The Consultative Committee for Space Data Systems

Draft Recommendation for Space Data System Standards

AOS SPACE DATA LINK PROTOCOL

DRAFT RECOMMENDED STANDARD

CCSDS 732.0-P-2.1

PINK SHEETS

December 2008

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

1.7 REFERENCES

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

[1] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1. 2nd ed. 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*. Recommendation for Space Data System Standards, CCSDS 131.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[4] *Space Link Identifiers*. Recommendation for Space Data System Standards, CCSDS 135.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, October 2006.

[5] *CCSDS Global Spacecraft Identification Field Code Assignment Control Procedures*. Recommendation for Space Data System Standards, CCSDS 320.0-B-5. Blue Book. Issue 5. Washington, D.C.: CCSDS, September 2007.

[6] *Space Packet Protocol.* Recommendation for Space Data System Standards, CCSDS 133.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, September 2003.

[7] Encapsulation Service. Recommendation for Space Data System Standards, CCSDS 133.1-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, June 2006.

NOTE – Informative references are listed in annex B.

2.2.3 SUMMARY OF SERVICES

2.2.3.1 Overview

Seven services are provided by the AOS Space Data Link Protocol. Five of them (Virtual Channel Packet, Bitstream, Virtual Channel Access, Virtual Channel Operational Control Field, and Virtual Channel Frame) are provided for a Virtual Channel. One of them (Master Channel Frame) is provided for a Master Channel. One of them (Insert) is provided for all Transfer Frames in a Physical Channel. Table 2-1 summarizes these services.

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

Service	Service Type	Service Data Unit	SAP Address
Virtual Channel Packet (VCP)†	Asynchronous	Packet	GVCID + Packet Version Number
Bitstream	Asynchronous or Periodic	Bitstream Data	GVCID
Virtual Channel Access (VCA)	Asynchronous or Periodic	VCA_SDU	GVCID
Virtual Channel Operational Control Field (VC_OCF)	Synchronous or Periodic	OCF_SDU	GVCID
Virtual Channel Frame (VCF)	Asynchronous or Periodic	Transfer Frame	GVCID
Master Channel Frame (MCF)	Asynchronous or Periodic	Transfer Frame	MCID
Insert	Periodic	IN_SDU	Physical Channel Name

† The term ‘Packet Service’ is used as an abbreviation for Virtual Channel Packet (VCP) Service.

2.2.3.2 Virtual Channel Packet (VCP) Service

The Virtual Channel Packet (VCP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link. The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS. For the Packet Version Numbers presently authorized by CCSDS, see reference [4].

2.2.3.8 Insert Service

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

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

2.2.4 RESTRICTIONS ON SERVICES

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

a) On one Physical Channel, the Insert service shall not exist simultaneously with the Virtual Channel Frame or Master Channel Frame service.

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

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

d) On a Virtual Channel on which the Virtual Channel Frame Service does not exist, only one of the Virtual Channel Packet Service, the Bitstream Service, or the Virtual Channel Access Service shall exist simultaneously.

2.3 OVERVIEW OF FUNCTIONS

2.3.1 GENERAL FUNCTIONS

The AOS Space Data Link Protocol transfers various service data units supplied by sending users encapsulated in a sequence of protocol data units using services of lower layers. The protocol data units, known as AOS Transfer Frames, have a fixed length and must be transferred over a Physical Channel at a constant rate.

The protocol entity performs the following protocol functions:

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

b) segmenting and blocking of service data units to transfer variable-length service data units in fixed-length protocol data units;

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

3 SERVICE DEFINITION

3.1 OVERVIEW

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

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

3.2 SOURCE DATA

3.2.1 SOURCE DATA OVERVIEW

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

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

a) Packet;

b) Bitstream Data;

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

d) Operational Control Field Service Data Unit (OCF_SDU);

e) AOS Transfer Frame;

f) Insert Service Data Unit (IN_SDU).

3.2.2 PACKET

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

3.2.2.2 The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS.

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.

3.2.6 AOS TRANSFER FRAME

NOTE – The AOS Transfer Frame is the fixed-length protocol data unit of the AOS Space Data Link Protocol, but also can be used as the service data units of the Virtual Channel Frame and Master Channel Frame Services. Its format is defined in 4.1 of this Recommendation. The length of any Transfer Frame transferred on a Physical Channel must be the same, and is established by management.

3.2.7 INSERT SERVICE DATA UNIT (IN_SDU)

NOTE – Insert Service Data Units (IN_SDUs) are periodic, octet-aligned data units of fixed length. Their length may be of any constant value which is an integral number of octets, between 1 octet and the maximum length of the data-carrying space of the Transfer Frames, and is established by management. The length of IN_SDUs at the sending interface is always equal to the length at the receiving interface.

3.3 VIRTUAL CHANNEL PACKET (VCP) SERVICE

3.3.1 OVERVIEW OF ~~PACKET~~VCP SERVICE

The Virtual Channel Packet (VCP) Service transfers a sequence of variable-length, delimited, octet-aligned service data units known as Packets across a space link. The Packets transferred by this service must have a Packet Version Number (PVN) authorized by CCSDS. For the Packet Version Numbers presently authorized by CCSDS, see reference [4]. The service is unidirectional, asynchronous and sequence-preserving. It does not guarantee completeness, nor does it signal gaps in the sequence of service data units delivered to a receiving user.

A user of this service is a protocol entity that sends or receives Packets with a single PVN, and identified with the PVN and a 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.

3.3.2 ~~PACKET~~VCP SERVICE PARAMETERS

3.3.2.1 General

The parameters used by the ~~Packet~~VCP Service primitives shall conform to the specifications contained in subsections 3.3.2.2 through 3.3.2.5.

