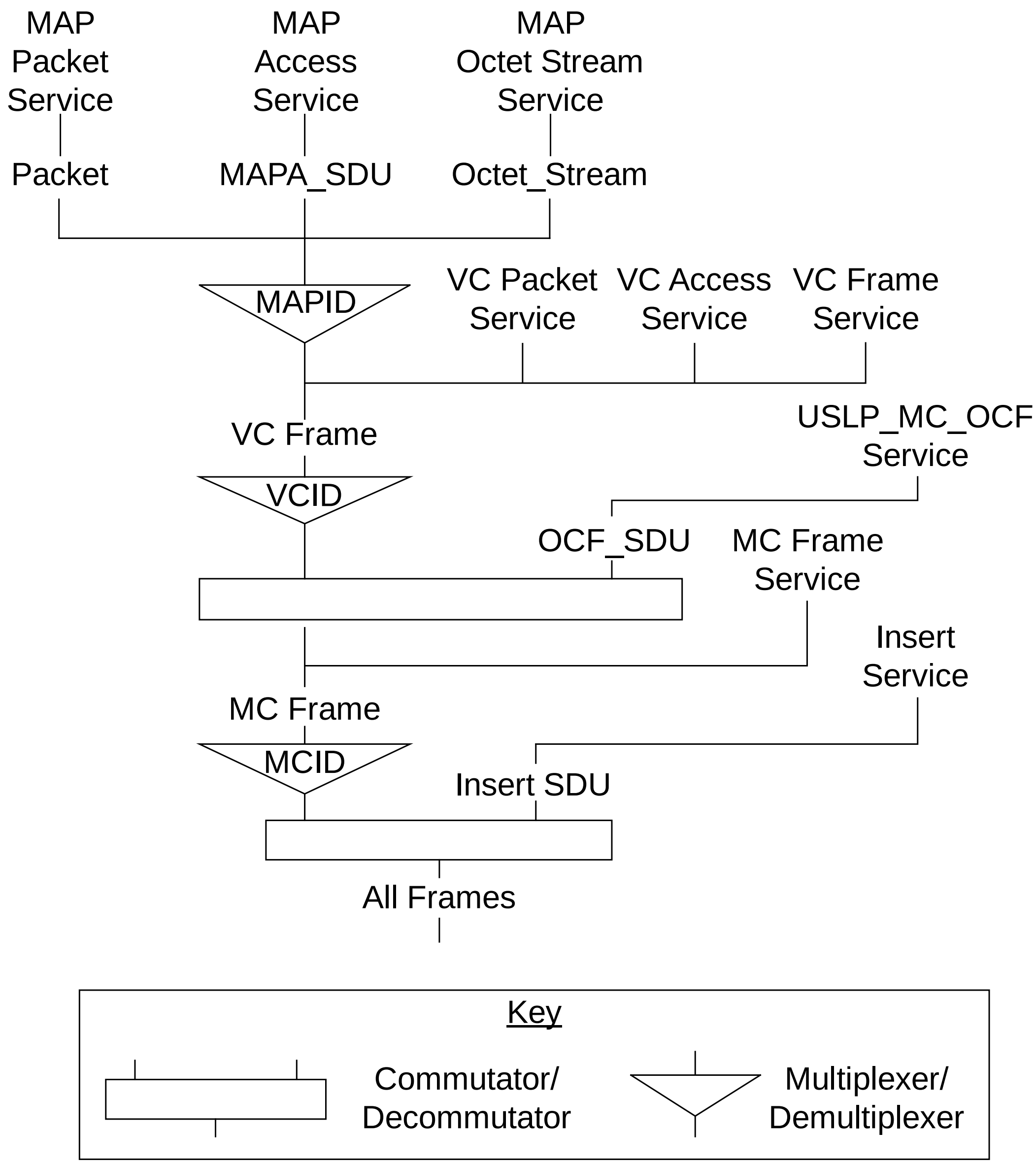


Figure 2-8: Unified Space Data Link Protocol Channel Tree

2.4 SERVICES ASSUMED FROM LOWER LAYERS

2.4.1 SERVICES ASSUMED FROM THE SYNCHRONIZATION AND CHANNEL CODING SUBLAYER

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

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

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

c) delimiting and synchronizing functions;

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

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

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

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

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

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

3 SERVICE DEFINITION

3.1 OVERVIEW

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

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

3.2 SOURCE DATA

3.2.1 SOURCE DATA OVERVIEW

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

The SDUs transferred by USLP shall be:

a) Packet (MAP Packet and VC Packet);

b) MAPA_SDU;

c) VCA_SDU;

d) Octet Stream Data;

e) OCF_SDU;

f) USLP Transfer Frame; and

g) Insert SDU (IN_SDU).

3.2.2 ~~MAP~~ PACKET ~~SERVICE DATA UNIT~~

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

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

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

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

NOTES

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

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

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

3.2.3 MAP ACCESS SERVICE DATA UNIT

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

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

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

3.2.4 VIRTUAL CHANNEL ACCESS SERVICE DATA UNIT

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

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

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

3.2.5 MAP OCTET STREAM DATA

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

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

3.4 VIRTUAL CHANNEL PACKET SERVICE

3.4.1 OVERVIEW

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

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

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

3.4.2 VCP SERVICE PARAMETERS

3.4.2.1 General

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

3.4.2.2 Packet

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

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

3.4.2.3 GVCID

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

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

3.4.2.4 Packet Version Number

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

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

3.4.2.5 SDU ID

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

3.4.2.6 Service Type

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

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

3.4.2.7 Notification Type

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

3.4.2.8 Packet Quality Indicator

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

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

3.4.2.9 Verification Status Code

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

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

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

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

3.4.3 VCP SERVICE PRIMITIVES

3.4.3.1 General

The service primitives associated with the VCP service are:

a) VCP.request;

b) VCP_Notify.indication;

c) VCP.indication.

3.4.3.2 VCP.request

3.4.3.2.1 Function

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

3.4.3.2.2 Semantics

The VCP.request primitive shall provide parameters as follows:

```
VCP_Notify.indication        (Packet,
                              GVCID,
                              Packet Version Number,
                              SDU_ID,
                              Service Type)
```

3.4.3.2.3 When Generated

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

3.4.3.2.4 Effect On Receipt

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

3.4.3.3 VCP_Notify.indication

3.4.3.3.1 Function

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

3.4.3.3.2 Semantics

The VCP_Notify.indication primitive shall provide parameters as follows:

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

3.4.3.3.3 When Generated

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

3.4.3.3.4 Effect On Receipt

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

3.4.3.4 VCP.indication

3.4.3.4.1 Function

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

3.4.3.4.2 Semantics

The VCP.indication primitive shall provide parameters as follows:

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

3.4.3.4.3 When Generated

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

3.4.3.4.4 Effect On Receipt

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

3.6 VIRTUAL CHANNEL ACCESS SERVICE

3.6.1 OVERVIEW

The VCA Service provides transfer of a sequence of privately formatted service data units of variable length across a space link. The length of the service data units transferred by this service may 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.

For fixed-length Transfer Frames only on one Virtual Channel, the VC Access Service cannot exist simultaneously with the VCP Service. 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 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 [9].

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 [15].

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

3.6.3 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)
```

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)
```

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.

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

Figure 4-4: Transfer Frame Data Field Header

4.1.4.2.2 TFDZ Construction Rules

4.1.4.2.2.1 General

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

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

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

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

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

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

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

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

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

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

4.1.4.2.2.2 Itemized Construction Rules

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

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

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

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

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

NOTE – An example is video.

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

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

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

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

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

a) one MAPA_SDU;

b) one VCA_SDU;

c) one or more complete Packets;

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

e) one or more Proximity-1 SPDUs.

NOTES

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

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

Table 4-3: Summary of the TFDZ Construction Rules

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

4.1.4.2.3 USLP Protocol Identifier

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

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

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

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

4.1.4.3 Transfer Frame Data Zone

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

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

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

NOTES

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

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

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

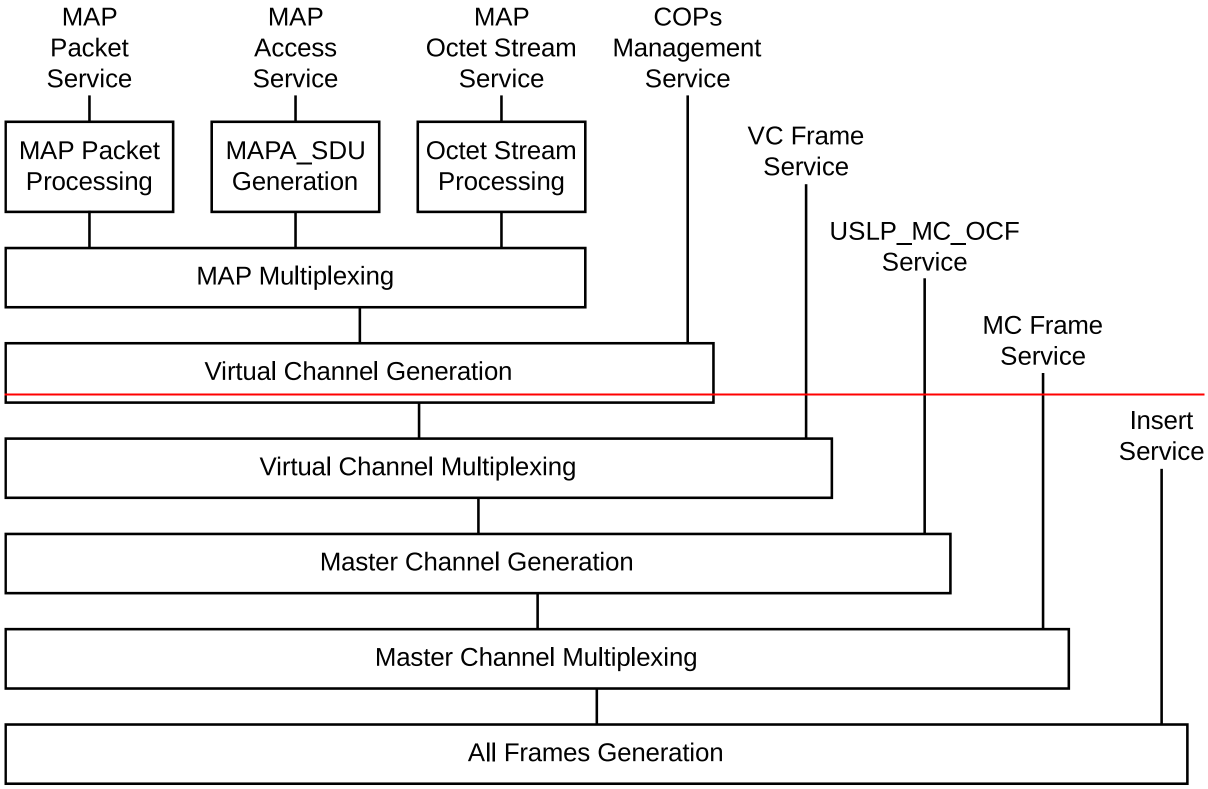
4.1.4.3.4 If a fixed-length TFDZ is partially completed with Packets when the required release time for a Transfer Frame of a VC has been reached, an Encapsulation Idle Packet (reference [13]) shall complete the remainder of the TFDZ.

NOTES

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

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

sending end, and shows logical relationships among these functions. The figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.



