

Category: Draft Intended Standards Track

November 2003

**Ultra Lightweight Encapsulation (ULE) for transmission of
IP datagrams over MPEG-2/DVB networks**

Status of this Draft

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

The MPEG-2 TS has been widely accepted not only for providing digital TV services, but also as a subnetwork technology for building IP networks. This document describes an Ultra Lightweight Encapsulation (ULE) mechanism for the transport of IPv4 and IPv6 Datagrams and other network protocol packets directly over ISO MPEG-2 Transport Streams (TS) as TS Private Data.

Expires May 2004

[page 1]

[RFC EDITOR NOTE:

This section must be deleted prior to publication]

DOCUMENT HISTORY

Draft -00

This draft is intended as a study item for proposed future work by the IETF in this area. Comments relating to this document will be gratefully received by the author(s) and the ip-dvb mailing list at: ip-dvb@erg.abdn.ac.uk

DRAFT 01 (Protocol update)

* Padding sequence modified to 0xFFFF, this change aligns with other usage by MPEG-2 streams. Treatment remains the same as specified for ULE.

* SDNU Format updated to include R-bit (reserved).

* Procedure for TS Packet carrying the final part of a SNDU with either less than two bytes of unused payload updated.

* A Receiver MUST silently discard the remainder of a TS Packet Payload when two or less bytes remain unprocessed following the end of a SNDU, irrespective of the PUSI value in the received TS Packet. It MUST NOT record an error when the value of the remaining byte(s) is identical to 0xFF or 0xFFFF. The receiver MUST then wait for a TS Packet with a PUSI value set to 1.

* Payload Pointer description updated.

* CRC Calculation added.

* Decapsulator processing revised.

* Type field split into two.

* References updated.

* Security considerations added (first draft).

* Appendix added with examples.

DRAFT - 02 (Improvement of clarity)

- * Corrected CRC-32 to follow standard practice in DSM-CC.
- * Removed LLC frame type, now redundant by Bridge-Type (==1)
- * Defined D-bit to use the reserved bit field (R) - Gorry, Alain, Bernhard
- * Changes to description of minimum payload length. - Gorry
- * MPEG-2 Error Indicator SHOULD be used - Hilmar & Gorry
- * MPEG-2 CC MAY be used (since CRC-32 is strong anyway) - Hilmar & Gorry
- * Corrected CRC-32 to now follow standard practice in DSM-CC - Gorry, Hilmar, Alain.
- * Changed description of Encapsulator action for Packing, Gorry & Hilmar.
- * Changed description of Receiver to clarify packing, Gorry & Alain.
- * Stuff/Pad of unused bytes MUST be 0xFF, to align with MPEG - Hilmar/Bernhard.
- * Recommend removal of section on Flushing bit stream - Gorry
- * Updated SNDU figures to reflect D-bit and correct a mistake in the bridged type field - Alain
- * Reorganised [section 5](#) to form sections [5](#) and [6](#), separating encapsulation and receiver processing - Gorry, Hilmar, Alain.
- * Added concept of Idle State and Reassembly State to the Receiver. Renumbered sections [5](#),[6](#) and following, - Gorry.
- * Nits from Alain, Hilmar and Gorry.
Moved security issue on the design of the protocol to appropriate sections, since this is not a concern for deployment: Length field usage and padding initialisation.
- * Changed wording: All multi-byte values in ULE (including Length, Type, and Destination fields) are transmitted in network byte order (most significant byte first) - old NiT from Alain, now fixed.
- * Frame byte size in diagrams now updated to -standard- format, and D bit action corrected, as requested by Alain.

Expires April 2004

[page 3]

* Frame format diagrams, redrawn to 32-bit format below:

0	1															2															3														
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1														

* Additional diagram requested by Alain for D=0 bridging (added, and subsequent figures renumbered).

* Diagrams of encapsulation process, redrawn for clarity (no change to meaning) - Gorry.

* Reworded last para of CRC description.

* Clarification to the statements in the CRC coverage - to make it clear that it is the entire SNDU (header AND payload) that is checksummed. (Fritsche@iabg.de, hlinder@cosy.sbg.ac.at).

* References added for RCS (spotted by Alain) and AAL5 (provided by Anthony Ang).

* Removed informative reference to MPEG part 1 - Alain.
Spelling correction -> Allain to Alain.

* Added description of Receiver processing of the address field.-
Gorry

* Added caution on LLC Length in bridged Packets thanks -
Gorry/wolfgang

* Removed Authors notes from text after their discussion on the list
- Gorry,

* Corrected text to now say maximum value of PP = 182 in ULE -

* Tidied diagrams at end (again) - Gorry.
Authors last call until 22-nov-03 - no new comments.

[END of RFC EDITOR NOTE]

Table of Contents

1. Introduction
 2. Conventions used in this document
 3. Description of method
 4. SNDU Format
 - 4.1 Destination Address Present Field
 - 4.2 Length Field
 - 4.3 End Indicator
 - 4.4 Type Field
 - 4.4.1 Type 1: IANA Assigned Type Fields
 - 4.4.2 Type 2: Ethertype Compatible Type Fields
 - 4.5 SNDU Destination Address Field
 - 4.6 SNDU Trailer CRC
 - 4.7 Description of SNDU Formats
 - 4.7.1 End Indicator
 - 4.7.2 IPv4 SNDU Encapsulation
 - 4.7.3 IPv6 SNDU Encapsulation
 - 4.7.4 Test SNDU
 5. Processing at the Encapsulator
 - 5.1 SNDU Encapsulation
 - 5.2 Procedure for Padding and Packing
 6. Receiver Processing
 - 6.1 Idle State
 - 6.1.1 Reassembly Payload Pointer Checking
 - 6.2 Processing of a Received SNDU
 - 6.2.1 Reassembly Payload Pointer Checking
 - 6.3 Other Error Conditions
 7. Summary
 8. Acknowledgments
 9. Security Considerations
 10. References
 - 10.1 Normative References
 - 10.2 Informative References
 11. Authors' Addresses
 12. IANA Considerations
- [Appendix A.](#)

1. Introduction

This document describes an encapsulation for transport of IP datagrams, or other network layer packets, over ISO MPEG-2 Transport Streams [[ISO-MPEG](#)]. It is suited to services based on MPEG-2, for example the Digital Video Broadcast (DVB) architecture, the Advanced Television Systems Committee (ATSC) system [ATSC; ATSC-G], and other similar MPEG-2 based transmission systems. Such systems typically provide unidirectional (simplex) physical and link layer standards. Support has been defined for a wide range of physical media (e.g. Terrestrial TV [ETSI-DVBT; ATSC-PSIP-TC], Satellite TV [ETSI-DVBS; ATSC-S], Cable Transmission [ETSI-DVBC; ATSC-PSIP-TC]). Bi-directional (duplex) links may also be established using these standards (e.g., DVB defines a range of return channel technologies, including the use of two-way satellite links [[ETSI-RCS](#)] and dial-up modem links [[RFC3077](#)]).

Protocol Data Units, PDUs, (Ethernet Frames, IP datagrams or other network layer packets) for transmission over an MPEG-2 Transport Multiplex are passed to an Encapsulator. This formats each PDU into a Subnetwork Data Unit (SNDU) by adding an encapsulation header and an integrity check trailer. The SNDU is fragmented into a series of TS Packets) that are sent over a single TS Logical Channel.

2. Conventions used in this document

ADAPTATION FIELD: An optional variable-length extension field of the fixed-length TS Packet header, intended to convey clock references and timing and synchronization information as well as stuffing over an MPEG-2 Multiplex [[ISO-MPEG](#)].