3.3.2.2 Packet

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

NOTE – The Packet parameter is the service data unit transferred by the ~~Packet~~VCP Service. For restrictions on the Packets transferred by the ~~Packet~~VCP Service, see 3.2.2.

3.3.2.3 GVCID

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

NOTE – The GVCID is part of the SAP address of the ~~Packet~~VCP Service.

3.3.2.4 Packet Version Number

The PVN shall identify the protocol entity of the upper layer that uses the ~~Packet~~VCP Service.

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

3.3.2.5 Packet Quality Indicator

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

3.3.3 ~~PACKET~~VCP SERVICE PRIMITIVES

3.3.3.1 General

The service primitives associated with this service are:

a) PACKET.request;

b) PACKET.indication.

3.3.3.2 PACKET.request

3.3.3.2.1 Function

At the sending end, the ~~Packet~~VCP Service user shall pass a PACKET.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.

NOTE – The PACKET.request primitive is the service request primitive for the ~~Packet~~VCP Service.

3.3.3.2.2 Semantics

The PACKET.request primitive shall provide parameters as follows:

PACKET.request	(Packet, GVCID, Packet Version Number)

3.3.3.2.3 When Generated

The PACKET.request primitive shall be passed to the service provider to request it to send the Packet.

3.3.3.2.4 Effect On Receipt

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

3.3.3.2.5 Additional Comments

The PACKET.request primitive shall be used to transfer Packets across the space link on the specified Virtual Channel.

3.3.3.3 PACKET.indication

3.3.3.3.1 Function

At the ~~sending~~receiving end, the service provider shall pass a PACKET.indication to the ~~Packet~~VCP Service user ~~at the receiving end~~ to deliver a Packet.

NOTE – The PACKET.indication primitive is the service indication primitive for the ~~Packet~~VCP Service.

3.3.3.3.2 Semantics

The PACKET.indication primitive shall provide parameters as follows:

PACKET.indication	(Packet, GVCID, Packet Version Number, Packet Quality Indicator (optional))

3.3.3.3.3 When Generated

The PACKET.indication primitive shall be passed from the service provider to the ~~Packet~~VCP Service user at the receiving end to deliver a Packet.

3.3.3.3.4 Effect On Receipt

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

3.3.3.3.5 Additional Comments

The PACKET.indication primitive shall be used to deliver Packets to the ~~Packet~~VCP Service user identified by the GVCID and Packet Version Number. Incomplete Packets may be delivered (optional).

3.4 BITSTREAM SERVICE

3.4.1 OVERVIEW OF BITSTREAM SERVICE

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

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

3.4.2 BITSTREAM SERVICE PARAMETERS

3.4.2.1 General

The parameters used by the Bitstream Service primitives shall conform to the specifications contained in subsections 3.4.2.2 through 3.4.2.4.

3.4.2.2 Bitstream Data

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

NOTE – For restrictions on the Bitstream Data transferred by the Bitstream Service, see 3.2.3.

3.4.2.3 GVCID

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

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

3.4.3.2.4 Effect On Receipt

Receipt of the BITSTREAM.request primitive shall cause the service provider to transfer the Bitstream Data.

3.4.3.2.5 Additional Comments

The BITSTREAM.request primitive shall be used to transfer Bitstream Data across the space link on the specified Virtual Channel.

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

3.4.3.3 BITSTREAM.indication

3.4.3.3.1 Function

At the ~~sending~~receiving end, the service provider shall pass a BITSTREAM.indication to the BITSTREAM Service user ~~at the receiving end~~ to deliver Bitstream Data.

NOTE – The BITSTREAM.indication primitive is the service indication primitive for the Bitstream Service.

3.4.3.3.2 Semantics

The BITSTREAM.indication primitive shall provide parameters as follows:

BITSTREAM.indication (Bitstream Data,
GVCID,
Bitstream Data Loss Flag (optional))

3.4.3.3.3 When Generated

The BITSTREAM.indication primitive shall be passed from the service provider to the Bitstream Service user at the receiving end to deliver Bitstream Data.

3.4.3.3.4 Effect On Receipt

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

3.5.3.2.5 Additional Comments

The VCA.request primitive shall be used to transfer VCA_SDUs across the space link on the specified Virtual Channel.

3.5.3.3 VCA.indication

3.5.3.3.1 Function

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

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

3.5.3.3.2 Semantics

The VCA.indication primitive shall provide parameters as follows:

VCA.indication (VCA_SDU,
GVCID,
VCA_SDU Loss Flag (optional))

3.5.3.3.3 When Generated

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

3.5.3.3.4 Effect On Receipt

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

3.5.3.3.5 Additional Comments

The VCA.indication primitive shall be used to deliver VCA_SDUs to the VCA Service user identified by the GVCID.

3.6.3.3 VC_OCF.indication

3.6.3.3.1 Function

At the ~~sending~~receiving end, the service provider shall pass a VC_OCF.indication to the VC_OCF Service user ~~at the receiving end~~ to deliver an OCF_SDU.

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

3.6.3.3.2 Semantics

The VC_OCF.indication primitive shall provide parameters as follows:

VC_OCF.indication (OCF_SDU,
GVCID,
OCF_SDU Loss Flag (optional))

3.6.3.3.3 When Generated

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

3.6.3.3.4 Effect On Receipt

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

3.6.3.3.5 Additional Comments

The VC_OCF.indication primitive shall be used to deliver OCF_SDUs to the VC_OCF Service user identified by the GVCID.