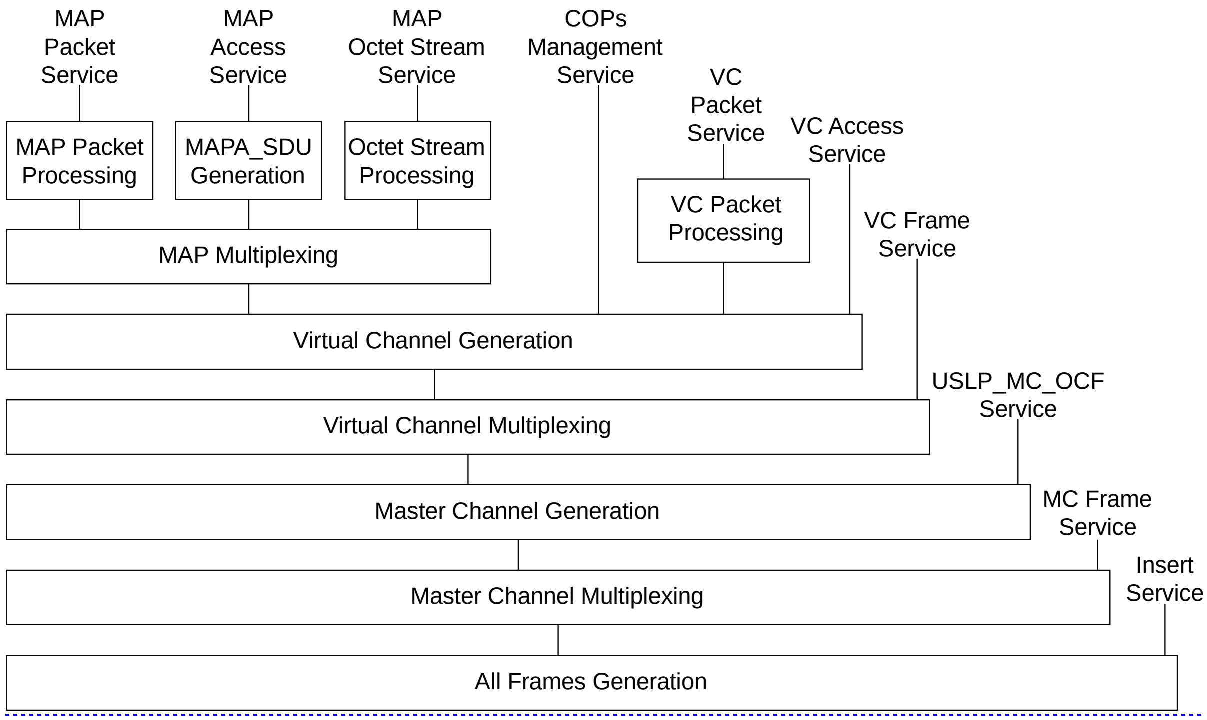


Figure 4-5: Internal Organization of Protocol Entity (Sending End)

4.2.2 MAP PACKET PROCESSING FUNCTION

4.2.2.1 MAP Packet Processing for Fixed-Length TFDZ

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

NOTES

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

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

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

4.2.6 VC PACKET PROCESSING FUNCTION

4.2.6.1 VC Packet Processing for Fixed-Length TFDZ

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

NOTES

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

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

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

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

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

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

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

NOTES

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

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

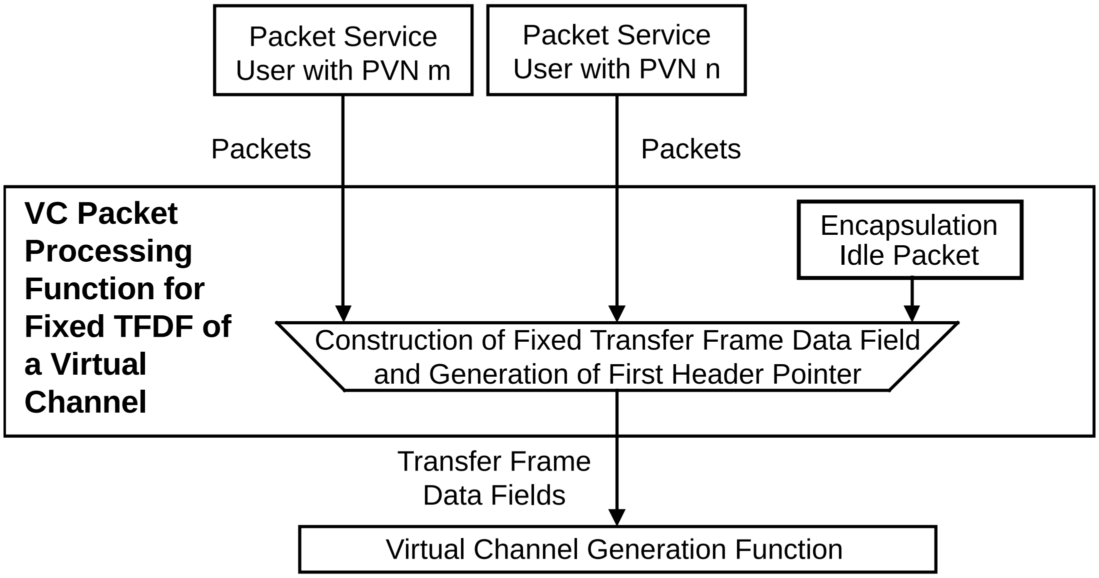


Figure 4-12: Abstract Model of Packet Processing Function for Fixed-Length TFDFs

4.2.6.2 VC Packet Processing for Variable-Length TFDZ

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NOTES

1 The associated TFDZ Construction Rule used is ‘111’.

2 An abstract model of the VC Packet Processing Function for variable TFDFs is illustrated in figure 4-13.

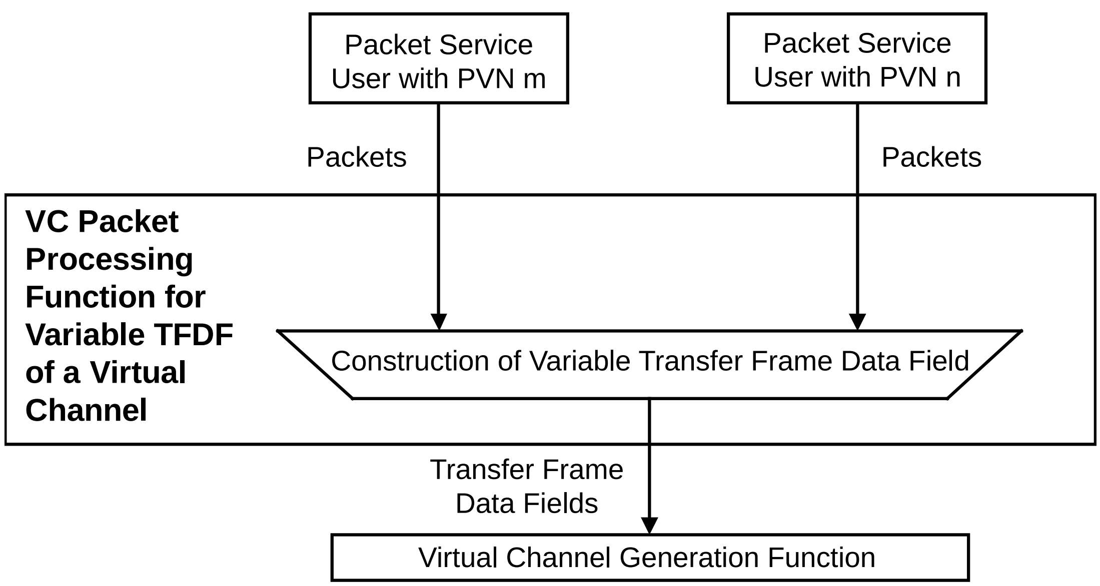


Figure 4-13: Abstract Model of Virtual Packet Processing Function for Variable-Length TFDFs

4.2.7 VIRTUAL CHANNEL GENERATION FUNCTION

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

4.2.7.1 General

The ~~VC~~Virtual Channel Generation Function shall perform the following ~~two~~three procedures in the following order~~ listed below~~:

~~a) the FOP, which is a sub-procedure of the COP; and~~

~~b) the Frame Generation Procedure.~~

a) the Frame Initialization Procedure;

b) the Frame Operation Procedure (FOP-1/FOP-P), which is a sub-procedure of the Communications Operation Procedure (COP-1/COP-P); and

c) the Frame Finalization Procedure.

~~NOTE – This function, with the procedures mentioned above, maintains, increments and includes the VC Frame Count together with the VC Frame Count Length field in the Transfer Frame Primary Header based upon the contents of the Bypass/Sequence Control Flag field.~~

4.2.7.2 Frame Initialization Procedure

The Frame Initialization Procedure shall accept Transfer Frame Data Fields (TFDFs) from the MAP Multiplexing Function, the VC Packet Processing Function, or a VCA Service User (one VCA_SDU is treated as one TFDF) and generate a partially complete USLP transfer frame that includes a Transfer Frame Primary Header and a Transfer Frame Data Field.

NOTE – Only a few (static) fields of the Transfer Frame Primary Header are filled in by this procedure.

4.2.7.3 FOP Procedure

4.2.7.3.1 The FOP shall accept TFDFs from the MAP Multiplexing Function and shall control transmission and retransmission of USLP Frames by examining the report contained in the CLCW/PLCWs and generating COP Control Commands.

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

~~NOTE This Recommended Standard does not specify the interfaces and methods by which CLCWs are delivered to the FOP.~~

NOTES

1 This Recommended Standard does not specify the interfaces and methods by which CLCWs are delivered to the FOP. The detailed specifications of the FOP-1/FOP-P are given in references [9] and [10].

2 Most of the Control Commands only cause internal processing in the FOP while some of them cause the generation of Transfer Frames, carrying Control Commands for configuring COPs (e.g., 'Set V(R)').

~~**4.2.7.4** The Frame Generation Procedure shall generate Transfer Frames by attaching a Transfer Frame Primary Header to each TFDF delivered to the FOP.~~

4.2.7.4 The Frame Finalization Procedure

The Frame Finalization Procedure shall fill in the values of the remainder of the transfer frame fields by completing the Transfer Frame Primary Header of each transfer frame or Control Command by adding the value delivered by the FOP (i.e., Frame Sequence Number).

NOTES

1 Values supplied by the FOP are inserted in fields in the Transfer Frame Primary Header.

2 The FOP supplies the Bypass/Sequence Control Flag and the Protocol Control Command Flag.

3 If the Bypass/Sequence Control Flag is set to '0', the Frame Sequence Number supplied by the FOP is placed in the VCF Count field.

4 The actual transmission of a Transfer Frame can be delayed by the next function, that is, the VC Multiplexing Function by the multiplexing algorithm defined by management (see 4.2.8.3). The maximum delay is specified by the managed parameter, 'Maximum delay in milliseconds between releases of USLP Frames of the same VC' in table 5-3.

5 An abstract model of the VC Generation Function is illustrated in figure 4-14.

4.2.11.5 Externally generated Transfer Frames associated with the VCF and MCF Services shall always bypass the error control encoding functions specified above.

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

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

4.2.11.7 Each data unit delivered by the All Frames Generation Function shall consist of one ~~Transfer Frame~~or more Transfer Frames as specified by the 'Maximum Number of Frames Given to the C&S Sublayer as a Single Data Unit' Managed Parameter.

~~NOTE – USLP does not replicate the TC Space Data Link Protocol (reference [F6]) feature that allows the All Frames Generation Function to deliver more than one Transfer Frame at a time.~~

4.2.11.8 When reference [6] is used as the Synchronization and Channel Coding Sublayer, the All Frames Generation Function may request the Synchronization and Channel Coding Sublayer to perform repeated transmissions of a data unit, that is, 'systematic retransmissions' as described in 2.4.2, unless the data unit contains a USLP Frame carrying service data on the Expedited Service (i.e., Type-BD frame in COP-1).