AFC: Adaptation Field Control, a pair of bits carried in the TS Packet header that signal the presence of the Adaptation Field and/or TS Packet payload.

ATSC: Advanced Television Systems Committee [[ATSC](#)]. A framework and a set of associated standards for the transmission of video, audio, and data using the ISO MPEG-2 standard.

DSM-CC: Digital Storage Management Command and Control [[ISO-DSMCC](#)]. A format for transmission of data and control information defined by the ISO MPEG-2 standard that is carried in an MPEG-2 Private Section.

DVB: Digital Video Broadcast [[ETSI-DVB](#)]. A framework and set of associated standards published by the European Telecommunications Standards Institute (ETSI) for the transmission of video, audio, and data, using the ISO MPEG-2 Standard.

ENCAPSULATOR: A network device that receives PDUs and formats these into Payload Units (known here as SNDUs) for output as a stream of TS Packets.

MAC: Medium Access and Control. The link layer header of the Ethernet IEEE 802 standard of protocols, consisting of a 6B destination address, 6B source address, and 2B type field.

MPE: Multiprotocol Encapsulation [[ETSI-DAT](#); [ATSC-DAT](#) ; [ATSC-DATG](#)]. A scheme that encapsulates PDUs, forming a DSM-CC Table Section. Each Section is sent in a series of TS Packets using a single TS Logical Channel.

MPEG-2: A set of standards specified by the Motion Picture Experts Group (MPEG), and standardized by the International Standards Organisation (ISO) [[ISO-MPEG](#)]

NPA: Network Point of Attachment. In this document, refers to a 6 B destination address within the MPEG-2 transmission network used to identify individual Receivers or groups of Receivers.

PDU: Protocol Data Unit. Examples of PDU include Ethernet frames, IPv4 or IPv6 datagrams, and other network packets

PES: Programme Elementary Scheme of MPEG-2 [[ISO-MPEG](#)].

Expires April 2004

[page 7]

PID: Packet Identifier. A field carried in the header of TS Packets. This is used to identify the TS Logical Channel to which a TS Packet belongs [\[ISO-MPEG\]](#). The TS Packets forming the parts of a Table Section, PES, or other payload unit must all carry the same PID value. The all 1s PID value indicates a Null TS Packet introduced to maintain a constant bit rate of a TS Multiplex.

PP: Payload Pointer. An optional one byte pointer that directly follows the TS Packet header. It contains the number of bytes between the end of the TS Packet header and the start of a Payload Unit. The presence of the Payload Pointer is indicated by the value of the PUSI bit in the TS Packet header. The Payload Pointer is present in DSM-CC, and Table Sections, it is not present in TS Logical Channels that use the PES-format.

PU: Payload Unit. A sequence of bytes sent using a TS. Examples of Payload Units include: an MPEG-2 Table Section or a ULE SNDU.

PUSI: Payload_Unit_Start_Indicator of MPEG-2 [\[ISO-MPEG\]](#). A single bit flag carried in the TS Packet header. A PUSI value of zero indicates that the TS Packet does not carry the start of a new Payload Unit. A PUSI value of one indicates that the TS Packet does carry the start of a new Payload Unit. In ULE, a PUSI bit set to 1 also indicates the presence of a one byte Payload Pointer (PP).

PRIVATE SECTION: a syntactic structure used for mapping all service information (e.g. an SI table) into TS Packets. A Table may be divided into a number of Table Sections, however all Table Sections must be carried over a single TS Logical Channel.

PSI: Programme SI. An table used to convey information about the service carried in a TS Multiplex. The set of PSI tables is defined by [\[ISO-MPEG\]](#), see also SI Table.

SI TABLE: Service Information Table. In this document, this term describes any table used to convey information about the service carried in a TS Multiplex. SI tables are carried in MPEG-2 private sections.

SNDU: Subnetwork Data Unit. An encapsulated PDU sent as an MPEG-2 Payload Unit.

TABLE SECTION: A Payload Unit carrying a part of a MPEG-2 SI Table.

TS: Transport Stream [\[ISO-MPEG\]](#), a method of transmission at the MPEG-2 level using TS Packets; it represents level 2 of the ISO/OSI reference model. See also TS Logical Channel and TS Multiplex.

TS LOGICAL CHANNEL: Transport Stream Logical Channel, a channel

identified at the MPEG-2 level [[ISO-MPEG](#)]. It exists at level 2 of the ISO/OSI reference model. All packets sent over a TS Logical Channel carry the same PID value. According to MPEG-2, some TS

Logical Channels are reserved for specific signalling purposes. Other standards (e.g., ATSC, DVB) also reserve specific TS Logical Channels.

TS MULTIPLEX: A set of MPEG-2 TS Logical Channels sent over a single common physical link (i.e. a transmission at a specified symbol rate, FEC setting, and transmission frequency). The same TS Logical Channel may be repeated over more than one TS Multiplex, for example to redistribute the same multicast content to two terrestrial TV transmission cells.

TS PACKET: A fixed-length 188B unit of data sent over a TS Multiplex [[ISO-MPEG](#)]. Operation resembles that of cell in an ATM network, and may also be referred to as a TS_Cell. Each TS Packet carries a 4B header, plus optional overhead including an Adaptation Field, encryption details and time stamp information to synchronise a set of related Transport Streams.

3. Description of the Method

PDUs (IP packets, Ethernet frames or packets from other network protocols) are encapsulated to form a Subnetwork Data Unit (SNDU). The SNDU is transmitted over an MPEG-2 transmission network by placing it either in the payload of a single TS Packet. If required, a SNDU may be fragmented into a series of TS Packets. Where there is sufficient space, the method permits a single TS Packet to carry more than one SNDU (or part thereof), sometimes known as Packing. All TS Packets comprising a SNDU MUST be assigned the same PID, and therefore form a part of the same TS Logical Channel.

The ULE encapsulation is limited to TS private streams only. The header of each TS Packet carries a one bit Payload Unit Start Indicator (PUSI) field. The PUSI identifies the start of a payload unit (SNDU) within the MPEG-2 TS Packet payload. The semantics of the PUSI bit are defined differently for PES and PSI packets [ISO-MPEG]; for private data, its use is not defined in the MPEG-2 Standard. In ULE, the operation follows that of PSI packets. Hence, the following PUSI values are defined:

0: The TS Packet does NOT contain the start of a SNDU, but contains the continuation, or end of a SNDU;

1: The TS Packet contains the start of a SNDU, and a one byte Payload Pointer follows the last byte of the TS Packet header.

If a Payload Unit (SNDU) finishes before the end of a TS Packet payload, but it is not convenient to start another Payload Unit, a stuffing procedure fills the remainder of the TS Packet payload with bytes with a value 0xFF [ISO-MPEG2], known as Padding or Stuffing.

A Receiver processing MPEG-2 Table Sections is aware that when it receives a table_id value of 0xFF, this indicates Padding/Stuffing occurred and silently discards the remainder of the TS Packet payload. The payload of the next TS Packet for the same TS Logical Channel will begin with a Payload Pointer of value 0x00, indicating that the next Payload Unit immediately follows the TS Packet header. The ULE protocol resembles this, but differs in the exact procedure (see the following sections).