3.7.3.2.5 Additional Comments

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

3.7.3.3 VCF.indication

3.7.3.3.1 Function

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

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

3.7.3.3.2 Semantics

The VCF.indication primitive shall provide parameters as follows:

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

3.7.3.3.3 When Generated

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

3.7.3.3.4 Effect On Receipt

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

3.7.3.3.5 Additional Comments

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

3.8.3.2.5 Additional Comments

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

3.8.3.3 MCF.indication

3.8.3.3.1 Function

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

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

3.8.3.3.2 Semantics

The MCF.indication primitive shall provide parameters as follows:

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

3.8.3.3.3 When Generated

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

3.8.3.3.4 Effect On Receipt

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

3.8.3.3.5 Additional Comments

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

3.9.3.3 INSERT.indication

3.9.3.3.1 Function

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

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

3.9.3.3.2 Semantics

The INSERT.indication primitive shall provide parameters as follows:

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

3.9.3.3.3 When Generated

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

3.9.3.3.4 Effect On Receipt

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

3.9.3.3.5 Additional Comments

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

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

NOTES

1 If only one Virtual Channel is used, these bits are set permanently to value 'all zeros'. A Virtual Channel used for transmission of ~~Idle~~Only Idle Data (OID) Transfer Frames (i.e., frames whose Data Fields contain only idle data—see 4.1.4) is indicated by setting these bits to the reserved value of 'all ones'.

2 There are no restrictions on the selection of Virtual Channel Identifiers except the rules described above. In particular, Virtual Channels are not required to be numbered consecutively.

3 A Transfer Frame on the 'Idle' Virtual Channel may not contain any valid user data within its Transfer Frame Data Field, but it must contain the Insert Zone if the Insert Service is supported.

4.1.2.4 Virtual Channel Frame Count

4.1.2.4.1 Bits 16–39 of the Transfer Frame Primary Header shall contain the Virtual Channel Frame Count.

4.1.2.4.2 This 24-bit field shall contain a sequential binary count (modulo-16,777,216) of each Transfer Frame transmitted within a specific Virtual Channel.

4.1.2.4.3 A resetting of the Virtual Channel Frame Count before reaching 16,777,215 shall not take place unless it is unavoidable.

NOTE – The purpose of this field is to provide individual accountability for each Virtual Channel, primarily to enable systematic Packet extraction from the Transfer Frame Data Field. If the Virtual Channel Frame Count is reset because of an unavoidable re-initialization, the completeness of a sequence of Transfer Frames in the related Virtual Channel cannot be determined.

f) The bit to R-S symbol mapping shall be:

bits in the header	symbol
0,1,2,3	0
4,5,6,7	1
8,9,10,11	2
12,13,14,15	3
40,41,42,43	4
44,45,46,47	5
48,49,50,51	6
52,53,54,55	7
56,57,58,59	8
60,61,62,63	9

NOTES

1 The purpose of this field is to provide a capability for protecting some key elements in the Transfer Frame Primary Header.

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

3 The header error correction code can correct up to and including two symbol errors. This is sufficient to meet the performance of <1×10E-07 Data Fields missing at a 1×10E-05 channel bit error rate, for random bit errors. In the case of convolutional coded channels, in particular when the convolutional coding is interleaved, the Data Field loss rate will drop to 2×10E-05 at an operating point equivalent to a channel bit error rate of 1×10E-05. This is due to the burst errors typical of the convolutional decoders.

4.1.3 TRANSFER FRAME INSERT ZONE

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

4.1.3.2 The presence or absence of the optional Transfer Frame Insert Zone shall be established by management.

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

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

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

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

NOTE – If the Insert Zone is present, management reduces the length of the Transfer Frame Data Field that is available to other service users by an amount equal to the constant length of the Insert Zone.

4.1.4 TRANSFER FRAME DATA FIELD

4.1.4.1 Overview

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

4.1.4.1.2 The Transfer Frame Data Field, which shall contain an integer number of octets, shall have a length which varies and is equal to:

a) the fixed Transfer Frame length which has been selected for use on a particular Physical Channel; minus

b) the length of the Transfer Frame Primary Header plus the length of the Transfer Frame Insert Zone and/or the Transfer Frame Trailer (if any of these are present).

4.1.4.1.3 The Transfer Frame Data Field shall contain one Multiplexing Protocol Data Unit (M_PDU), one Bitstream Protocol Data Unit (B_PDU), one Virtual Channel Access Service Data Unit (VCA_SDU), or Idle Data.

4.1.4.1.4 M_PDUs, B_PDUs, VCA_SDUs, and Idle Data shall not be mixed in a Virtual Channel (i.e., if a Virtual Channel transfers M_PDUs, every Transfer Frame of that Virtual Channel shall contain an M_PDU). Management shall decide whether M_PDUs, B_PDUs or VCA_SDUs are transferred on a particular Virtual Channel, and this decision shall remain static throughout a Mission Phase.

4.1.4.1.5 In the case where no valid Transfer Frame Data Field is available for transmission at release time for a Transfer Frame, a Transfer Frame with a Data Field containing only Idle Data shall be transmitted. Such a Transfer Frame is called an ~~Idle~~OID Transfer Frame. The Virtual Channel ID of an ~~Idle~~OID Transfer Frame shall be set to the value of ‘all ones’ and a project-specified ‘idle’ pattern shall be inserted into the Transfer Frame Data Field.

NOTES

1 Transfer Frames containing Idle Data in their Data Fields are sent to maintain synchronization at the receiver and also to transmit data in the Transfer Frame Insert Zone when there is no Data Field to send.

2 Idle Data in the Transfer Frame Data Field of an ~~Idle~~OID Transfer Frame must not be confused with the Idle Packet specified in reference [6].

3 The idle pattern used in the OID Transfer Frame is project specific, but a random pattern is preferred. Problems with the reception of frames have been encountered because of insufficient randomization.