NOTES

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

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

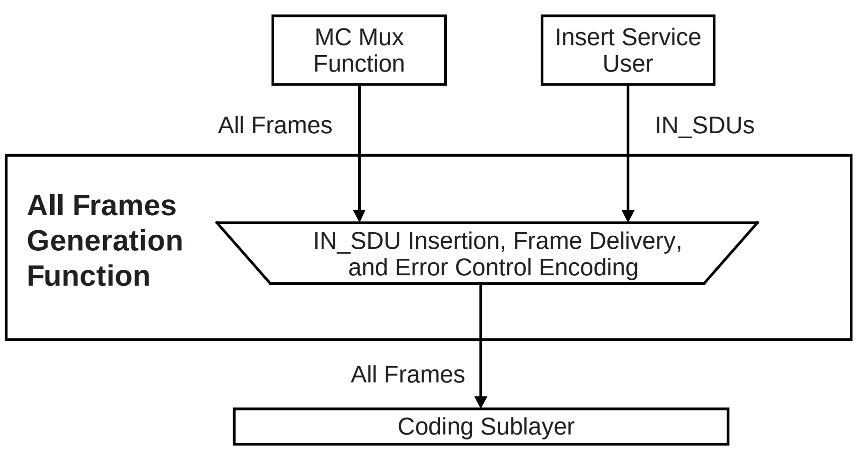
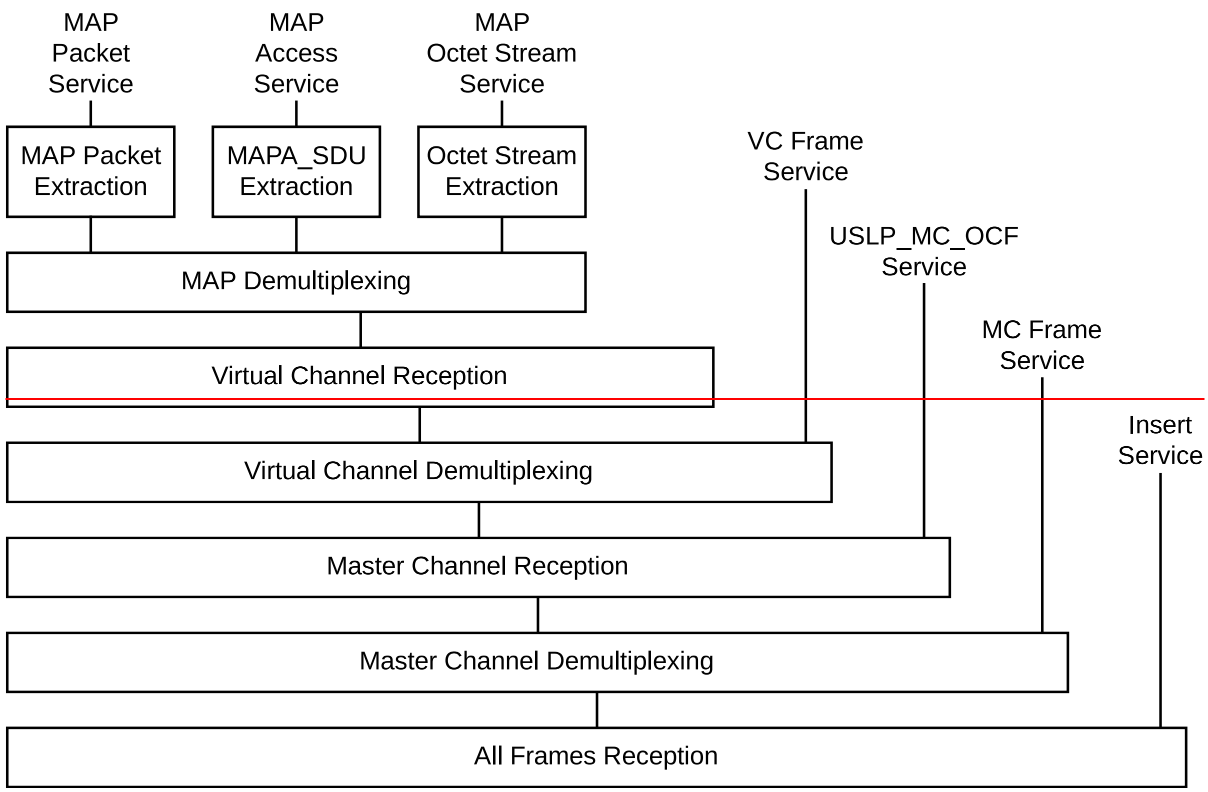


Figure 4-18: Abstract Model of All Frames Generation Function

4.3 PROTOCOL PROCEDURES AT THE RECEIVING END

4.3.1 OVERVIEW

This subsection describes procedures at the receiving end associated with each of the functions shown in figure 4-19 (which is identical to figure 2-7). In the figure, data flows from bottom to top. The figure identifies data-handling functions performed by the protocol entity at the receiving end and shows logical relationships among these functions. The figure is not intended to imply any hardware or software configuration in a real system. Depending on the services actually used for a real system, not all the functions may be present in the protocol entity. The procedures described in this subsection are defined in an abstract sense and are not intended to imply any particular implementation approach of a protocol entity.



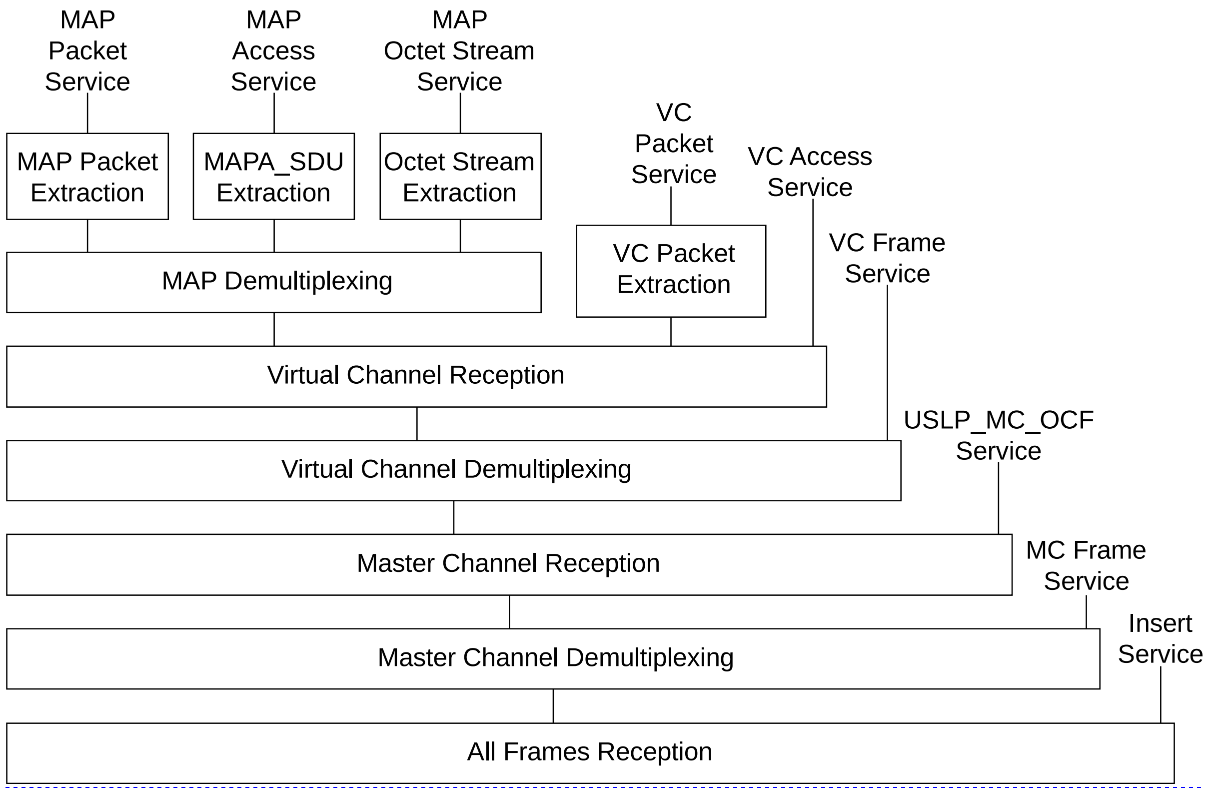


Figure 4-19: Internal Organization of Protocol Entity (Receiving End)

4.3.2 MAP PACKET EXTRACTION FUNCTION

4.3.2.1 Discussion

The extraction of Packets from fixed-length TFDFs is different than from variable-length TFDFs. The TFDZ Construction Rules identify the method used to load the Packets in the TFDF, as well as identify the set of procedures used to extract the packets.

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

4.3.2.2 MAP Packet Extraction Function for a Fixed-Length TFDZ

NOTE – The MAPP Extraction Function used to extract variable-length Packets from the fixed-length TFDZs is associated with TFDZ Construction Rule ‘000’ when a packet spans multiple USLP Frames.

4.3.2.2.1 The MAPP Extraction Function shall extract Packets from TFDZs received from the MAP Demultiplexing Function.

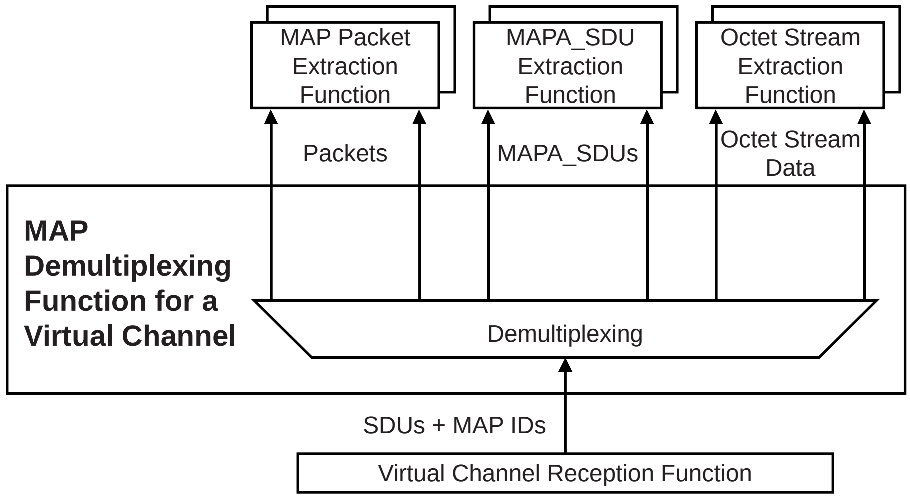


Figure 4-25: Abstract Model of MAP Demultiplexing Function

4.3.6 VC PACKET EXTRACTION FUNCTION

4.3.6.1 Discussion

The extraction of Packets from fixed-length TFDFs is different than from variable-length TFDFs. The TFDZ Construction Rules identify the method used to load the Packets in the TFDF, as well as identify the set of procedures used to extract the packets.

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

4.3.6.2 VC Packet Extraction Function for a Fixed-Length TFDZ

NOTE – The VC Packet Extraction Function used to extract variable-length Packets from the fixed-length TFDZs is associated with TFDZ Construction Rule '000' when a packet spans multiple USLP Frames.

4.3.6.2.1 The VC Packet Extraction Function shall extract Packets from TFDZs received from the Virtual Channel Reception Function.

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

4.3.6.2.3 If the last Packet removed from the TFDZ is incomplete, the VC Packet Extraction Function shall retrieve its remainder from the beginning of the next sequential TFDF received on the same GVCID.

4.3.6.2.4 The FHP for the next TFDF shall be used to determine the length of the remainder and, hence, the beginning of the next Packet to be extracted.

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

NOTES

3 Incomplete Packets are not required to be delivered in cross-support situations. Idle Packets are discarded. TFDFs that contain only idle data are also discarded.

4 An abstract model of the VC Packet Extraction Function for fixed-length TFDFs is illustrated in figure 4-26.

4.3.6.2.6 MAPPs associated with an invalid MAP ID shall be discarded.

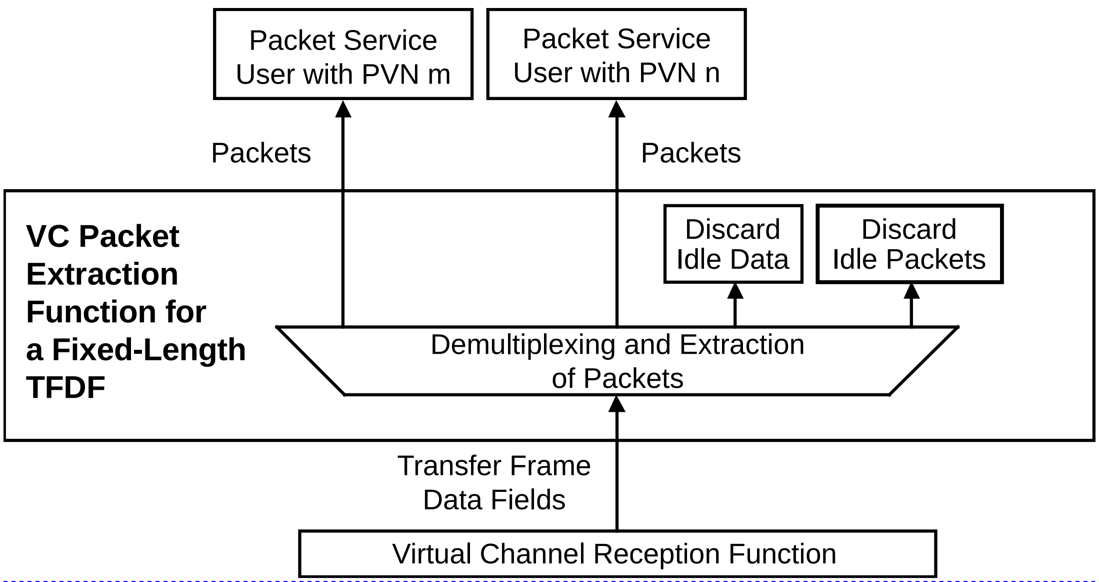


Figure 4-26: Abstract Model of VC Packet Extraction Function for Fixed-Length TFDFs

4.3.6.3 VC Packet Extraction Function for a Variable-Length TFDZ

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

4.3.6.3.1 The VC Packet Extraction Function shall extract Packets from variable-length TFDFs received from the Virtual Channel Reception Function.