The TS Packet Header also carries a two bit Adaptation Field Control (AFC) value. The purpose of the adaptation field is primarily to carry timing and synchronisation information and may be used to also include stuffing bytes before a TS Packet payload. Standard Receivers discard TS Packets with an adaptation_field_control field value of '00'. Adaptation Field stuffing is NOT used in this encapsulation method, and TS packets from a ULE Encapsulator MUST be

sent with an AFC value of '01'. Receivers MUST discard TS Packets that carry other AFC values.

Expires April 2004

[page 10]

4. SNDU Format

PDUs (IP packets and bridged Ethernet frames) are encapsulated using ULE to form a SNDU. Each SNDU is sent as an MPEG-2 Payload Unit. The encapsulation format to be used for PDUs is illustrated below:

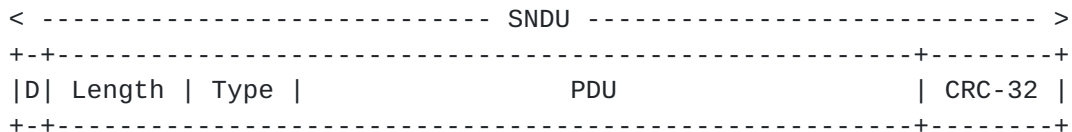


Figure 1: SNDU Encapsulation

All multi-byte values in ULE (including Length, Type, and Destination fields) are transmitted in network byte order (most significant byte first). [Appendix A](#) provides informative examples of usage.

4.1 The Destination Address Present Field

The most significant bit of the Length Field carries the value of the Destination Address Present Field, the D-bit. A value of 0 indicates the presence of the Destination Address Field (see [section 4.5](#)). A value of 1 indicates that a Destination Address Field is not present (i.e. it is omitted).

By default, the D-bit value MUST be set to a value of 0, except for the transmission of an End Indicator (see 4.3), in which this bit MUST be set to the value of 1.

4.2 Length Field

A 15-bit value that indicates the length, in bytes, of the SNDU (encapsulated Ethernet frame, IP datagram or other packet) counted from the byte following the type field up to and including the CRC. Note the special case described in 4.3.

4.3 End Indicator

When the first two bytes of a SNDU has the value 0xFFFF, this denotes an End Indicator (i.e., all 1s length combined with a D-bit value of 1). It indicates to the Receiver that there are no further SNDU are present within the current TS packet (see [section 6](#)), and that no Destination Address Field is present. The value 0xFF has specific semantics in MPEG-2 framing, where it is used to indicate the presence of padding. This use resembles [\[ISO-DSMCC\]](#).

4.4 Type Field

Expires April 2004

[page 11]

The 16-bit Type field indicates the type of payload carried in a SNDU. The set of values that may be assigned to this field is divided into two parts, similar to the allocations for Ethernet.

Ethertypes were originally specified by Xerox under the DIX framework for Ethernet. After specification of IEEE 802.3 [[LLC](#)], the set of Ethertypes less than or equal to 1500 (0x05FC), assumed the role of a length indicator. Ethernet receivers use this feature to discriminate LLC format frames. Hence any IEEE Ethertype ≤ 1500 indicates an LLC frame, and the actual value indicates the length of the LLC frame.

There is a potential security issue when a Receiver receives a PDU with two length fields: The Receiver would need to validate the actual length and the Length field and ensure that inconsistent values are not propagated by the network. Specification of two independent length fields is therefore undesirable. In the ULE header, this is avoided in the SNDU header by including only one length value, but bridging of LLC frames re-introduces this consideration ([section 4.7.5](#)).

The Ethernet LLC mode of identification is not required in ULE, since the SNDU format always carries an explicit Length Field, and therefore the procedure in ULE is modified, as below:

The first set of ULE Type Field values comprise the set of values ≤ 1500 . These Type Field values are IANA assigned (see 4.4.1).

The second set of ULE Type Field values comprise the set of values > 1500 . In ULE, this indicates that the value is identical to the corresponding type codes specified by the IEEE/DIX type assignments for Ethernet and recorded in the IANA EtherType registry.

4.4.1 Type 1: IANA Assigned Type Fields

The first part of the Type space corresponds to the values 0x0000 to 1500 Decimal. These values may be used to identify link-specific protocols and/or to indicate the presence of extension headers that carry additional optional protocol fields (e.g. a bridging encapsulation). Use of these values is co-ordinated by an IANA registry.

The following types are defined:

[XXX IANA ACTION REQUIRED XXX]

0x0000: Test SNDU, discarded by the Receiver.

0x0001: Bridged Ethernet Frame (i.e. MAC source address follows)

[XXX END OF IANA ACTION REQUIRED XXX]

Expires April 2004

[page 12]

The remaining values within the first part of the Type space are reserved for allocation by the IANA.

[Author NOTE: Type allocation and appropriate IANA Procedure to be determined.]

4.4.2 Type 2: Ethertype compatible Type Fields

The second part of the Type space corresponds to the values 1500 Decimal and 0xFFFF. This set of type assignments follow DIX/IEEE assignments (but exclude use of this field as a frame length indicator) [[LLC](#)]. The following types are defined in this document for part 2:

0x0800 : IPv4 Payload (according to IANA EtherTypes)

0x86DD : IPv6 Payload (according to IANA EtherTypes)

All other assignments in part two of this space should be coordinated with the values defined for IANA EtherType encapsulations.

4.5 SNDU Destination Address Field

The SNDU Destination Address Field is optional (see [section 4.1](#)). This field MUST be carried for IP unicast packets destined to routers (i.e. D=0). A sender MAY omit this field (D=1) for an IP unicast packet and/or multicast packets delivered to Receivers that are able to utilise a discriminator field (e.g. the IPv4/IPv6 destination address), which in combination with the PID value, could be interpreted as a Link-Level address.

When the SNDU header indicates the presence of a SNDU Destination Address field (i.e. D=0), a Network Point of Attachment, NPA, field directly follows the SNDU Type Field. NPA destination addresses are 6 B numbers, normally expressed in hexadecimal, used to identify the Receiver(s) in a MPEG-2 transmission network that should process a received SNDU. The value 0x00:00:00:00:00:00, MUST NOT be used as a destination address in a SNDU. The least significant bit of the first byte of the address is set to 1 for multicast frames, and the remaining bytes specify the link layer multicast address. The specific value 0xFF:FF:FF:FF:FF:FF is the link broadcast address, indicating this SNDU is to be delivered to all Receivers.

4.6 SNDU Trailer CRC

Each SNDU MUST carry a 32-bit CRC field in the last four bytes of the SNDU. This position eases CRC computation by hardware. The CRC-32 polynomial is to be used. This is a 32 bit value calculated

Expires April 2004

[page 13]

according to the generator polynomial represented 0x04C11DB7 in hexadecimal:

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+x^0.$$

Examples where this polynomial is also employed include Ethernet, DSM-CC section syntax [ISO-DSMCC] and AAL5 [ITU3563]. The use resembles, but is different to that in SCTP [[RFC3309](#)].

The Encapsulator initialises the CRC-32 accumulator register to the value 0xFFFF FFFF. It then accumulates a transmit value for the CRC32 that includes all bytes from the start of the SNDU header to the end of the SNDU (excluding the 32-bit trailer), and places this in the CRC Field. The Receiver performs an integrity check by independently calculating the same CRC value and comparing this with the transmitted value in the SNDU trailer. SNDUs that do not have a valid CRC, are discarded, causing the Receiver to enter the Idle State.

This description may be suited for hardware implementation, but this document does not imply any specific implementation. Software-based table-lookup or hardware-assisted software-based implementations are also possible.