4.1.4.2 Multiplexing Protocol Data Unit

4.1.4.2.1 Overview

4.1.4.2.1.1 The Multiplexing Protocol Data Unit (M_PDU) shall follow, without gap, the Transfer Frame Primary Header or the Transfer Frame Insert Zone if present.

4.1.4.2.1.2 The length of the M_PDU shall be fixed by management for any particular Virtual Channel, since it is required to fit exactly within the fixed-length Transfer Frame Data Field.

NOTE – The length of M_PDUs carried by a Physical Channel which supports the Insert Service must take into account the fixed length of the optional Insert Zone.

4.1.4.2.1.3 The M_PDU shall be divided as follows:

a) M_PDU Header (2 octets, mandatory);

b) M_PDU Packet Zone (integral number of octets, mandatory).

4.1.4.2.1.4 The M_PDU Header shall be sub-divided as follows:

a) Reserved Spare (5 bits, mandatory);

b) First Header Pointer (11 bits, mandatory).

4.1.4.2.1.5 The format of the M_PDU is shown in figure 4-3.

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

4.1.4.2.3.5 If the M_PDU Packet Zone contains only Idle Data, the First Header Pointer shall be set to 'all ones minus one'.

4.1.4.2.4 M_PDU Packet Zone

4.1.4.2.4.1 The M_PDU Packet Zone shall follow, without gap, the M_PDU Header.

4.1.4.2.4.2 The M_PDU Packet Zone shall contain either Packets or Idle Data (a fixed-length project-specified 'idle' pattern).

4.1.4.2.4.3 Packets shall be inserted contiguously and in forward order into the M_PDU Packet Zone.

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

4.1.4.2.4.4 When insufficient Packets (including Idle Packets) are available at release time of a Transfer Frame of a Virtual Channel carrying M_PDUs, an M_PDU that contains only Idle Data in its Packet Zone shall be generated.

NOTES

1 M_PDUs that contain only Idle Data in their Packet Zones are sent to maintain synchronous transmission of Transfer Frames and also to transmit data in the Operational Control Field on a specific Virtual Channel when there is no Packet to send.

2 An M_PDU that contains only Idle Data in its Packet Zone can be generated whenever it is necessary (even in the middle of transmission of a Packet that is split into multiple M_PDUs).

3 An M_PDU that contains only Idle Data in its Packet Zone must not be confused with the ~~Idle~~OID Transfer Frame defined in ~~4.1.4~~4.1.4.1.5.

4 Idle Data in the M_PDU Packet Zone should not be confused with the Idle Packet specified in reference [4].

NOTE – Because it may be necessary to insert idle data if an insufficient number of Bitstream Data bits have been received before a B_PDU is released for transmission, the Bitstream Data Pointer indicates the location of the last valid user data bit within the B_PDU Bitstream Data Zone (i.e., the boundary between user data and any inserted idle data).

4.1.4.3.3.2 The locations of the bits in the B_PDU Bitstream Data Zone shall be numbered in ascending order. The first bet in this zone is assigned the number 0. The Bitstream Data Pointer shall contain the binary representation of the location of the last valid user data bit within B_PDU Bitstream Data Zone.

4.1.4.3.3.3 If there are no idle data in the Bitstream Data Zone (i.e., the B_PDU contains only valid user data), the Bitstream Data Pointer shall be set to the value 'all ones'.

4.1.4.3.3.4 If there are no valid user data in the Bitstream Data Zone (i.e., the B_PDU contains only idle data), the Bitstream Data Pointer shall be set to the value 'all ones minus one'.

4.1.4.3.4 B_PDU Bitstream Data Zone

4.1.4.3.4.1 The B_PDU Bitstream Data Zone shall follow, without gap, the B_PDU Header.

4.1.4.3.4.2 The Bitstream Data Zone shall contain either a fixed-length block of the user Bitstream Data (possibly terminated with idle data at a location delimited by the Bitstream Data Pointer), or Idle Data (a fixed-length project-specified 'idle' pattern).

4.1.4.3.4.3 When no Bitstream Data are available at release time of a Transfer Frame of a Virtual Channel carrying B_PDUs, a B_PDU that contains only Idle Data in its Data Zone shall be generated.

NOTES

1 B_PDUs that contain only Idle Data in its Data Zone are sent to maintain synchronous transmission of Transfer Frames, and also to transmit data in the Operational Control Field on a specific Virtual Channel when there are no Bitstream Data to send.

2 A B_PDU that contains only Idle Data in its Data Zone must not be confused with the ~~Idle~~OID Transfer Frame defined in 4.1.4.

4.1.5 OPERATIONAL CONTROL FIELD

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

4.1.5.2 The Operational Control Field is optional; its presence or absence is established by management for each Virtual Channel.

NOTES

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

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

4.1.6.2 Frame Error Control Field Encoding Procedure

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

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

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

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

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

where

all arithmetic is modulo 2;

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

n is the number of bits in the encoded message;

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

L(X) is the presetting polynomial given by

$$\cancel{\mathrm{L}(X) = \sum_{i=0}^{15} x^i ;}$$

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

G(X) is the generating polynomial given by

$$G(X) = X^{16} + X^{12} + X^{5} + 1.$$

~~NOTE – The $X^{(n-16)} \cdot$ L(X) term has the effect of presetting the shift register to all '1' state prior to encoding.~~

NOTES

1 The $X^{(n-16)} \cdot$ L(X) term has the effect of presetting the shift register to all '1' state prior to encoding.

2 A possible FECF generator implementation is shown in figure 4-5. For each frame, the shift register cells are initialized to '1'. The ganged switch is in position 1 while the information bits are being transferred and in position 2 for the sixteen FECF bits.