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

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

NOTE – Incomplete Packets are not required to be delivered in cross-support situations.

4.3.6.3.4 VC Packets associated with an invalid VCID shall be discarded.

NOTE – An abstract model of the VC Packet Extraction Function for a variable-length TFDZ is illustrated in figure 4-27.

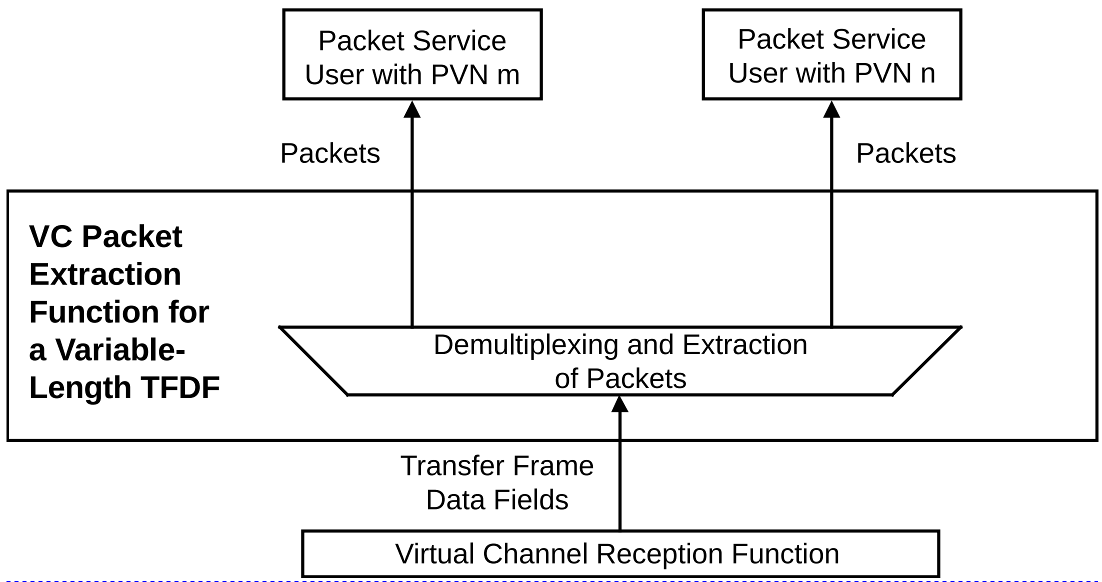


Figure 4-27: Abstract Model of VC Packet Extraction Function for Variable-Length TFDFs

4.3.7 VIRTUAL CHANNEL RECEPTION FUNCTION

4.3.7.1 The VC Reception Function shall:

a) be used to perform the Frame Acceptance and Reporting Mechanism (FARM/FARM-P) when applicable; and

b) decommutate fields of Transfer Frames of a VC.

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

4.3.7.2 The Frame Acceptance and Reporting Mechanism procedure, when applicable, shall perform the operations defined in reference [9] or [10] and, in case of successful checks, shall deliver the VC Frames to the Decommutation Procedure.

NOTES

1 The operation performed by the Frame Acceptance and Reporting Mechanism procedure (i.e., either FARM or FARM-P) include, for example, Accepting/Discarding Frames, generating CLCW/PLCW, etc., as per references [9] and [10].

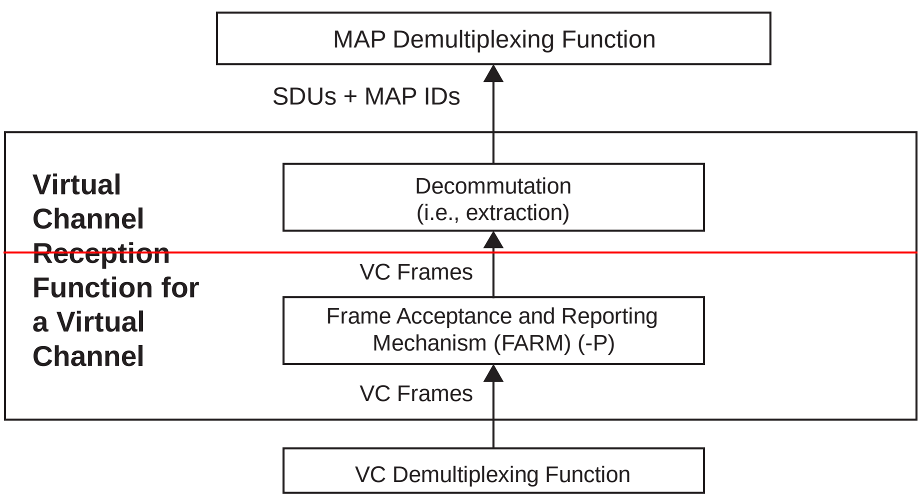
2 When no FARM is applicable to the given VC, VC Frames are passed directly to the Decommutation Procedure without performing any check.

~~**4.3.7.3** The Decommutation Procedure shall extract SDUs contained in the TFDF along with the associated MAP ID from the Transfer Frame Primary Header and deliver them to the MAP Demultiplexing Function.~~

4.3.7.3 The Decommutation Procedure shall extract the contents of the Transfer Frame Data Zone (Packets, or MAPA_SDU/VCA_SDU, or Octet Stream Data) along with the associated MAP ID or VCID from the Transfer Frame Primary Header and deliver them to the appropriate receiving function as shown in figure 4-28.

4.3.7.4 If a gap in the VCF Count is detected, a Loss Flag may (optionally) be delivered to the MAP Demultiplexing Function, VC Packet Extraction Function, or VCA Service User associated with this VC.

NOTE – An abstract model of the VC Reception Function is illustrated in figure 4-28.



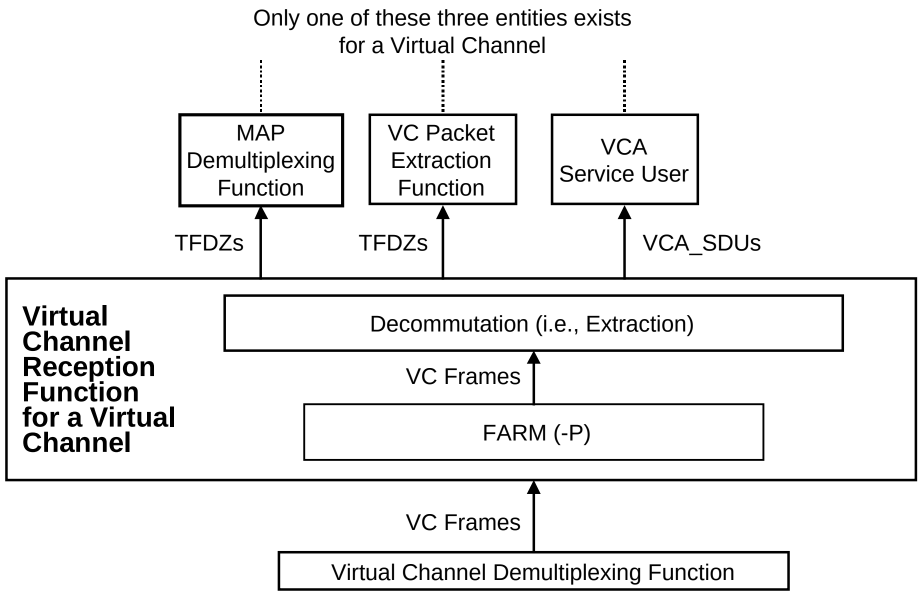


Figure 4-28: Abstract Model of Virtual Channel Reception Function

4.3.8 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION

4.3.8.1 The VC Demultiplexing Function shall be used to demultiplex Transfer Frames of different VCs of an MC.

NOTE – There is an instance of the VC Demultiplexing Function for each MC that has multiple VCs.

4.3.8.2 The VC Demultiplexing Function shall examine the VCID in the incoming stream of Transfer Frames and route them to the instances of the VC Reception Function and, if present, to the VCF Service users.

4.3.8.3 If a gap in the VCF Count is detected, a Loss Flag may (optionally) be delivered to the users.

NOTES

1 Any OID Transfer Frames encountered are discarded, once the Insert Zone (if present) is extracted.

2 Transfer Frames with an invalid VCID are also discarded.

3 An abstract model of the VC Demultiplexing Function is illustrated in figure 4-29.

4.3.11.1.4 If the optional Insert Service is used, the All Frames Reception Function shall extract the IN_SDU from the Insert Zone of the incoming stream of Transfer Frames, whose managed parameter 'Physical Channel Transfer Frame Type' is 'Fixed Length', regardless of their GMAP ID, and deliver them to the Insert Service user.

4.3.11.1.5 If error protection of the IN_SDU is not required, extraction of the IN_SDU may be performed prior to decoding of FECF described above.

4.3.11.1.6 If the Synchronization and Channel Coding Sublayer used is *TC Synchronization and Channel Coding* (reference [6]), then the All Frames Reception Function shall be used to reconstitute ~~a single~~one or more Transfer Frames from the error free data stream provided by the TC Channel Coding Sublayer. ~~and perform~~ Thereafter, the following two procedures shall be performed to determine whether the reconstituted Transfer Frames ~~is~~are valid or not in the following order:

a) Frame Delimiting and Fill Data Removal Procedure (4.3.11.2); and

b) USLP Frame Validation Check Procedure (4.3.11.3).

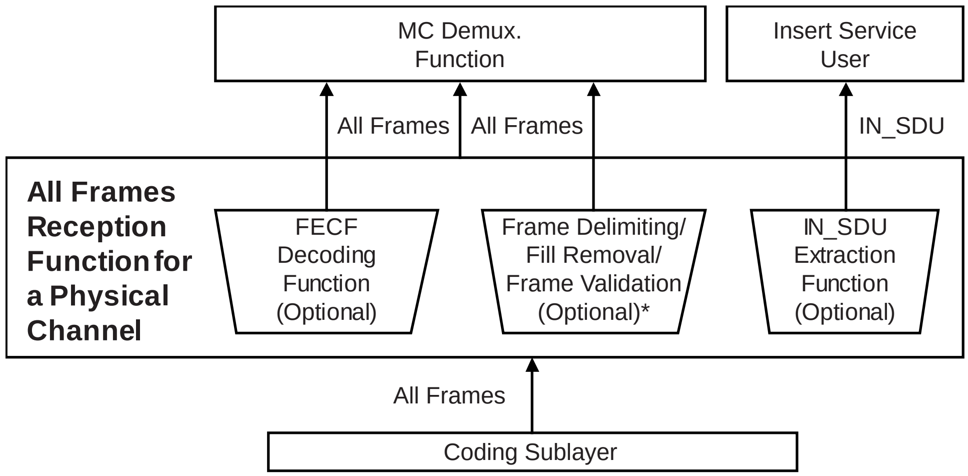
NOTES

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

2 The 'Maximum Number of Frames Given to the C&S Sublayer as a Single Data Unit' is specified as a Managed Parameter in table 5-1.

3 The USLP Frame Validation Check Procedure is used to perform Frame Validation Checks on all Transfer Frames reconstituted by the Frame Delimiting and Fill Data Removal Procedure.

4 An abstract model of the All Frames Reception Function is illustrated in figure 4-32.



*Applicable only when TC Synchronization and Channel Coding (reference [6]) is used.

Figure 4-32: Abstract Model of All Frames Reception Function

4.3.11.2 Frame Delimiting and Fill Data Removal Procedure

NOTES

1 At the sending end, USLP passes one or more Transfer Frames to the TC Synchronization and Channel Coding Sublayer at a time.

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

4.3.11.2.1 The receiving end of this protocol shall receive as an input from the TC Synchronization and Channel Coding Sublayer a series of error-free data octets, corresponding to the decoded Transfer Frame, possibly incomplete or containing fill data.

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

4.3.11.2.2 If one or more valid Frame Length fields isare detected by the Frame Delimiting and Fill Data Removal Procedure, and the number of octets received when the Data Stop condition occurs equals the number of octets specified by the Frame Length(s), then theeach Transfer Frame shall be passed on to the USLP Frame Validation Check Procedure (see 4.3.11.3) as it is delimited.

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

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

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

a) assume that the octets following the final expected octet of the USLP Frame are Fill Data appended by the sending end of the Synchronization and Channel Coding Sublayer to complete the last codeword (see reference [6]);

b) discard that Fill Data;

c) pass the USLP Frame to the USLP Frame Validation Check Procedure (see 4.3.11.3).