The primary purpose of this CRC is to protect the SNDU (header, and payload) from undetected reassembly errors and errors introduced by unexpected software / hardware operation while the SNDU is in transit across the MPEG-2 subnetwork and during processing at the encapsulation gateway and/or the receiver. It may also detect the presence of uncorrected errors from the physical link (however, in some cases, these may also be detected by other means).

4.7 Description of SNDU Formats

The format of a SNDU is determined by the combination of the Destination Address Present bit (D) and the SNDU Type Field. The simplest encapsulation places a PDU directly into a SNDU payload. Some Type 1 encapsulations may require additional header fields. These are inserted in the SNDU directly preceding the PDU.

The following SNDU Formats are defined here:

End Indicator: The Receiver should enter the Idle State.
IPv4 SNDU: The payload is a complete IPv4 datagram

IPv6 SNDU: The payload is a complete IPv6 datagram.
Test SNDU: The payload will be discarded by the Receiver.

Bridged SNDU: The payload carries a bridged MAC or LLC frame.

All other formats are currently reserved.

Expires April 2004

[page 14]

4.7.1 End Indicator

The format of the End Indicator is shown in figure 2. This format MUST carry a D-bit value of 1.

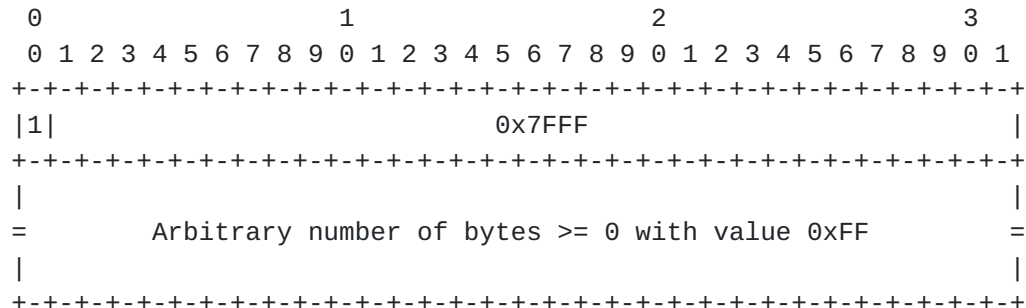


Figure 2: SNDU Format for an End Indicator.

4.7.2 IPv4 SNDU

IPv4 datagrams are transported using one of the two standard SNDU structures, in which the PDU is placed directly in the SNDU payload. The two encapsulations are shown in figures 3 and 4. (Note that in this, and the following figures, the IP datagram payload is of variable size, and is directly followed by the CRC-32).

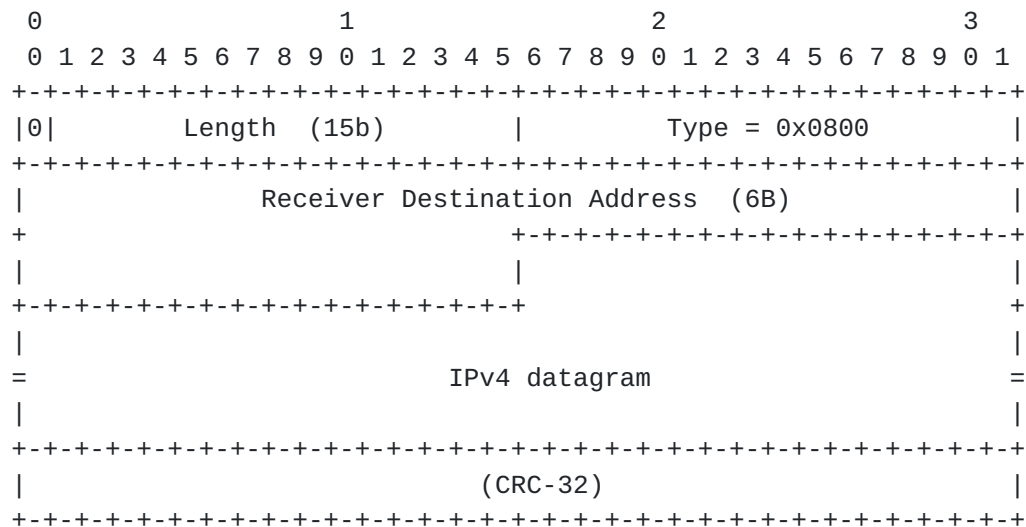


Figure 3: SNDU Format for an IPv4 Datagram using L2 filtering (D=0).

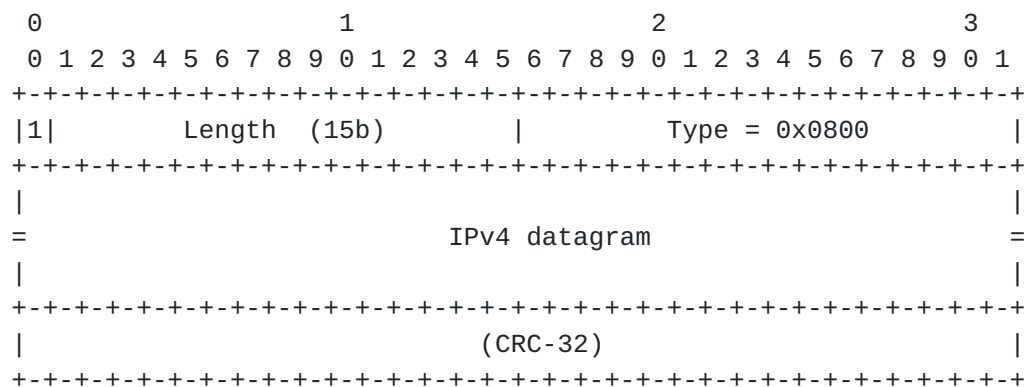


Figure 4: SNDU Format for an IPv4 Datagram using L3 filtering (D=1).

4.7.3 IPv6 SNDU Encapsulation

IPv6 datagrams are transported using one of the two standard SNDU structures, in which the PDU is placed directly in the SNDU payload. The two encapsulations are shown in figures 5 and 6.

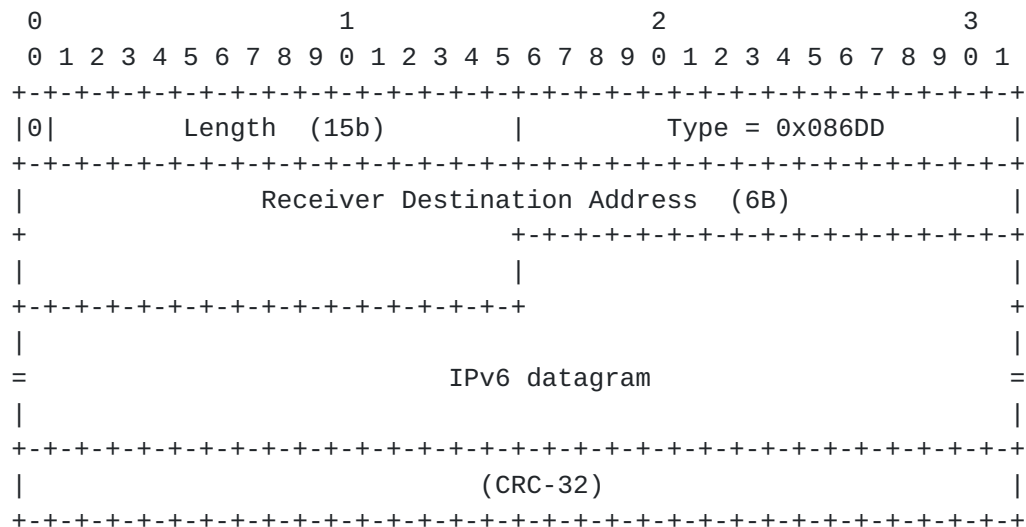


Figure 5: SNDU Format for an IPv6 Datagram using L2 filtering (D=0).