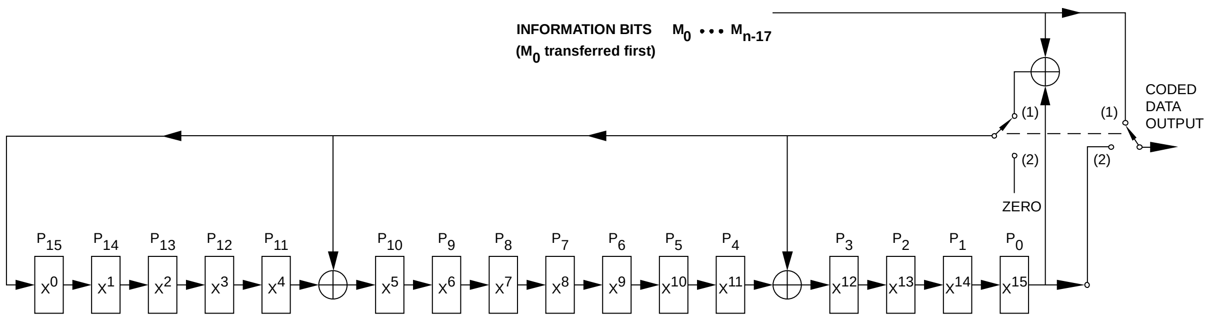

Figure 4-5: Logic Diagram of the Encoder

4.1.6.3 Frame Error Control Field Decoding Procedure

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

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

where

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

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

The received block C*(X) equals the transmitted codeblock C(X) plus (modulo two) the n-bit error block E(X), C*(X) = C(X) + E(X), where both are expressed as polynomials of the same form, i.e., with the most significant bit C_0 or E_0 taken as the binary coefficient of the highest power of X.

NOTE – A possible syndrome polynomial generator implementation is shown in figure 4-6. For each frame, the shift register cells are initialized to ‘1’. The frame includes *n* bits, i.e., (*n*-16) information message bits plus the 16 bits of the FECF. All the *n* bits of the frame are clocked into the input and then the storage stages are examined. For an error-free block, the contents of the shift register cells will be ‘zero’. A non-zero content indicates an erroneous block.

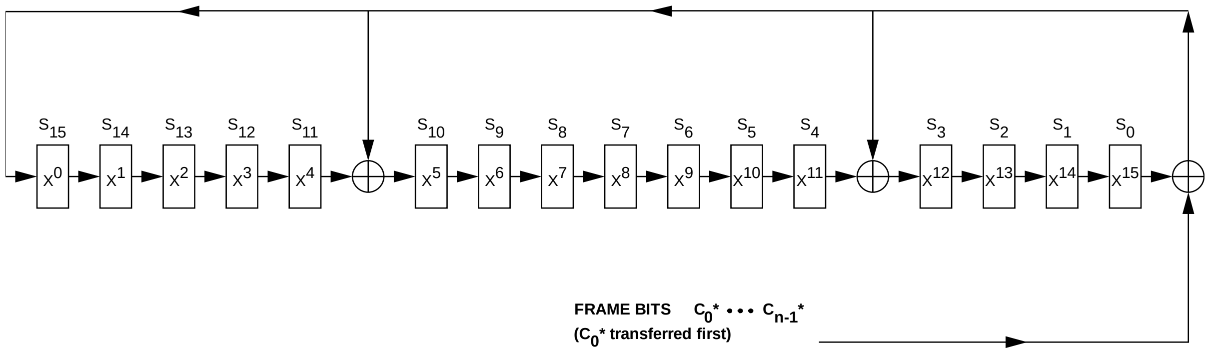

Figure 4-6: Logic Diagram of the Decoder

4.2 PROTOCOL PROCEDURES AT THE SENDING END

4.2.1 OVERVIEW

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

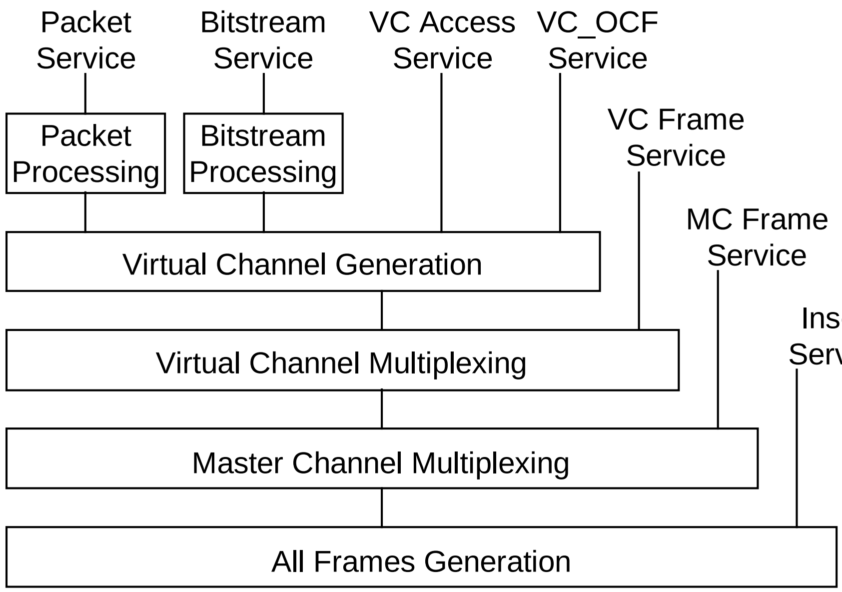

Figure 4-7: Internal Organization of Protocol Entity (Sending End)

4.2.2 PACKET PROCESSING FUNCTION

4.2.2.1 The Packet Processing Function shall be used to transfer variable-length Packets in the fixed-length M_PDU of Transfer Frames.