NOTES

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

2 If fewer than five trailing octets of Fill Data are present, then they cannot possibly form a Transfer Frame Header, and they will be immediately discarded by this procedure. If five or more trailing octets of Fill Data exist (up to six are possible for the BCH code, up to seven for the (128,64) LDPC code, and up to 31 for the (512,256) LDPC code), this procedure might attempt to interpret the Fill Data as a new Transfer Frame Header. In most cases, the subsequent Frame Validation Checks (see 4.3.11.3) will prevent this from happening because the Fill pattern of ‘01010101’ appearing in each octet will violate at least one of the validation tests; in particular, this pattern appearing where the Frame Length field might be expected will indicate a frame length that exceeds the number of octets received from the Channel Coding Sublayer. The only exception is with the BCH code, if randomization is used, and if the fill data is added after randomization at the transmitting side. In this case, the fill octets will be derandomized without having been randomized, and there is no guarantee that the result will fail the Frame Validation Checks.

5.4 MANAGED PARAMETERS FOR A VIRTUAL CHANNEL

The managed parameters associated with each valid VC shall conform to the definitions in table 5-3.

Table 5-3: Managed Parameters for a Virtual Channel

Managed Parameter	Allowed Values
VC Transfer Frame Type	Fixed Length or Variable Length
VCID	One value from a selectable set of integers (from 0 to 62, with 63 reserved)
VC Count Length for Sequence Control QoS	Integer (maximum 56-bit)
VC Count Length for Expedited QoS	Integer (maximum 56-bit)
COP in Effect	COP-1, COP-P, None
CLCW Version Number	1
CLCW Reporting Rate	as required to support COP-1
MAP IDs	Selectable set of integers (from 0 to 15)
MAP Multiplexing Scheme	Mission Specific
Truncated Transfer Frame Length (octets)	Integer (range 6 to 32—see note 5)
SDU Type (Data Field Content)	CCSDS Packet, VCA_SDU
Inclusion of OCF Allowed (only valid if VC Transfer Frame Type = Variable Length)	True (‘1’), False (‘0’)
Inclusion of OCF Required (only valid if VC Transfer Frame Type = Fixed Length)	True (‘1’), False (‘0’)
Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying service data on the Sequence-Controlled Service	Integer (see note 6)
Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying COP Control Commands	Integer (see note 6)
Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released	Integer

5.5 MANAGED PARAMETERS FOR A MAP CHANNEL

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

Table 5-4: Managed Parameters for a MAP Channel

Managed Parameter	Allowed Values
MAP ID	One value from a selectable set of integers (from 0 to 15)
SDU Type (Data Field Content)	CCSDS Packet, MAPA_SDU, Octet Stream Data
UPID supported	Integer (see reference [14])

5.6 MANAGED PARAMETERS FOR PACKET TRANSFER

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

Table 5-5: Managed Parameters for Packet Transfer

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

6 PROTOCOL SPECIFICATION WITH SDLS OPTION

6.1 OVERVIEW

This section specifies the PDU and the procedures of USLP with support for the SDLS Protocol (reference [15]). If the USLP protocol entity supports SDLS, it has managed parameters for each VC or each MAP to indicate whether SDLS is in use for that channel (see 6.6). Section 4 contains the specification of the protocol without the SDLS option.

6.2 USE OF SDLS PROTOCOL

If SDLS as defined in reference [15] is required over the USLP space data link, then the SDLS protocol shall be used.

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

6.3 USLP TRANSFER FRAME WITH SDLS

6.3.1 OVERVIEW

To support the use of the SDLS security features, a Security Header and a Security Trailer are defined for a USLP Transfer Frame. The use of SDLS can vary between VCs and between MAPs on a Virtual Channel, so a managed parameter defined in table 6-2 of 6.6 indicates the presence of the Security Header. If the Security Header is present, then SDLS is in use for the VC or MAP. This subsection specifies the USLP Transfer Frames on a VC or a MAP that is using SDLS.

If a VC or MAP is not using SDLS, then the USLP Frames are as specified in 4.1.

~~The Security Header and Security Trailer~~Transfer frames containing protocol control information, that is, Protocol Control Command Flag in the Transfer Frame Primary Header set to ‘1’, do not contain the Security Header and Security Trailer and are therefore not protected by SDLS.When SDLS applies, the Security Header and Security Trailer (optional) are placed before and after the TFDF, and they reduce the length of the TFDF compared to a USLP Frame without SDLS. Figure 6-1 compares the USLP Frame fields for a USLP Frame without SDLS and a USLP Frame with SDLS. The upper part of figure 6-1 shows the USLP Transfer Frame without the SDLS fields and is the same as figure 4-1.

NOTE – The presence of the FECF is controlled by the managed parameters (see section 5).

NOTES

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

2 The requirements for the length and contents of the Security Header are specified in reference [15].

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

6.3.5 TRANSFER FRAME DATA FIELD IN A FRAME WITH SDLS

6.3.5.1 The TFDF of a USLP Frame with SDLS shall conform to the specifications of 4.1.4.1.1 through 4.1.4.1.2 as modified by 6.3.5.2.

6.3.5.2 In a Transfer Frame with SDLS, the TFDF shall

a) follow, without gap, the Security Header;

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

b) contain an integer number of octets equal to the Transfer Frame length, minus

1) the lengths of the Transfer Frame Primary Header and of the Security Header;

2) the lengths of the Transfer Frame Insert Zone, of the Security Trailer, OCF, and of the FECF, if any of these are present.

6.3.6 SECURITY TRAILER

If present, the Security Trailer shall follow, without gap, the TFDF.

NOTES

1 The use of the Security Trailer is optional in a USLP Transfer Frame with SDLS. The presence of the Security Trailer is a managed parameter of the VC or MAP (see 6.6).

2 The requirements for the length and contents of the Security Trailer are specified in reference [15].

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

6.4.2 ORDER OF PROCESSING BETWEEN USLP, COP, AND SDLS PROTOCOLS

6.4.2.1 Virtual Channel Generation Function

In the Virtual Channel Generation Function at the sending end, the order of processing between the functions of the USLP, COP, and SDLS protocols shall occur as follows:

a) the Frame Initialization Procedure including SDLS;

b) the SDLS ApplySecurity Function;

c) the FOP, which is a subprocedure of the COP and an integral part of the Virtual Channel Generation Function (final step of processing by the function);

d) the Frame Finalization Procedure including SDLS.

6.4.2.2 Discussion

For completeness, figure 6-2 shows the order of processing between USLP, COP, and SDLS functions at both the sending and receiving ends in the context of the lower layers of the CCSDS protocol stack. In addition, in table 6-1, a reference is provided to the section within each applicable CCSDS document per numbered step in the figure. The detailed processing of the Virtual Channel Generation Function under the SDLS option differs from that of 4.2.7.

6.4.2.3 Frame Initialization Procedure Including SDLS

6.4.2.3.1 The Frame Initialization Procedure (4.2.7.2) shall be applied.

6.4.2.3.2 A Security Header field and, if in use (see 6.6), a Security Trailer field, whose values at this point are not yet known, shall be added to the partially complete TC transfer frame.

6.4.2.4 SDLS ApplySecurity Function

6.4.2.4.1 The SDLS ApplySecurity Function (reference [15]) shall be called to process a USLP Transfer Frame to apply security features to the frame.

6.4.2.4.2 The input parameters of the function shall include the partially formatted frame and the identifiers of the Virtual Channel or the MAP channel.

NOTE – When the function is called, SDLS applies encryption and/or authentication to the data in the frame. The way the Transfer Frame data is passed between the Virtual Channel Generation Function and the SDLS ApplySecurity Function is implementation dependent. Reference [15] defines which transfer frame data fields apply for this function.

6.4.2.5 FOP Procedure

When COP-1 is used, the FOP-1 Procedure specified in reference [9] shall be applied. When COP-P is used, the FOP-P Procedure specified in reference [10] shall be applied.

6.4.2.6 Frame Finalization Procedure Including SDLS

6.4.2.6.1 The Frame Finalization Procedure (4.2.7.4) shall be applied.

6.4.2.6.2 The values provided by the SDLS ApplySecurity Function shall be inserted into the Security Header and, if in use (see 6.6), Security Trailer within the Transfer Frame.

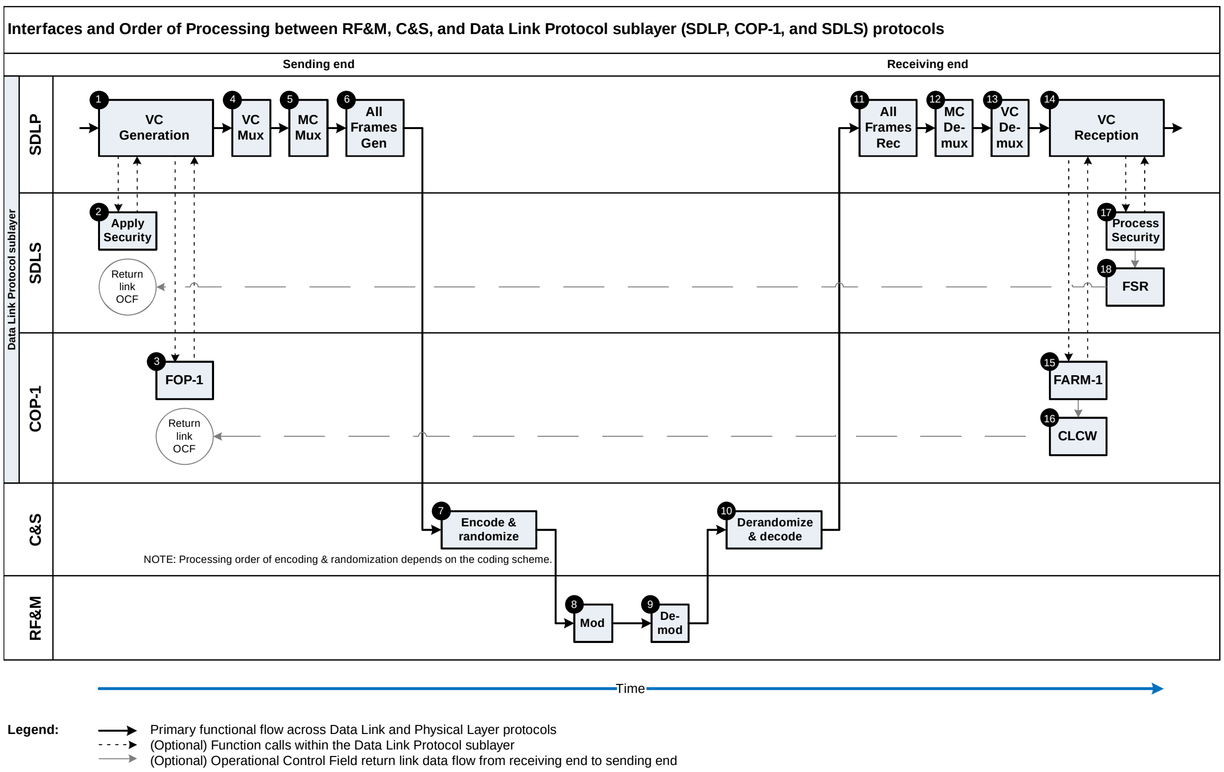


Figure 6-2: Order of Processing between USLP, COP-1, and SDLS Functions

Table 6-1: CCSDS Order of Processing for USLP User Data Frames Using SDLS (Sequential with Figure 6-2)

Numbered Step in Figure 6-2	CCSDS Document	Section Referenced
On the Ground (sending end):		
1. Virtual Channel Generation Function with SDLS	This document	6.4.2.1
Frame Initialization Procedure including SDLS (Virtual Channel Generation function)	This document	6.4.2.3
2. SDLS ApplySecurity Function	This document	6.4.2.4
Encrypt only the Transfer Frame Data Field (if Encryption or Authenticated Encryption selected)	355.0-B-2	4.2.3.3
Populate the Security Header and the optional Security Trailer with the computed MAC (if Authentication or Authenticated Encryption selected)	355.0-B-2	4.2.3.4
3. FOP (Frame Operation Procedure)	This document	6.4.2.5
Frame Finalization Procedure including SDLS (Virtual Channel Generation function)	This document	6.4.2.6
4. Virtual Channel Multiplexing Function	This document	4.2.8
5. Master Channel Multiplexing Function	This document	4.2.10
6. All Frames Generation Function	This document	4.2.11
Compute and add CRC to FECF	This document	4.2.11.3
7. Encode and Randomize the Transfer Frame (When BCH encoding, Randomization is done first; the opposite for LDPC)	231.0-B-4	Section 3 (BCH) or 4 (LDPC) and section 6
8. Modulate onto Subcarrier/Carrier and transmit	401.0-B-32	2.2
On the Spacecraft (receiving end):		
9. Receive and Demodulate	401.0-B-32	2.2
10. Derandomize and Decode the Transfer Frame (Note the order is dependent upon the coding scheme)	231.0-B-4	6.3, 3.5, or 4.5
11. All Frames Reception Function with SDLS	This document	6.5.12
Frame Delimiting and Fill Removal Procedure (invalid code blocks reported by C&S sublayer + fill removal)	This document	4.2.11.2
Frame Validation Check Procedure (includes optional CRC)	This document	4.2.11.3
12. Master Channel Demultiplexing Function	This document	4.3.10
13. Virtual Channel Demultiplexing Function	This document	4.3.8
14. Virtual Channel Reception Function	This document	4.3.7
15. FARM (Frame Acceptance and Reporting Mechanism, subprocedure of the COP)	This document	4.3.7.2
16. CLCW appears within either TM, AOS, or USLP OCF Field	This document	4.3.7.2
17. SDLS ProcessSecurity Function	This document	6.5.2.1 b)
Validate the MAC, if invalid, report security error in Frame Status Report placed into the OCF in telemetry frame (if Authentication or Authenticated Encryption selected)	355.0-B-2	4.2.4.4
Decrypt the Transfer Frame Data Field (if Encryption or Authenticated Encryption selected)	355.0-B-2	4.2.4.5