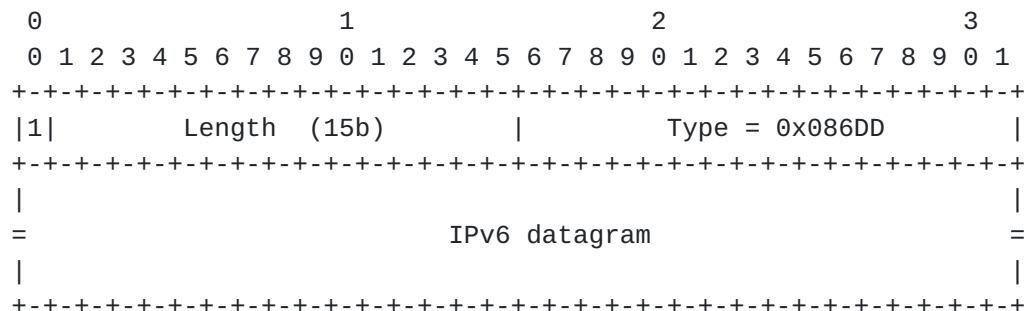


Figure 6: SNDU Format for an IPv6 Datagram using L3 filtering (D=1).

Figure 8: SNDU Format for a Bridged Payload (D=0)

Expires April 2004

[page 17]

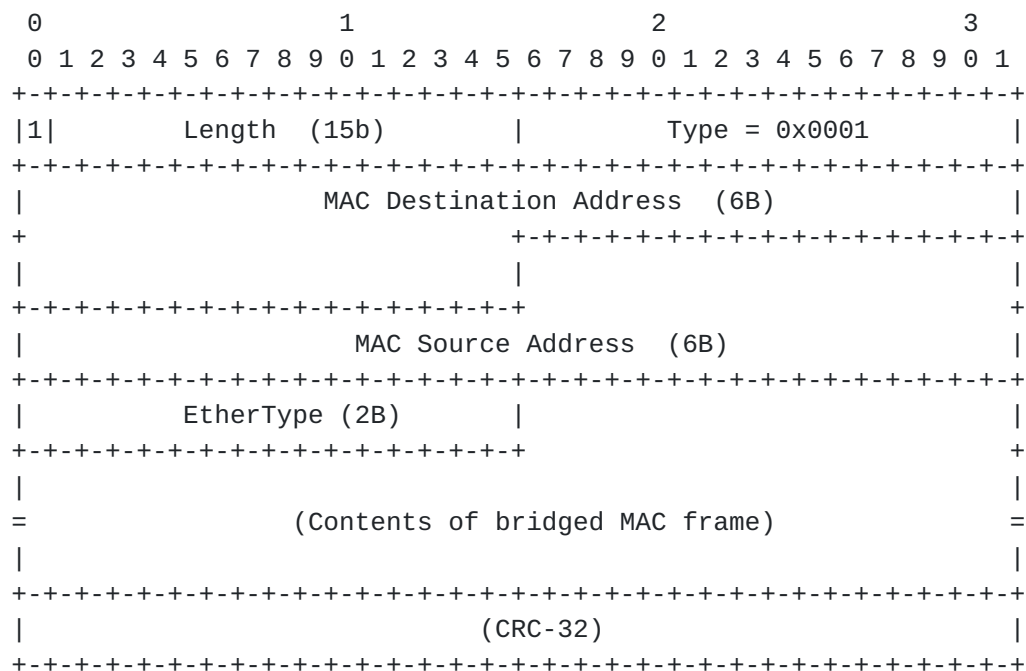


Figure 9: SNDU Format for a Bridged Payload (D=1)

The MAC addresses are those specified in the frame being bridged and are SHOULD be assigned according to the rules specified by the IEEE and may denote unknown, unicast, broadcast, and multicast link addresses. These MAC addresses denote the intended recipient in the destination LAN, and therefore have a different function to the NPA addresses carried in the SNDU header. The EtherType field of frame is defined according to Ethernet/LLC [[LLC](#)].

A frame type <1500 for a bridged frames, introduces a LLC Length Field. The Receiver MUST check this length and discard any frame with a length greater than permitted by the SNDU payload size.

In normal operation, it is expected that any padding appended to the Ethernet frame will be removed prior to forwarding. This requires the sender to be aware of such padding.

Ethernet frames received at the Encapsulator for onward transmission over ULE carry a Local Area Network Frame Check sequence, LAN FCS, field (e.g. CRC-32 for Ethernet). The Encapsulator MUST check the LAN-FCS value of all frames received, prior to further processing. Frames received with an invalid LAN FCS MUST be discarded. After checking, the LAN FCS is then removed (i.e., it is NOT forwarded in the bridged SNDU). As in other ULE frames, the Encapsulator appends a CRC-32 to the transmitted SNDU. At the Receiver, an appropriate LAN-FCS field will be appended to the bridged frame prior to onward transmission on the Ethernet interface.

This design is readily implemented using existing network interface cards, and does not introduce an efficiency cost by transmitting two integrity check fields for bridged frames. However, it also

introduces the possibility that a frame corrupted within the processing performed at an Encapsulator and/or Receiver may not be detected by the final recipient(s) (i.e. such corruption would not normally result in an invalid LAN FCS).

5. Processing at the Encapsulator

The Encapsulator forms the PDUs queued for transmission into SNDUs by adding a header and trailer to each PDU ([section 4](#)). It then segments the SNDU into a series of TS Packet payloads (figure 9). These are transmitted using a single TS Logical Channel over a TS Multiplex. The TS Multiplex may be processed by a number of MPEG-2 (re)multiplexors before it is finally delivered to a Receiver.

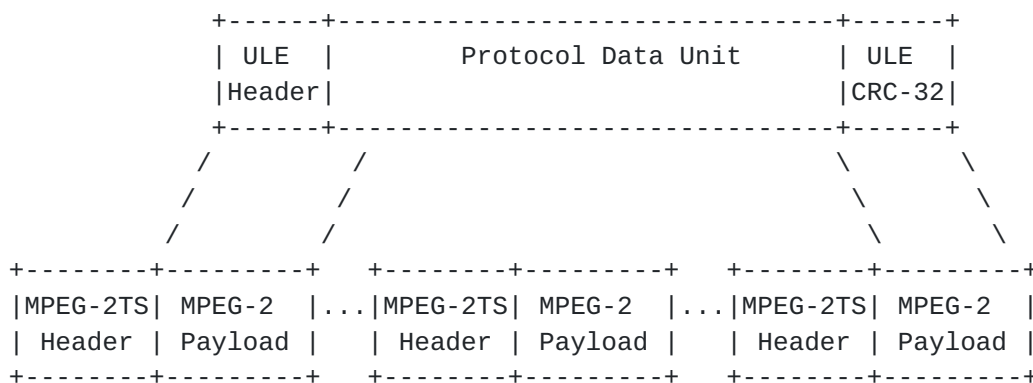


Figure 10: Encapsulation of a SNDU into a series of TS Packets

5.1 SNDU Encapsulation

When an Encapsulator has not previously sent a TS Packet for a specific TS Logical Channel, or after an idle period, it starts to send a SNDU in the first available TS Packet. This first TS Packet generated MUST carry a PUSI value of 1. It MUST also carry a Payload Pointer value of zero indicating the SNDU starts in the first available byte of the TS Packet payload.