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

4.2.2.2 The M_PDUs shall be constructed by concatenating Packets together until the maximum M_PDU length is exceeded. Any Packet which exceeds the maximum M_PDU length shall be split, filling the M_PDU completely, and starting a new M_PDU on the same Virtual Channel with the remainder. Construction of the next M_PDU shall continue with the concatenation of Packets until it overflows.

4.2.2.3 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 M_PDUs.

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

4.2.2.5 The Packet Processing Function may generate 'idle' data in the absence of sufficient Packets supplied from the users at release time. The mechanism for generating idle data shall be to create ~~an Idle Packet~~one or more Idle Packets of appropriate lengths~~ defined by reference [~~3~~].~~, where an Idle Packet is either

– an Idle Packet defined by reference [6], or

– a Fill Encapsulation Packet defined by reference [7].

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

4.2.2.6 If it is necessary, the Packet Processing Function may generate an 'idle' M_PDU by setting the First Header Pointer to 'all ones minus one'. An abstract model of the Packet Processing Function is illustrated in figure 4-8.

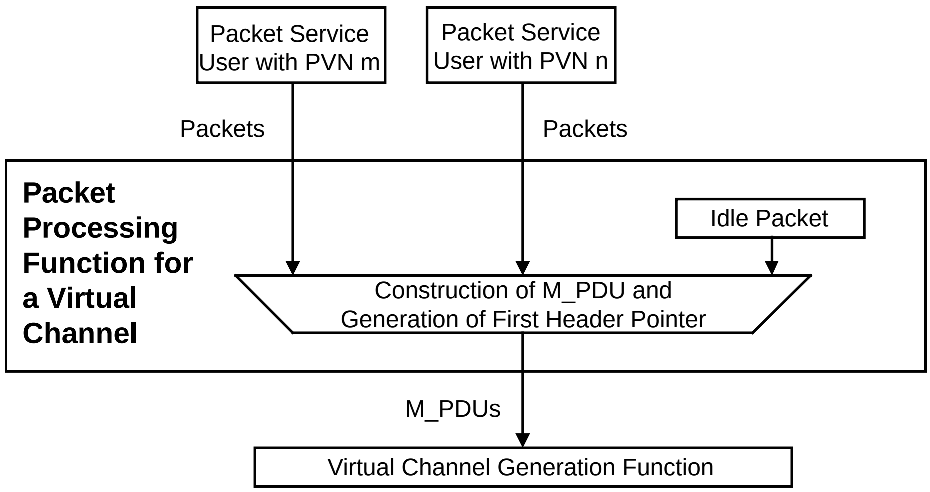

Figure 4-8: Abstract Model of Packet Processing Function

4.2.3 BITSTREAM PROCESSING FUNCTION

4.2.3.1 The Bitstream Processing Function shall be used to transfer variable-length streams of bits in the fixed-length B_PDU of Transfer Frames.

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

4.2.3.2 The Bitstream Processing Function shall be used to fill the Bitstream Data Zone of the B_PDU with the Bitstream Data supplied by the user. Each bit shall be placed sequentially, and unchanged, into the B_PDU Bitstream Data Zone. When the Bitstream Data have filled one particular B_PDU, the continuation of the Bitstream Data shall be placed in a new B_PDU on the same Virtual Channel.

Function. An abstract model of the Virtual Channel Generation Function is illustrated in figure 4-10.

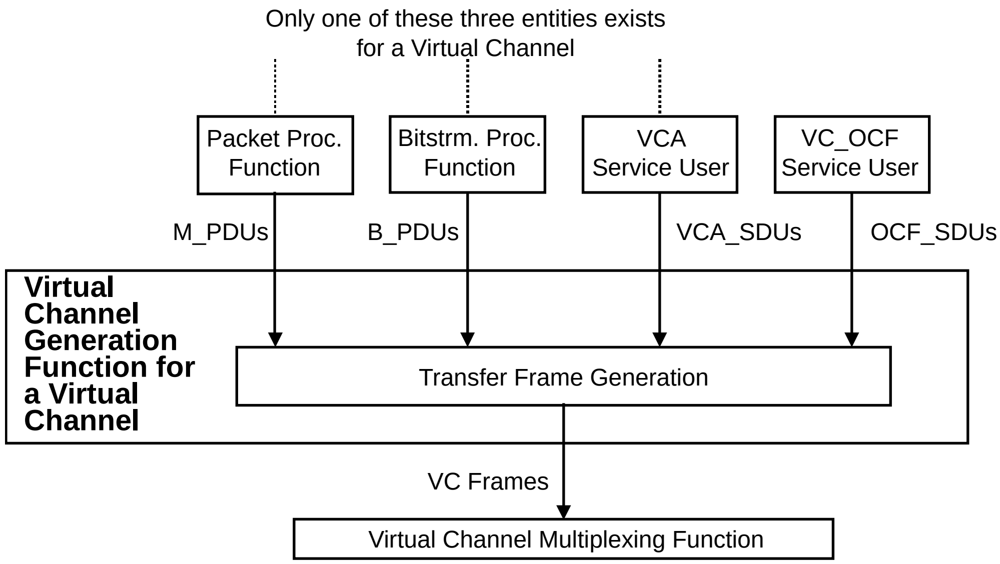

Figure 4-10: Abstract Model of Virtual Channel Generation Function

4.2.5 VIRTUAL CHANNEL MULTIPLEXING FUNCTION

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

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

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

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

4.2.5.4 If there is only one Master Channel on the Physical Channel, then the Virtual Channel Multiplexing Function shall create an ~~Idle~~OID Transfer Frame to preserve the continuity of the transmitted stream in the event that there are no valid Transfer Frames available for transmission at a release time.