Numbered Step in Figure 6-2	CCSDS Document	Section Referenced
18. SDLS FSR (Frame Status Report) appears within either TM, AOS, or USLP OCF Field	132.0-B-3 732.0-B-4 This document	4.1.5.5 4.1.5.5 4.1.5.2.2
Thereafter, Transfer Frame Data Units provided to on-board processing (e.g., perform VC or MAP Packet Extraction function or MAP Demultiplexing function or provide to VCA Service User)	N/A	N/A

6.4.3 MAP PACKET PROCESSING FUNCTION WITH SDLS

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

6.4.3.2 When handling MAP Packets on a VC that uses SDLS, the MAP Packet Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAP Packet Processing Function generates a TFDZ to fit exactly within the TFDF (see 4.1.4).

6.4.4 MAPA_SDU GENERATION FUNCTION WITH SDLS

6.4.4.1 The MAPA_SDU Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.3.

6.4.4.2 When handling MAPA_SDUs on a VC that uses SDLS, the MAPA_SDU Generation Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAPA_SDU Generation Function generates a TFDZ to fit exactly within the TFDF (see 4.1.4).

6.4.5 MAP OCTET STREAM PROCESSING FUNCTION WITH SDLS

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

6.4.5.2 When handling Octet Stream Data on a VC that uses SDLS, the MAP Octet Stream Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

NOTE – The MAP Octet Stream Processing Function generates TFDZs to fit exactly within the TFDF (see 4.1.4).

6.4.6 MAP MULTIPLEXING FUNCTION WITH SDLS

The MAP Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.5.

6.4.7 VC PACKET PROCESSING FUNCTION WITH SDLS

6.4.7.1 The VC Packet Processing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.6.

6.4.7.2 When handling Packets on a VC that uses SDLS, the VC Packet Processing Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZ that it generates.

~~6.4.8 VIRTUAL CHANNEL GENERATION FUNCTION WITH SDLS~~

~~**6.4.8.1** When assembling a Transfer Frame, the VC Generation Function shall conform to the specifications of 4.2.6, 6.3, and 6.4.4.2 through 6.4.4.3.~~

~~**6.4.8.2** The Security Header, and the Security Trailer if it is present for the VC, shall be kept empty by the VC Generation Function.~~

~~NOTES~~

~~1 The SDLS ApplySecurity Function specified in reference [15] provides the contents of these security fields as necessary and may modify the contents of the TFDF by encrypting the data.~~

~~2 The lengths of the Security Header and Security Trailer are managed parameters of the VC (see 6.6).~~

~~**6.4.8.3** If the VC Generation Function contains the interface to the SDLS protocol,~~

~~a) it shall call the SDLS ApplySecurity function for the Transfer Frames that it assembles for VCs that use SDLS;~~

~~b) the order of processing between the functions of the USLP and SDLS protocols shall occur as follows in the VC Generation Function:~~

~~1) the USLP Frame assembly processing by the VC Generation Function;~~

~~2) the call by the VC Generation Function to the SDLS ApplySecurity Function.~~

~~NOTE The way in which Transfer Frame data is passed between the VC Generation Function and the SDLS ApplySecurity Function is implementation-dependent.~~

6.4.8 VIRTUAL CHANNEL GENERATION FUNCTION WITH SDLS

6.4.8.1 Discussion

There can be security configurations in which, for example, the physical location of the security processing is not the same for all Virtual Channels, at the sending end or at the receiving end. This case is supported by the order of processing within the Virtual Channel Generation Function. However, the use of multiple Virtual Channels sharing an SDLS Security Association is not supported.

6.4.8.2 Field Lengths and Positions

When assembling a user-data Transfer Frame on a Virtual Channel or MAP that uses SDLS, the Virtual Channel Generation Function shall

a) apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame;

b) conform to the specifications of 4.2.7.

NOTES

1 The lengths of the Security Header and Security Trailer are managed parameters of the Virtual Channel or MAP (see 6.6).

2 The Virtual Channel Generation Function contains the interface to the SDLS protocol. In this case, it calls the SDLS ApplySecurity function for the user data Transfer Frames that it assembles for Virtual Channels or MAPs that use SDLS.

3 The order of processing for the Virtual Channel Generation Function of a USLP protocol entity that supports SDLS is specified in 6.4.2.

6.4.9 VIRTUAL CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The VC Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.8.

~~NOTE There is no interface between the SDLS ApplySecurity function with the USLP VC Multiplexing Function. Multiple data streams may be secured under a single SA by defining multiple secure MAP channels within a single VC.~~

6.4.10 MASTER CHANNEL GENERATION FUNCTION WITH SDLS

The Master Channel Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.9.

6.4.11 MASTER CHANNEL MULTIPLEXING FUNCTION WITH SDLS

The MC Multiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of ~~4.2.9~~4.2.10.

6.4.12 ALL FRAMES GENERATION FUNCTION WITH SDLS

The All Frames Generation Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.11.

~~NOTE There is no interface between the SDLS ApplySecurity function and the USLP 'All Frames Generation' function in order to guarantee that the FECF is computed after the SDLS function has processed the USLP Frame.~~

6.5 RECEIVING END PROTOCOL PROCEDURES WITH SDLS

6.5.1 OVERVIEW

~~When the USLP Transfer Frame Protocol supports the use of the SDLS protocol, there are differences in the receiving end procedures compared to the procedures described in 4.3. This subsection defines those differences.~~

~~The position of the SDLS interface is generally selected to reflect the position of the corresponding interface at the sending end. These choices include the VC Demultiplexing Function or the VC Reception Function, corresponding to the options discussed in 6.4.1.~~

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

~~6.5.2 ERROR REPORTING~~

~~6.5.2.2 Discussion~~

~~Depending on the security features in use, the SDLS ProcessSecurity function specified in reference [15] can verify the authenticity of the USLP Frame and it can decrypt the contents of the TFDF. If the SDLS ProcessSecurity Function detects any errors, these are reported to either the VC Demultiplexing Function or the VC Reception Function. The way that Transfer Frame data is passed between either of these Functions and the SDLS ProcessSecurity Function is implementation-dependent.~~

~~6.5.2.2 Requirements~~

~~**6.5.2.2.1** If the SDLS ProcessSecurity Function does not report an error, the VC Reception Function shall extract the contents of the TFDF from the USLP Frame and deliver it to its user (or Function).~~

~~**6.5.2.2.1** If the SDLS ProcessSecurity Function reports an error, either the VC Demultiplexing Function or the VC Reception Function shall discard the USLP Frame (depending on the interface point).~~

~~NOTE In this case, the optional Verification Status Code parameter can be used to inform the user of the relevant service (see 3.3.2.9, 3.5.2.8, and 3.7.2.7).~~

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

6.5.2.1 Virtual Channel Reception Function

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

a) FARM-1, which is a subprocedure of COP-1 and an integral part of the Virtual Channel Reception Function;

b) the SDLS ProcessSecurity Function;

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

6.5.2.2 Discussion

Figure 6-2 above shows the order of processing between 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:

All Frames Reception Function:

– The All Frames Reception Function is the first procedure that receives valid octets from the Channel Coding sublayer. A decoding failure will cause the Frame Delimiting and Fill Removal Procedure within the All Frames Reception Function to discard all the transfer frames contained within a CLTU.

– The Frame Validation Check Procedure within the All Frames Reception Function includes the check of the FECF, if it is present. A transfer frame that fails any of these checks is discarded.

Virtual Channel Reception Function:

– FARM-1 function: the FARM-1 specified in reference [9] 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.

– SDLS ProcessSecurity function: Depending on the security features in use, the SDLS ProcessSecurity function specified in reference [15] can verify the authenticity of the frame, and it can decrypt the contents of the Transfer Frame Data Field. Any errors detected by the SDLS ProcessSecurity Function are reported to the Virtual Channel Reception Function. The way that Transfer Frame data is passed between the Virtual Channel Reception Function and the SDLS ProcessSecurity Function is implementation dependent.

  If the SDLS ProcessSecurity Function does not report an error, the Virtual Channel Reception Function extracts the Frame Data Unit from the frame and delivers it to its user (or Function). If the SDLS ProcessSecurity Function reports an error, the Virtual Channel Reception Function discards the frame; in this case, the optional Verification Status Code parameter can be used to inform the user of the relevant service. (See 3.3.2.9, 3.4.2.9, 3.5.2.8, and 3.6.2.7.)

NOTES

1 These error conditions are detected either by the FARM and/or by the Frame Validation Check Procedure. To understand the reporting mechanisms with the FARM, one would go either to COP-1 (reference [9]) or COP-P (currently specified in reference [10]) and examine the various frame rejection cases. These are reported in the CLCWs. Additionially and for links without ARQ, frame sequence counters are used to detect transfer frame gaps. For non-ARQ links, gaps are not reported but are detected by the receiving side via the frame sequence counter.

2 Whenever retransmission of Type-AD Transfer Frames is required in a series of frames that originally included both Type-AD and Type-BD frames, the SDLS ProcessSecurity Anti-Replay function will reject any retransmitted frames, if authentication is used, based upon their lower anti-replay sequence count in comparison to the Type-BD anti-replay sequence count (falsely labelling them as SDLS security failures). Therefore, mixing Type-AD and Type-BD frames on the same VC secured by SDLS is generally not advised unless Type-BD frames are sent only after all Type-AD frames have been accepted on board.

6.5.3 MAP PACKET EXTRACTION FUNCTION WITH SDLS

6.5.3.1 The Packet Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.2 ~~and 6.5.3.2~~.

6.5.3.2 When handling Packets on a MAP and VC that uses SDLS, the Packet Extraction Function shall apply the TFDF specification in 6.3.5 to determine the expected length of the TFDZs that it receives.

NOTE – The Packet Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).

~~6.5.4 OCTET STREAM EXTRACTION FUNCTION WITH SDLS~~

~~**6.5.4.1** The Octet Stream Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.2.4 and 6.5.4.2.~~

~~**6.5.4.2** When handling Octet Stream Data on a VC that uses SDLS, the Octet Stream Extraction Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZs that it receives.~~

~~NOTE – The Octet Stream Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).~~

~~6.5.5 VIRTUAL CHANNEL RECEPTION FUNCTION WITH SDLS~~

~~**6.5.5.1** The VC Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.6 and 6.5.5.2 through 6.5.5.3.~~

~~**6.5.5.2** If the VC Reception Function contains the interface to the SDLS protocol, it shall call the SDLS ProcessSecurity function for the Transfer Frames that it handles for VCs that use SDLS.~~

~~**6.5.5.3** When handling a Transfer Frame on a VC that uses SDLS, the VC Reception Function shall apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame.~~

~~6.5.6 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS~~

~~**6.5.6.1** The VC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.7 and 6.5.6.2.~~

~~**6.5.6.2** If the VC Demultiplexing Function contains the interface to the SDLS protocol, it shall call the SDLS ProcessSecurity function for Transfer Frames on VCs that use SDLS, before the demultiplexing is applied.~~

6.5.7 MASTER CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The MC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.8.

6.5.8 ALL FRAMES RECEPTION FUNCTION WITH SDLS

The All Frames Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.10.

6.5.4 MAPA_SDU EXTRACTION FUNCTION WITH SDLS

6.5.4.1 The MAPA_SDU Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.3.

6.5.4.2 When handling MAPA_SDUs on a MAP and VC that uses SDLS, the MAPA_SDU Extraction Function shall apply the TFDF specification in 6.3.5 to determine the expected length of the TFDZs that it receives.