The Encapsulation MUST ensure that all TS Packets set the MPEG-2 Continuity Counter carried in the TS Packet header. This value MUST be incremented by one (using modulo arithmetic) for each TS Packet sent using a TS Logical Channel [[ISO-MPEG](#)].

An Encapsulator may decide not to immediately send another SNDU, even if space is available in a partially filled TS Packet. This procedure is known as Padding (figure 11). It informs the Receiver that there are no more SNDUs in this TS Packet payload. The End Indicator is followed by zero or more unused bytes until the end of

the TS Packet payload. All unused bytes MUST be set to the value of 0xFF, following current practice in MPEG-2 [[ISO-DSMCC](#)]. The padding procedure trades decreased efficiency against improved latency.

Expires April 2004

[page 19]

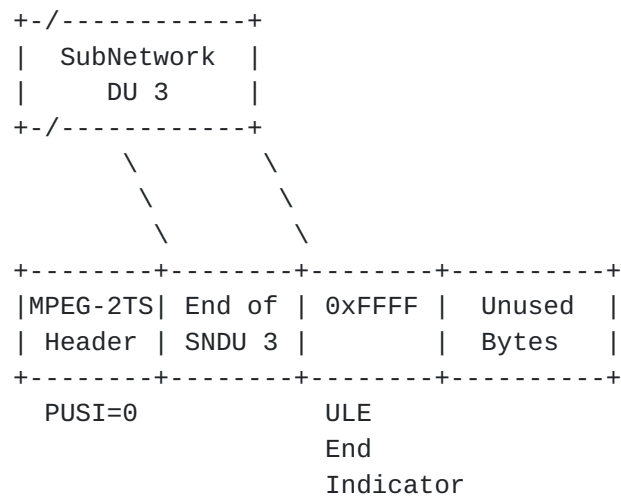


Figure 11: A TS Packet carrying the end of SNDU 3, followed by an End Indicator.

Alternatively, when more packets are waiting at an Encapsulator, and a TS Packet has sufficient space remaining in the payload, the Encapsulator can follow a previously encapsulated SNDU with another SNDU using the next available byte of the TS Packet payload (see 5.2). This is called Packing (figure 12).

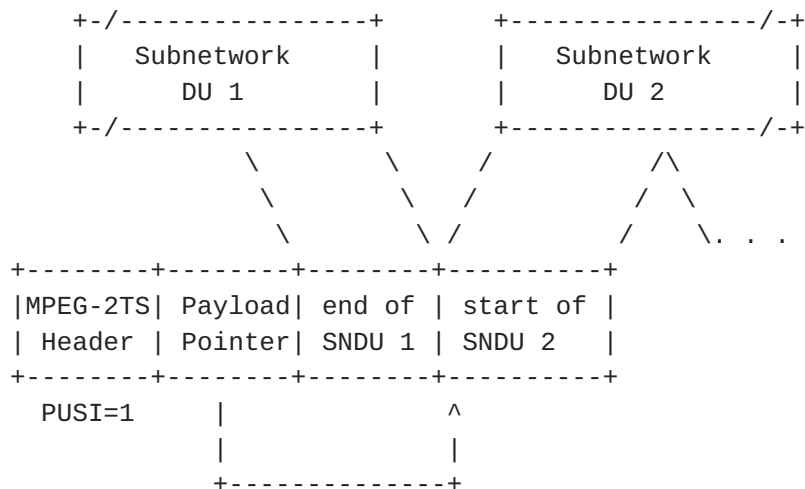


Figure 12: A TS Packet with the end of SNDU 1, followed by SNDU 2.

5.2 Procedure for Padding and Packing

Five possible actions may occur when an Encapsulator has completed encapsulation of an SNDU:

(i) If the TS Packet has no remaining space, the Encapsulator transmits this TS Packet. It starts transmission of the next SNDU in a new TS Packet. (The standard rules require the header of this new

TS Packet to carry a PUSI value of 1, and a Payload Pointer value of 0x00.)

(ii) If the TS Packet carrying the final part of a SNDU has one byte of unused payload, the Encapsulator MUST place the value 0xFF in this final byte, and transmit the TS Packet. This rule provides a simple mechanism to resolve the complex behaviour that may arise when the TS Packet has no PUSI set: To send another SNDU in the current TS Packet, would otherwise require the addition of a Payload Pointer that would consume the last remaining byte of TS Packet payload. The behaviour follows similar practice for other MPEG-2 payload types [[ISO-DSMCC](#)]. The Encapsulator MUST start transmission of the next SNDU in a new TS Packet. (The standard rules require the header of this new TS Packet to carry a PUSI value of 1 and a Payload Pointer value of 0x00.)

(iii) If the TS Packet carrying the final part of a SNDU has exactly two bytes of unused payload, and the PUSI was NOT already set, the Encapsulator MUST place the value 0xFFFF in this final two bytes, providing an End Indicator (4.7.1), and transmit the TS Packet. This rule prevents fragmentation of the SNDU Length Field over two TS Packets. The Encapsulator MUST start transmission of the next SNDU in a new TS Packet. (The standard rules require the header of this new TS Packet to carry a PUSI value of 1 and a Payload Pointer value of 0x00.)

(iv) If the TS Packet has more than two bytes of unused payload, the Encapsulator MAY transmit this partially full TS Packet but MUST first place the value 0xFF in all remaining unused bytes (i.e. setting an End Indicator followed by padding). The Encapsulator MUST start transmission of the next SNDU in a new TS Packet. (The standard rules require the header of this new TS Packet to carry a PUSI value of 1 and a Payload Pointer value of 0x00.)

(v) If at least two bytes are available for Payload data in the TS Packet payload (i.e. three bytes if the PUSI was NOT previously set, and two bytes if it was previously set), the Encapsulator MAY encapsulate further queued PDUs, by starting the next SNDU in the next available byte of the current TS Packet Payload. The PUSI MUST be set. When the Encapsulator packs further SNDUs into a TS Packet where the PUSI has NOT already been set, this requires the PUSI to be updated (set to 1) and an 8-bit Payload Pointer MUST be inserted in the first byte directly following the TS Packet header. The value MUST be set to the position of the byte following the end of the first SNDU in the TS Packet payload. If no further PDUs are available, an Encapsulator MAY wait for additional PDUs to fill the incomplete TS Packet. The maximum period of time an Encapsulator can wait MUST be bounded and SHOULD be configurable by the user. If no

additional PDUs are received after this period of time, it MUST insert an End Indicator instead (using rule iv).

Expires April 2004

[page 21]

Use of the Packing method (v) by an Encapsulation Gateway is optional, and may be determined on a per-session, per-packet, or per-SNDU basis.

When a SNDU is less than the size of a TS Packet payload, a TS Packet may be formed that carries a PUSI value of one and also an End Indicator.

6. Receiver Processing

A Receiver tunes to a specific TS Multiplex and sets a receive filter to accept all TS Packets with a specific PID. These TS Packets are associated with a specific TS Logical Channel and are reassembled to form a stream of SNDUs. A single Receiver may be able to receive multiple TS Logical Channels, possibly using a range of TS Multiplexes. In each case, reassembly is performed independently for each TS Logical Channel. To perform this reassembly, the receiver may use a buffer to hold the partially assembled SNDU, referred to here as the Current SNDU buffer. Other implementations may choose to use other data structures, but must provide equivalent operations.