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

4.2.5.6 An abstract model of the Virtual Channel Multiplexing Function is illustrated in figure 4-11.

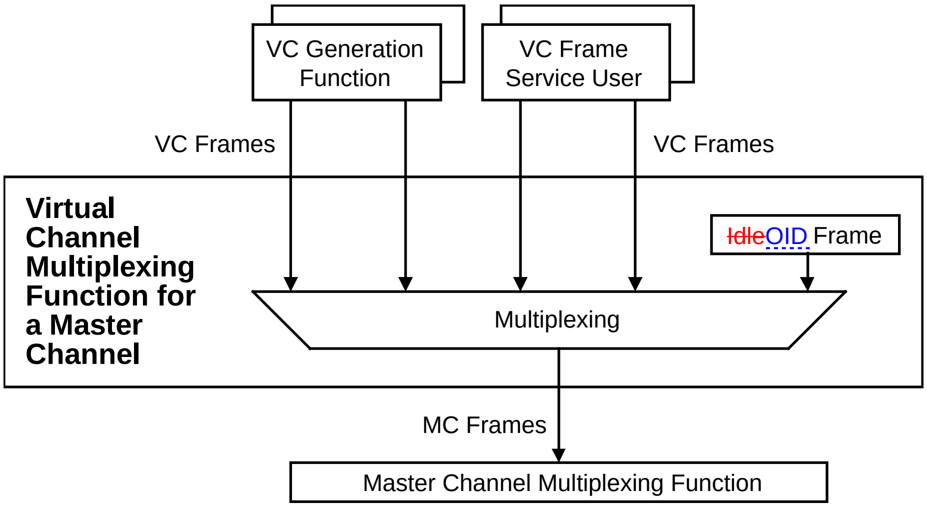

Figure 4-11: Abstract Model of Virtual Channel Multiplexing Function

4.2.6 MASTER CHANNEL MULTIPLEXING FUNCTION

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

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

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

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

4.2.6.4 The Master Channel Multiplexing Function shall create an ~~Idle~~OID Transfer Frame to preserve the continuity of the transmitted stream in the event that there are no valid Transfer Frames available for transmission at a release time. The ~~Idle~~OID Transfer Frame shall have its VCID set to the reserved value of 'all ones' and its MCID set to one of the allowable values. It is not required to maintain a Virtual Channel Frame Count for ~~Idle~~OID Transfer Frames.

4.2.6.5 An abstract model of the Master Channel Multiplexing Function is illustrated in figure 4-12.

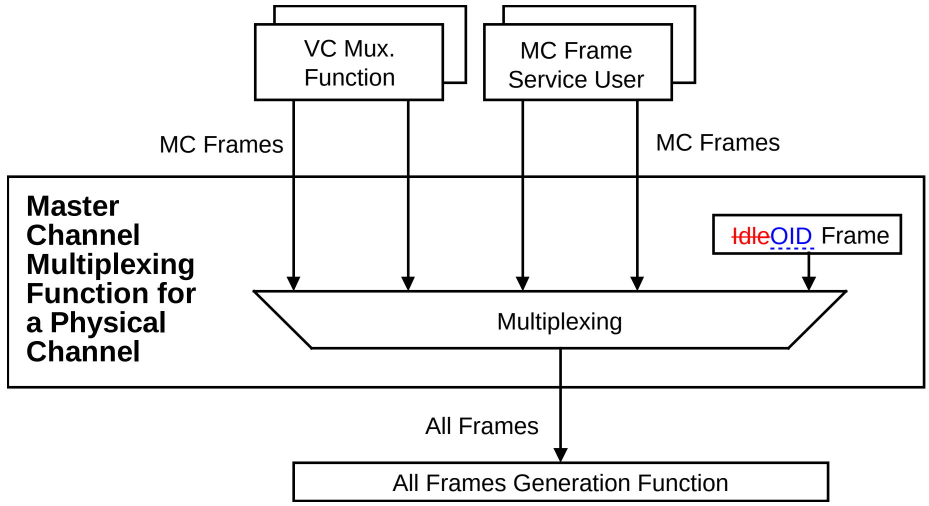

Figure 4-12: Abstract Model of Master Channel Multiplexing Function

4.2.7 ALL FRAMES GENERATION FUNCTION

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

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

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

4.2.7.3 If the Frame Header Error Control is present, check bits shall be generated using the encoding procedure described in 4.1.2.5.3 and added to the Transfer Frame Primary Header. If this field is present, it must be present in all the Transfer Frames transmitted in a particular Physical Channel.

4.2.7.4 If the Frame Error Control Field is present, check bits shall be generated using the encoding procedure described in 4.1.6.2 and inserted into the Transfer Frame Trailer. If this field is present, it must be present in all the Transfer Frames transmitted in a particular Physical Channel.

4.2.7.5 Externally generated Transfer Frames associated with the Virtual Channel Frame and Mater Channel Frame Services shall always bypass the error control encoding functions specified above. The users of these Services must therefore ensure that the Transfer Frames contain an error control option which conforms with that used by the service provider for the