NOTE – The MAPA_SDU Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).

6.5.5 MAP OCTET STREAM EXTRACTION FUNCTION WITH SDLS

6.5.5.1 The MAP Octet Stream Extraction Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.4.

6.5.5.2 When handling MAP Octet Stream Data on a MAP and VC that uses SDLS, the MAP Octet Stream Extraction Function shall apply the TFDF specification in 6.3.5 to determine the length of the TFDZs that it receives.

NOTE – The MAP Octet Stream Extraction Function receives TFDZs that fit exactly within the TFDF (see 4.1.4).

6.5.6 MAP DEMULTIPLEXING FUNCTION WITH SDLS

The MAP Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.5

6.5.7 VC PACKET EXTRACTION FUNCTION WITH SDLS

The VC Packet Extraction Function of a USLP protocol entity that supports SDLS shall

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

b) conform to the specifications of 4.3.6.

6.5.8 VIRTUAL CHANNEL RECEPTION FUNCTION WITH SDLS

When handling a user-data Transfer Frame on a Virtual Channel or MAP that uses SDLS, the Virtual Channel Reception Function shall

a) apply the Transfer Frame specification in 6.3 to determine the lengths and positions of the fields in the Transfer Frame;

b) conform to the specifications of 4.3.7.

NOTES

1 The lengths of the Security Header and Security Trailer are managed parameters of the Virtual Channel or MAP (see 6.6).

2 The Virtual Channel Reception Function contains the interface to the SDLS protocol. In this case, it calls the SDLS ProcessSecurity function for the user data Transfer Frames that it handles for Virtual Channels or MAPs that use SDLS.

6.5.9 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

6.5.9.1 The VC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.8.

6.5.10 MASTER CHANNEL RECEPTION FUNCTION WITH SDLS

The MC Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.9.

6.5.11 MASTER CHANNEL DEMULTIPLEXING FUNCTION WITH SDLS

The MC Demultiplexing Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.10.

6.5.12 ALL FRAMES RECEPTION FUNCTION WITH SDLS

The All Frames Reception Function of a USLP protocol entity that supports SDLS shall conform to the specifications of 4.3.11.

A2.2 REQUIREMENTS LIST

Table A-1: USLP Service Data Units

Item	Description	Reference	Status	Support
USLP-1	Packet SDU	3.2.2	M	
USLP-2	MAPA SDU	3.2.3	M	
USLP-3	VCA SDU	3.2.4	M	
USLP-4	Octet Stream SDU	3.2.5	M	
USLP-5	OCF_SDU	3.2.6	M	
USLP-6	USLP Transfer Frame	3.2.7	M	
USLP-7	Insert Data SDU	3.2.8	M	

Table A-2: Service Parameters

Item	Description	Reference	Status	Values Allowed	Support
MAP Packet Service Parameters					
USLP-8	Packet	3.3.2.2	M		
USLP-9	GMAP ID	3.3.2.3	M		
USLP-10	PVN	3.3.2.4	M		
USLP-11	SDU ID	3.3.2.5	M		
USLP-12	QoS	3.3.2.6	M		
USLP-13	Notification Type	3.3.2.7	O	(see ref. [9])	
USLP-14	Packet Quality Indicator	3.3.2.8	O		
USLP-15	Verification Status Code	3.3.2.9	C2	(see ref. [15])	
VC Packet Service Parameters					
USLP-16	Packet	3.4.2.2	M		
USLP-17	GCVID	3.4.2.3	M		
USLP-18	PVN	3.4.2.4	M		
USLP-19	SDU ID	3.4.2.5	M		
USLP-20	Service Type	3.4.2.6	M		
USLP-21	Notification Type	3.4.2.7	O	(see ref. [9])	
USLP-22	Packet Quality Indicator	3.4.2.8	O		
USLP-23	Verification Status Code	3.4.2.9	C2	(see ref. [15])	
MAPA SDU Service Parameters					
USLP-24	MAPA_SDU	3.5.2.2	M		
USLP-25	GMAP ID	3.5.2.3	M		
USLP-26	SDU ID	3.5.2.4	M		
USLP-27	QoS	3.5.2.5	M		
USLP-28	MAPA_SDU Loss Flag	3.5.2.7	O		
USLP-29	Verification Status Code	3.5.2.8	C2	(see ref. [15])	

Item	Description	Reference	Status	Values Allowed	Support
VCA Service Parameters					
USLP-30	VCA_SDU	3.6.2.2	M		
USLP-31	GVCID	3.6.2.3	M		
USLP-32	SDU ID	3.6.2.4	M		
USLP-33	Service Type	3.6.2.5	M		
USLP-34	Notification Type	3.6.2.6	O	(see ref. [9])	
USLP-35	Verification Status Code	3.6.2.7	C2	(see ref. [15])	
MAP Octet Stream Service Parameters					
USLP-36	Octet Stream Data	3.7.2.2	M		
USLP-37	GMAP ID	3.7.2.3	M		
USLP-38	SDU ID	3.7.2.4	M		
USLP-39	QoS	3.7.2.5	M		
USLP-40	Octet Stream Data Loss Flag	3.7.2.6	O		
USLP-41	Verification Status Code	3.7.2.7	C2	(see ref. [15])	
USLP_MC_OCF Service Parameters					
USLP-42	OCF_SDU	3.8.2.2	M		
USLP-43	GVCID	3.8.2.3	M		
USLP-44	OCF_SDU Loss Flag	3.8.2.4	O		
VCF Service Parameters					
USLP-45	USLP Frame	3.9.2.2	M		
USLP-46	GVCID	3.9.2.3	M		
USLP-47	Frame Loss Flag	3.9.2.4	O		
MCF Service Parameters					
USLP-48	USLP Frame	3.10.2.2	M		
USLP-49	MCID	3.10.2.3	M		
USLP-50	Frame Loss Flag	3.10.2.4	O		
Insert Service Parameters					
USLP-51	IN_SDU	3.11.2.2	M		
USLP-52	Physical Channel Name	3.11.2.3	M		
USLP-53	IN_SDU Loss Flag	3.11.2.4	O		

Table A-3: Service Primitives

Item	Description	Reference	Status	Support
	MAPP Service Primitives			
USLP-61	MAPP.request	3.3.3.2	M	
USLP-62	MAPP_Notify.indication	3.3.3.3	M	
USLP-63	MAPP.indication	3.3.3.4	M	
	VCP Service Primitives			
USLP-64	VCP.request	3.4.3.2	M	
USLP-65	VCP_Notify.indication	3.4.3.3	M	
USLP-66	VCP.indication	3.4.3.4	M	
	MAPA Service Primitives			
USLP-67	MAPA.request	3.5.3.2	M	
USLP-68	MAPA_Notify.indication	3.5.3.3	M	
USLP-69	MAPA.indication	3.5.3.4	M	
	VC Access Service Primitives			
USLP-70	VCA.request	3.6.3.2	M	
USLP-71	VCA_Notify.indication	3.6.3.3	M	
USLP-72	VCA.indication	3.6.3.4	M	
	MAP Octet Stream Service Primitives			
USLP-73	OCTET_STREAM.request	3.7.3.2	M	
USLP-74	OCTET_STREAM.indication	3.7.3.3	M	
USLP-75	OCTET_STREAM_Notify.indication	3.7.3.4	M	
	USLP_MC_OCF Service Primitives			
USLP-76	USLP_MC_OCF.request	3.8.3.2	M	
USLP-77	USLP_MC_OCF.indication	3.8.3.3	M	
	VCF Service Primitives			
USLP-78	VCF.request	3.9.3.2	M	
USLP-79	VCF.indication	3.9.3.3	M	
	MCF Service Primitives			
USLP-80	MCF.request	3.10.3.2	M	
USLP-81	MCF.indication	3.10.3.3	M	
	Insert Service Primitives			
USLP-82	INSERT.request	3.11.3.2	M	
USLP-83	INSERT.indication	3.11.3.3	M	
	COPs Management Service Primitives			
USLP-84	Directive.request	3.12.3.2	M	
USLP-85	Directive_Notify.indication	3.12.3.3	M	
USLP-86	Async_Notify.indication	3.12.3.4	M	

Table A-4: USLP Protocol Data Unit

Item	Description	Reference	Status	Support
USLP-87	USLP Transfer Frame	4.1.1	M	
USLP-88	Transfer Frame Primary Header	4.1.2	M	
USLP-89	Transfer Frame Insert Zone	4.1.3	M	
USLP-90	Transfer Frame Data Field	4.1.4	M	
USLP-91	Operational Control Field	4.1.5	M	
USLP-92	Frame Error Control Field	4.1.6	M	

Table A-5: Protocol Procedures

Item	Description	Reference	Status	Support
USLP-93	MAPP Processing Function	4.2.2	M	
USLP-94	MAPA_SDU Generation Function	4.2.3	M	
USLP-95	MAP Octet Stream Processing Function	4.2.4	M	
USLP-96	MAP Multiplexing Function	4.2.5	M	
USLP-97	Virtual Channel Packet Processing	4.2.6	M	
USLP-98	Virtual Channel Generation Function	4.2.6	M	
USLP-99	Virtual Channel Multiplexing Function	4.2.8	M	
USLP-100	Master Channel Generation Function	4.2.9	M	
USLP-101	Master Channel Multiplexing Function	4.2.10	M	
USLP-102	All Frames Generation Function	4.2.11	M	
USLP-103	MAPP Extraction Function	4.3.2	M	
USLP-104	MAPA_SDU Extraction Function	4.3.3	M	
USLP-105	MAP Octet Stream Extraction Function	4.3.4	M	
USLP-106	MAP Demultiplexing Function	4.3.5	M	
USLP-107	Virtual Channel Packet Extraction Function	4.3.6	M	
USLP-108	Virtual Channel Reception Function	4.3.6	M	
USLP-109	Virtual Channel Demultiplexing Function	4.3.8	M	
USLP-110	Master Channel Reception Function	4.3.9	M	
USLP-111	Master Channel Demultiplexing Function	4.3.10	M	
USLP-112	All Frames Reception Function	4.3.11	M	

Table A-6: Management Parameters

Item	Description	Reference	Status	Values Allowed	Support
Managed Parameters for a Physical Channel					
USLP-113	Physical Channel Name	Table 5-1	M	Character String	
USLP-114	Physical Channel Transfer Frame Type	Table 5-1	M	Fixed Length or Variable Length	
USLP-115	Transfer Frame Length	Table 5-1	M	Integer (C = 4 to 65535 octets) (see 4.1.2.7)	
USLP-116	TFVN	Table 5-1	M	'1100' binary	
USLP-117	MC Multiplexing Scheme	Table 5-1	M		
USLP-118	Presence of Insert Zone	Table 5-1	M	Present ('1'), Absent ('0')	
USLP-119	Insert Zone Length (octets)	Table 5-1	M	Integer, 1–65514 octets	
USLP-120	Presence of Frame Error Control	Table 5-1	M	Present ('1'), Absent ('0')	
USLP-121	[Requirement Deleted]				
USLP-122	Maximum Number of Transfer Frames Given to the Coding and Synchronization Sublayer as a single data unit	Table 5-1	M	~~1~~Integer	
USLP-123	Maximum Value of the Repetitions Parameter to the Coding and Synchronization Sublayer	Table 5-1	M	Integer	
Managed Parameters for a Master Channel					
USLP-124	MC Transfer Frame Type	Table 5-2	M	Fixed Length or Variable Length	
USLP-125	SCID	Table 5-2	M	16-bit Integer	
USLP-126	VCIDs	Table 5-2	M	0 to 62	
USLP-127	VC Multiplexing Scheme	Table 5-2	M		
Managed Parameters for a Virtual Channel					
USLP-128	VC Transfer Frame Type	Table 5-3	M	Fixed Length or Variable Length	
USLP-129	VCID	Table 5-3	M	0 to 62	
USLP-130	VC Count Length for Sequence Control QoS	Table 5-3	M	Integer (max. 56 bit)	
USLP-131	VC Count Length for Expedited QoS	Table 5-3	M	Integer (max. 56 bit)	
USLP-132	COP in Effect	Table 5-3	M	COP-1, COP-P, None	
USLP-133	CLCW Version Number	Table 5-3	M	1	
USLP-134	CLCW Reporting Rate	Table 5-3	M		
USLP-135	MAP IDs	Table 5-3	M	0–15	
USLP-136	MAP Multiplexing Scheme	Table 5-3	O		