Receipt of a TS Packet with a PUSI value of 1 indicates that the TS Packet contains the start of a new SNDU. It also indicates the presence of the Payload Pointer (indicating the number of bytes to the start of the first SNDU in the TS-Packet currently being reassembled). It is illegal to receive a Payload Pointer value greater than 182, and this MUST cause the SNDU reassembly to be aborted and the Receiver to enter the Idle State. This event SHOULD be recorded as a payload pointer error.

A Receiver MUST support the use of both the Packing and Padding method for any received SNDU, and MUST support reception of SNDUs with or without a Destination Address Field (i.e. D=0 and D=1).

6.1 Idle State

After initialisation or on receipt of an End Indicator, the Receiver enters the Idle State. In this state, the Receiver discards all TS Packets until it discovers the start of a new SNDU, when it then enters the Reassembly State. Figure 13 outlines these state transitions:

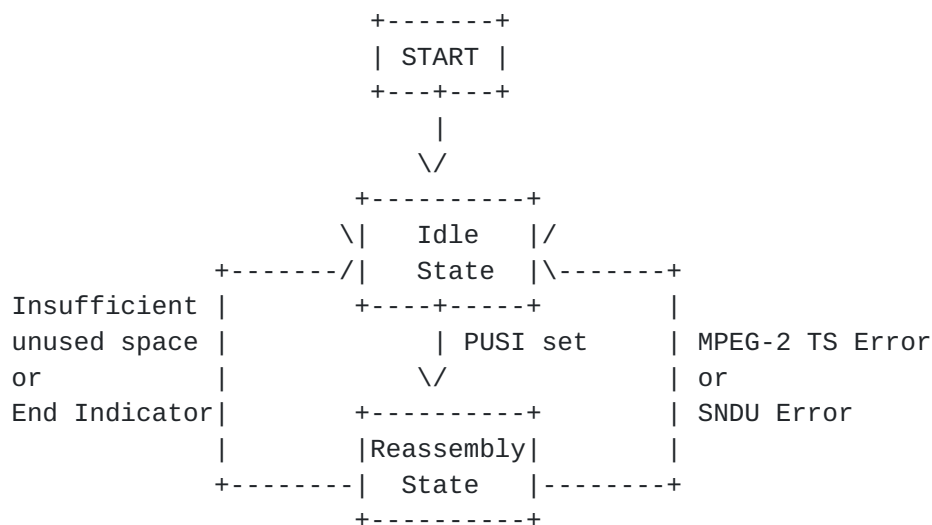


Figure 13: Receiver state transitions

6.1.1 Idle State Payload Pointer Checking

A Receiver in the Idle State MUST check the PUSI value in the header of all received TS Packets. A PUSI value of 1 indicates the presence of a Payload Pointer. For the first TS Packet received, the Payload Pointer will also have a value of 0. Following a loss of synchronisation, values between 1 and 182 are permitted, in which case the receiver MUST discard the number of bytes indicated by the Payload Pointer from the start of the TS Packet payload, before leaving the Idle State. It then enters the Reassembly State, and starts reassembly of a new SNDU at this point.

6.2 Processing of a Received SNDU

When in the Reassembly State, the Receiver reads a 2 byte SNDU Length Field from the TS Packet payload. If the value is less than or equal to 4, or equal to 0xFFFF, the Receiver discards the Current SNDU and the remaining TS Packet payload and returns to the Idle State. Receipt of an invalid Length Field is an error event SHOULD be recorded as an SNDU length error.

If the Length of the Current SNDU is greater than 4, it then accepts bytes from the TS Packet payload to the Current SNDU buffer until either Length bytes in total are received, or the end of the TS Packet is reached. When Current SNDU length equals the value of the Length Field, the receiver MUST calculate and verify the CRC value. SNDUs that contain an invalid CRC value MUST be discarded, causing the Receiver to re-enter the Idle State.

When the Destination Address is present, the Receiver accepts SNDUs that match one of a set of addresses specified by the Receiver (this includes the NPA address of the Receiver, the NPA broadcast address

Expires April 2004

[page 23]

and any required multicast NPA addresses). The Receiver MUST silently discard an SNDU with an unmatched address.

After receiving a valid SNDU, the receiver MUST check the Type Field (and process any Type 1 extensions specified). The SNDU payload is then passed to the next protocol layer specified. An SNDU with an unknown Type value MUST be discarded. This error event SHOULD be recorded as a SNDU type error.

The receiver then starts reassembly of the next SNDU. This MAY directly follow the previously reassembled SNDU within the TS Packet Payload.

(i) If the Current SNDU finishes at the end of a TS Packet payload, the Receiver MUST enter the Idle State.

(ii) If only one byte remains unprocessed in the TS Packet payload after completion of the Current SNDU, the Receiver MUST discard this final byte of TS Packet Payload. It then enters the Idle State. It MUST NOT record an error when the value of the remaining byte is identical to 0xFF.

(iii) If two or more bytes of TS Packet payload data remain after completion of the Current SNDU, the Receiver accepts the next 2 bytes and examines if this is an End Indicator. When an End Indicator is received, a Receiver MUST silently discard the remainder of the TS Packet Payload and transition to the Idle State. Otherwise this is the start of the next Packed SNDU, and the Receiver continues by processing this SNDU.

6.2.1 Reassembly Payload Pointer Checking

A Receiver that has partially received a SNDU (in the Current SNDU buffer) MUST check the PUSI value in the header of all received TS Packets. If it receives a TS Packet with a PUSI value of 1, it MUST then verify the Payload Pointer. If the Payload Pointer does NOT equal the number of bytes remaining to complete the Current SNDU, i.e., the difference between the SNDU Length field and the number of reassembled bytes, the Receiver has detected a delimiting error.

Following a delimiting error, the Receiver MUST discard the partially assembled SNDU (in the Current SNDU buffer), and SHOULD record a reassembly error. It MUST then re-enter the Idle State.

6.3 Other Error Conditions

The Receiver SHOULD check the MPEG-2 Transport Error indicator

carried in the TS Packet header. This flag indicates a transmission error for a TS Logical Channel. If the flag is set to a value of one, a transmission error event SHOULD be recorded. Any partially

received SNDU MUST be discarded. The Receiver then enters the Idle State.

The Receiver MAY also check the MPEG-2 Continuity Counter carried in the TS Packet header. If the Receiver does perform Continuity Counter checking and the received value does not increment by one for successive TS Packets (modulo 16), the Receiver has detected a continuity error. Any partially received SNDU MUST be discarded. A continuity counter error event SHOULD be recorded. The Receiver then enters the Idle State.

7. Summary

This document defines an Ultra Lightweight Encapsulation (ULE) to perform efficient and flexible support for IPv4 and IPv6 network services over networks built upon the MPEG-2 Transport Stream (TS). The encapsulation is also suited to transport of other protocol packets and bridged Ethernet frames.

8. Acknowledgments

This draft is based on a previous draft authored by: Horst D. Clausen, Bernhard Collini-Nocker, Hilmar Linder, and Gorry Fairhurst. The authors wish to thank the members of the ip-dvb mailing list for their input provided. In particular, the many comments received from Patrick Capiere, Wolfgang Fritzsche, and Alain Ritoux. Alain also provided the original examples of usage.

9. Security Considerations

There is a known security issue with un-initialised stuffing bytes. In ULE, these bytes are set to 0xFF.

There are known integrity issues with the removal of the LAN FCS in a bridged networking environment. The removal for bridged frames exposes the traffic to potentially undetected corruption while being processed by the Encapsulator and/or Receiver.