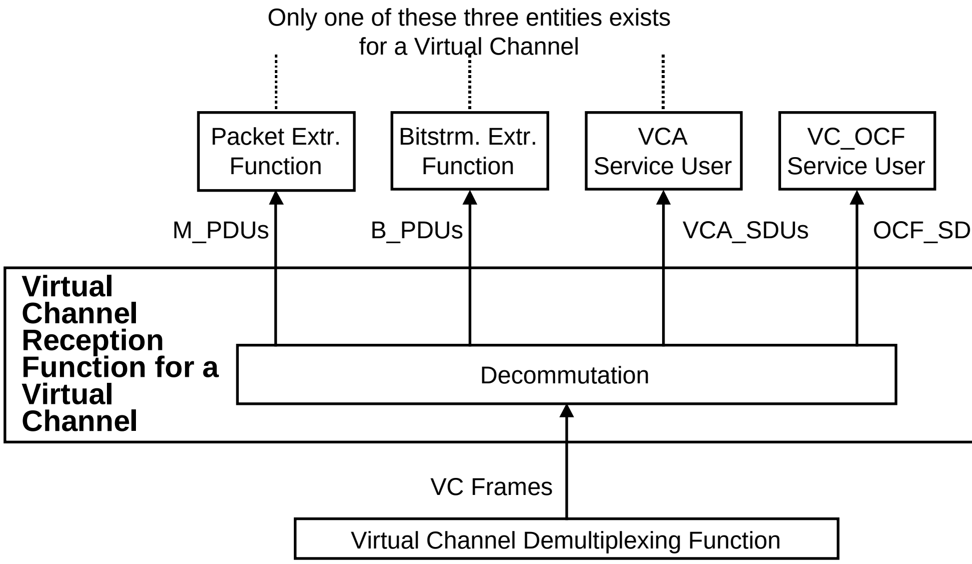

Figure 4-17: Abstract Model of Virtual Channel Reception Function

4.3.5 VIRTUAL CHANNEL DEMULTIPLEXING FUNCTION

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

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

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

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

NOTE – ~~Idle~~OID Transfer Frames are discarded. Transfer Frames with an invalid VCID are also discarded.

4.3.5.4 An abstract model of the Virtual Channel Multiplexing Function is illustrated in figure 4-18.

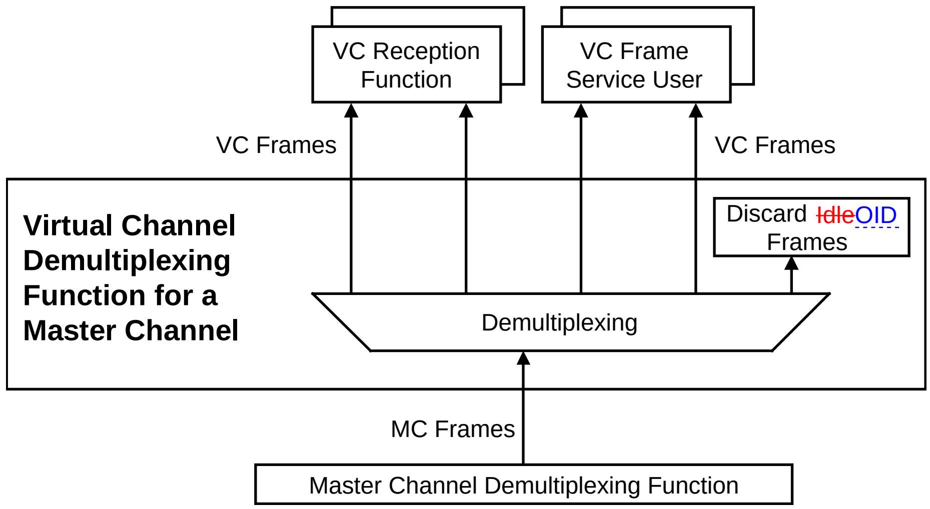

Figure 4-18: Abstract Model of Virtual Channel Demultiplexing Function

4.3.6 MASTER CHANNEL DEMULTIPLEXING FUNCTION

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

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

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

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

NOTE – Transfer Frames with an invalid MCID are discarded.

4.3.6.4 An abstract model of the Master Channel Demultiplexing Function is illustrated in figure 4-19.

5.5 MANAGED PARAMETERS FOR PACKET TRANSFER

Table 5-5 lists the managed parameters associated with a Virtual Channel used for the Virtual Channel Packet ~~Transfer~~ Service.

Table 5-5: Managed Parameters for Packet Transfer

Managed Parameter	Allowed Values
Transfer Frame Version Number	2
Spacecraft ID	Integer
VCID	0, 1, …, 63
Valid Packet Version Numbers	Set of Integers
Maximum Packet Length (octets)	Integer

ANNEX A

ACRONYMS

(This annex **is not** part of the Recommendation)

This annex lists the acronyms used in this Recommendation.

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

C2.10 GRADES OF SERVICE

If SLAP is not used, the Grades of Service specified in reference [B2] only pertain to the selection of coding options. Since the coding options to be used with this Recommendation are specified in reference [3], the notion of Grades of Service is not used in this Recommendation.

C3 TERMINOLOGY CHANGES

Table C-1 lists the terms that have been changed from reference [B2].

Table C-1: Mapping of Terms That Have Been Redefined

Terms Used in Reference [B2]	Terms Used in This Recommendation
Fill Packet	Idle Packet
Fill VCDU	~~Idle~~OID Frame
Isochronous Service	Periodic Service
LinkID	Physical Channel Name
Multiplexing Service	Virtual Channel Packet Service
M_SDU	Packet
VCDU	(AOS) Transfer Frame
VCDU Data Unit Zone	Transfer Frame Data Field
VCDU Error Control Field	Frame Error Control Field
VCDU Header Error Control	Frame Header Error Control
VCDU Insert Zone	Transfer Frame Insert Zone
VCDU Primary Header	Transfer Frame Primary Header
VCDU Service	Virtual Channel Frame Service
VCDU Trailer	Transfer Frame Trailer
VCDU-ID	SCID + VCID
VC_PDU	(AOS) Transfer Frame
Virtual Channel Link Control Sublayer	(No Longer Used)
Virtual Channel Access Sublayer	(No Longer Used)
Virtual Channel Data Unit Counter	Virtual Channel Frame Count