Item	Description	Reference	Status	Values Allowed	Support
USLP-137	Truncated Transfer Frame Length	Table 5-3	M	Integer	
USLP-138	SDU Type (Data Field Content)	Table 5-3	M	CCSDS Packet, VCA_SDU	
USLP-139	Inclusion of OCF Allowed (variable-length USLP Frames)	Table 5-3	M	True ('1'), False ('0')	
USLP-140	Inclusion of OCF Required (fixed-length USLP Frames)	Table 5-2	M	True ('1'), False ('0')	
USLP-141	Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying service data on the Sequence-Controlled Service	Table 5-3	M	Integer (see ref. [6])	
USLP-142	Value for the Repetitions parameter to the Coding Sublayer when transferring USLP Frames carrying COP Control Commands	Table 5-3	M	Integer (see ref. [6])	
USLP-143	Maximum delay in milliseconds for a TFDF to be completed, once started, before it must be released	Table 5-3	M	Integer	
USLP-144	Maximum delay in milliseconds between releases of USLP Frames of the same VC	Table 5-3	M	Integer	
Managed Parameters for a MAP Channel					
USLP-145	MAP ID	Table 5-4	M	0–15	
USLP-146	SDU Type	Table 5-4	M	CCSDS Packet, MAPA_SDU, Octet Stream Data	
USLP-147	UPID supported	Table 5-4	M	Integer (see reference [14])	
Managed Parameters for a Packet Transfer					
USLP-148	Valid PVNs	Table 5-5	M	Set of Integers	
USLP-149	Maximum Packet Length (octets)	Table 5-5	M	Integer	
USLP-150	Whether incomplete Packets are required to be delivered to the user at the receiving end	Table 5-5	M	Required, not required	

Table A-7: Protocol Specification with SDLS Option

Item	Description	Reference	Status	Support
USLP-151	SDLS Protocol	(see ref. [15])	O	
USLP-152	Security Header	6.3.4	C3	
USLP-153	Security Trailer	6.3.6	C4	
USLP-154	Transfer Frame Data Field in a USLP Frame with SDLS	6.3.5.2	C3	
USLP-155	Operational Control Field in a USLP Frame with SDLS	6.3.7.2	C3	
USLP-156	Frame Error Control Field in a USLP Frame with SDLS	6.3.8.2	C3	
USLP-157	MAP Packet Processing Function with SDLS	~~6.4.3.~~6.4.3	C3	
USLP-158	MAP Octet Stream Processing Function with SDLS	~~6.4.5.2~~6.4.5	C3	
USLP-159	MAP Multiplexing Function with SDLS	6.4.6	C3	
USLP-160	Virtual Channel Packet Processing with SDLS	6.4.7	C3	
USLP-161	Virtual Channel Generation Function with SDLS	~~6.4.4.2, 6.4.4.3~~6.4.8	C3	
~~USLP-162~~	~~Error reporting~~	~~6.5.2.2~~	~~C4~~	
USLP-163	Virtual Channel Multiplexing Function with SDLS	6.4.9	C3	
USLP-164	Master Channel Generation Function with SDLS	6.4.10	C3	
USLP-165	Master Channel Multiplexing Function with SDLS	6.4.11	C3	
USLP-166	All Frames Generation Function with SDLS	6.4.12	C3	
USLP-167	MAP Packet Extraction Function with SDLS	~~6.5.3.2~~6.5.3	C3	
USLP-168	MAPA SDU Extraction Function with SDLS	6.5.4	C3	
USLP-169	MAP Octet Stream Extraction Function with SDLS	6.5.5	C3	
USLP-170	MAP Demultiplexing Function with SDLS	6.5.6	C3	
USLP-171	Virtual Channel Packet Extraction Function with SDLS	6.5.7	C3	
USLP-172	Virtual Channel Reception Function with SDLS	~~6.5.5.2, 6.5.5.3~~6.5.8	C3	
USLP-173	Virtual Channel Demultiplexing Function with SDLS	~~6.5.6.2~~6.5.9	C3	
USLP-174	Master Channel Reception Function with SDLS	6.5.10	C3	
USLP-175	Master Channel Demultiplexing Function with SDLS	6.5.11	C3	
USLP-176	All Frames Reception Function with SDLS	6.5.12	C3	

C3: M if SDLS Option else N/A.

C4: O if SDLS Option else N/A.

Table A-8: Additional Managed Parameters with SDLS Option

Item	Description	Reference	Status	Values Allowed	Support
USLP-177	Presence of Space Data Link Security Header	Table ~~6-1~~6-2	C5	Present ('1') / Absent ('0')	
USLP-178	Presence of Space Data Link Security Trailer	Table ~~6-1~~6-2	C5	Present ('1') / Absent ('0')	
USLP-179	Length of Space Data Link Security Header (octets)	Table ~~6-1~~6-2	C5	Integer (see ref. [15])	
USLP-180	Length of Space Data Link Security Trailer (octets)	Table ~~6-1~~6-2	C5	Integer (see ref. [15])	

C5: M if SDLS Option else N/A.

ANNEX D

TRUNCATED TRANSFER FRAME

(NORMATIVE)

D1 TRUNCATED TRANSFER FRAME PDU

D1.1 OVERVIEW

The truncated Transfer Frame Protocol Data Unit is provided for transmitting short-length forward-link (reference [10]) or direct-from-Earth telecommands (reference [F6]). There is no current expectation that these frames will be used on the Direct to Earth link for cross-support.

D1.2 GENERAL REQUIREMENTS

D1.2.1 The truncated Transfer Frame shall only be allowed when the managed parameter 'VC Transfer Frame Type' is set to 'Variable Length'.

NOTE – All truncated Transfer Frames for a given Virtual Channel ID will have the same length~~; however, they are expected to be just a subset of the stream of variable-length Transfer Frames provided for that VCID~~.

D1.2.2 The Truncated USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

a) truncated Transfer Frame Primary Header (4 octets; mandatory);

b) Transfer Frame Data Field (integer number of octets; mandatory).

NOTES

1 The truncated Transfer Frame does not contain an Insert Zone, OCF, or FECF.

2 The format of the truncated Transfer Frame without SDLS and with SDLS is shown in figure D-1.

D1.2.3 When SDLS is in use, the Truncated USLP Transfer Frame shall encompass the major fields, positioned contiguously, in the following sequence:

a) truncated Transfer Frame Primary Header (4 octets; mandatory);

b) SDLS Security Header (6 octets; mandatory);

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

d) SDLS Security Trailer (16 octets; mandatory).

D1.2.4 SDLS over Truncated USLP Transfer Frames shall only be allowed when the VC managed parameter ‘Truncated Transfer Frame Length’ is at least 28 octets.

NOTE – SDLS Security Association(s) for use with Virtual Channels that contain truncated Transfer Frames should use a different Security Association (see reference [15]) than that used for Virtual Channels containing non-truncated Transfer Frames.

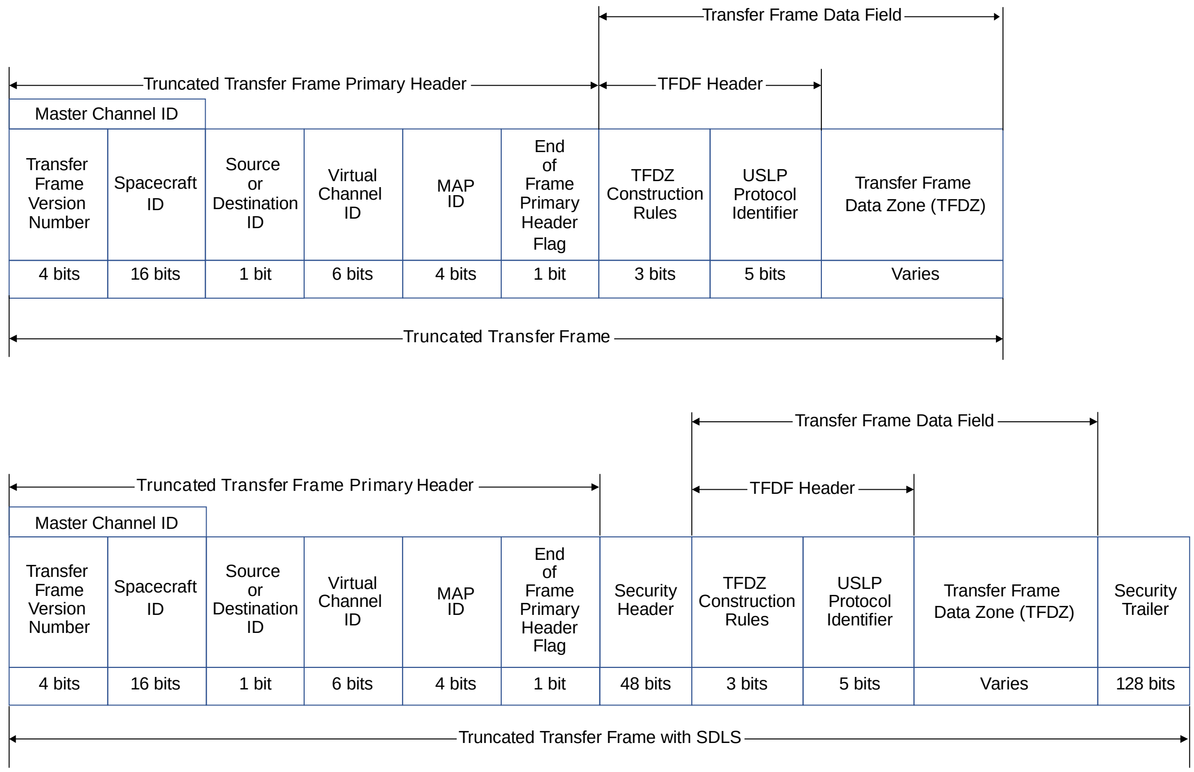


Figure D-1: USLP Truncated Transfer Frame without SDLS and with SDLS

D1.2.5 Truncated and non-truncated USLP Transfer Frames shall not be mixed together within the same CLTU.

D1.3 TRUNCATED TRANSFER FRAME PRIMARY HEADER

D1.3.1 The truncated Transfer Frame Primary Header is mandatory and shall consist of 6 fields, positioned contiguously, in the following sequence:

a) TFVN (4 bits);

b) SCID (16 bits);

c) Source or Destination Identifier (1 bit);

d) VCID (6 bits);

ANNEX F

INFORMATIVE REFERENCES

(INFORMATIVE)

[F1] *Organization and Processes for the Consultative Committee for Space Data Systems*. Issue 4. CCSDS Record (Yellow Book), CCSDS A02.1-Y-4. Washington, D.C.: CCSDS, April 2014.

[F2] *Overview of Space Communications Protocols*. Issue ~~3~~4. Report Concerning Space Data System Standards (Green Book), CCSDS 130.0-G-~~3~~4. Washington, D.C.: CCSDS, ~~July 2014~~April 2023.

[F3] *Space Communications Cross Support—Architecture Description Document*. Issue 1. Report Concerning Space Data System Standards (Green Book), CCSDS 901.0-G-1. Washington, D.C.: CCSDS, November 2013.

[F4] *Space Communications Cross Support—Architecture Requirements Document*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 901.1-M-1. Washington, D.C.: CCSDS, May 2015.

[F5] *Cross Support Reference Model—Part 1: Space Link Extension Services*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 910.4-B-2. Washington, D.C.: CCSDS, October 2005.

[F6] *TC Space Data Link Protocol*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 232.0-B-4. Washington, D.C.: CCSDS, October 2021.

[F7] *The Application of Security to CCSDS Protocols*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 350.0-G-3. Washington, D.C.: CCSDS, March 2019.

[F8] *Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services*. ETSI EN 300 421 V1.1.2 (1997-08). Sophia-Antipolis: ETSI, 1997.

[F9] *A 48/56/64 kbit/s Data Circuit-Terminating Equipment Standardized for Use on Digital Point-to-Point Leased Circuits*. ITU-T Recommendation V.38. Geneva: ITU, 1996.

[F10] *Performance Characteristics for Intermediate Data Rate Digital Carriers Using Convolutional Encoding/Viterbi Encoding*. Rev. 10. IESS 308. Washington, DC: INTELSAT, 2000.

Term	Meaning
Sync	synchronization
TC	telecommand
TFDF	Transfer Frame data field
TFDZ	Transfer Frame data zone
TFVN	Transfer Frame version number
TM	telemetry
UPID	USLP protocol identifier
USLP	Unified Space Data Link Protocol
VC	virtual channel
VCA	virtual channel access
VCF	virtual channel frame
VCID	virtual channel identifier
VCP	virtual channel packet
VN	version number