There is a potential security issue when a Receiver receives a PDU with two length fields: The Receiver would need to validate the actual length and the Length field and ensure that inconsistent values are not propagated by the network. In the ULE header, this is avoided by including only one SNDU length value. However, this issue still arises in bridged LLC frames, and frames with a LLC Length greater than the SNDU payload size MUST be discarded.

Expires April 2004

[page 25]

[10](#). References

10.1 Normative References

[ISO-MPEG] ISO/IEC DIS 13818-1 "Information technology -- Generic coding of moving pictures and associated audio information: Systems", International Standards Organisation (ISO).

[RFC2026] Bradner, S., "The Internet Standards Process - Revision 3", [BCP 9](#), [RFC 2026](#), [BCP 9](#), 1996.

[RFC2119] Bradner, S., "Key Words for Use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), 1997.

10.2 Informative References

[ATSC] A/53, "ATSC Digital Television Standard", Advanced Television Systems Committee (ATSC), Doc. A/53, 1995.

[ATSC-DAT] A/90, "ATSC Data Broadcast Standard", Advanced Television Systems Committee (ATSC), Doc. A/090, 2000.

[ATSC-DATG] A/91, "Recommended Practice: Implementation Guidelines for the ATSC Data Broadcast Standard", Advanced Television Systems Committee (ATSC), Doc. A/91, 2001.

[ATSC-G] A/54, "Guide to the use of the ATSC Digital Television Standard", Advanced Television Systems Committee (ATSC), Doc. A/54, 1995.

[ATSC-PSIP-TC] A/65A, "Program and System Information Protocol for Terrestrial Broadcast and Cable", Advanced Television Systems Committee (ATSC), Doc. A/65A, 23 Dec 1997, Rev. A, 2000.

[ATSC-S] A/80, "Modulation and Coding Requirements for Digital TV (DTV) Applications over Satellite", Advanced Television Systems Committee (ATSC), Doc. A/80, 1999.

[CLC99] Clausen, H., Linder, H., and Collini-Nocker, B., "Internet over Broadcast Satellites", IEEE Commun. Mag. 1999, pp.146-151.

[ETSI-DAT] EN 301 192 "Specifications for Data Broadcasting", European Telecommunications Standards Institute (ETSI).

[ETSI-DVBC] EN 300 800 "Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV)", European Telecommunications Standards Institute (ETSI).

Expires April 2004

[page 26]

[ETSI-DVBS] EN 301 421 "Digital Video Broadcasting (DVB); Modulation and Coding for DBS satellite systems at 11/12 GHz", European Telecommunications Standards Institute (ETSI).

[ETSI-DVBT] EN 300 744 "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T)", European Telecommunications Standards Institute (ETSI).

[ETSI-RCS] ETSI 301 791 "Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems", European Telecommunications Standards Institute (ETSI).

[ISO-DSMCC] ISO/IEC IS 13818-6 "Information technology -- Generic coding of moving pictures and associated audio information -- Part 6: Extensions for DSM-CC is a full software implementation", International Standards Organisation (ISO).

[ITU-I363] ITU-T I.363.5 B-ISDN ATM Adaptation Layer Specification Type AAL5, International Standards Organisation (ISO), 1996.

[LLC] "IEEE Logical Link Control" (ANSI/IEEE Std 802.2/ ISO 8802.2), 1985.

[RFC3077] E. Duros, W. Dabbous, H. Izumiyama, Y. Zhang, "A Link Layer Tunneling Mechanism for Unidirectional Links", [RFC3077](#), Proposed Standard, 2001.

[RFC3309] Stone, J., R. Stewart, D. Otis. "Stream Control Transmission Protocol (SCTP) Checksum Change". [RFC3095](#), Proposed Standard, 2001.

[SI-DAT] SI-DAT Group, "Second Draft DVB Specification for Data Broadcasting", Geneva, 1997.

11. Authors' Addresses

Godred Fairhurst
Department of Engineering
University of Aberdeen
Aberdeen, AB24 3UE
UK
Email: gorry@erg.abdn.ac.uk
Web: <http://www.erg.abdn.ac.uk/users/Gorry>

Bernhard Collini-Nocker
Institute of Computer Sciences
University of Salzburg

Jakob Haringer Str. 2
5020 Salzburg
Austria

Expires April 2004

[page 27]

Email: [bnocker]@cosy.sbg.ac.at

Web: <http://www.cosy.sbg.ac.at/cs/>

Full Copyright Statement

"Copyright (C) The Internet Society (date). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

12. IANA Considerations

This document will require IANA involvement.

The payload type field defined in this document must be aligned with an existing IANA registry or the following values need to be assigned by the IANA:

Payload Type Field

ANNEXE A: Informative [Appendix](#)

This appendix provides some examples of use. The appendix is informative. It does not provide a description of the protocol. The examples provide the complete TS Packet sequence for some sample encapsulated IP packets.

The specification of the TS Packet header operation and field values is provided in [ISO-MPEG]. The specification of ULE is provided in the body of this document.

The key below is provided for the following examples.

HDR	4B TS Packet Header
PUSI	Payload Unit Start Indicator
PP	Payload Pointer
***	TS Packet Payload Pointer (PP)

Example A.1: Two 186B PDUs.

SNDU A is 200 bytes (including destination MAC address)
SNDU B is 200 bytes (including destination MAC address)

The sequence comprises 3 TS Packets:

```

                                SNDU
                PP=0                Length
+-----+-----+-----+-----+-----+
| HDR | 0x00 | 0x00 | 0xC8 | ... | A182 |
+-----+-----*-----*-----+-----+
PUSI=1          *      *
                *****

                                SNDU
                PP=16                CRC for A                Length
+-----+-----+-----+-----+-----+-----+-----+-----+
| HDR | 0x10 | A183 | ... | A199 | 0x00 | 0xC0 | ... | B165 |
+-----+-----*-----*-----+-----+-----*-----+-----+
PUSI=1          *                                  *
                *****

                                End                Stuffing
                                CRC for A Indicator    Bytes
+-----+-----+-----+-----+-----+-----+-----+
| HDR | B166 | ... | B199 | 0xFF | 0xFF | ... | 0xFF |
+-----+-----+-----+-----+-----+-----+-----+
PUSI=0

```


Example A.2: Usage of last byte in a TS-Packet

SNDU A is 183 bytes
 SNDU B is 182 bytes
 SNDU C is 181 bytes
 SNDU D is 185 bytes

The sequence comprises 4 TS Packets:

	PP=0	SNDU Length		CRC for A		
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
HDR	0x00	A000	A001	...	A182	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
PUSI=1	*	*				

	PP=0	SNDU Length		CRC for B		Unused byte
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
HDR	0x00	B000	B001	...	B181	0xFF
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
PUSI=1	*	*				

	PP=0	SNDU Length		CRC for C		SNDU Length	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
HDR	0x00	C000	C001	...	C180	D000	D001
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	
PUSI=1	*	*					

			Unused byte		
+-----+	+-----+	+-----+	+-----+	+-----+	
HDR	D002	...	D184	0xFF	
+-----+	+-----+	+-----+	+-----+	+-----+	
PUSI=0					

Example A.3: Large SNDUs

SNDU A is 732 bytes

SNDU B is 284 bytes

The sequence comprises 6 TS Packets:

	PP=0	SNDU Length		
+-----+-----+-----+-----+-----+-----+				
HDR 0x00 A000 A000 ... A182				
+-----+-----+-----+-----+-----+-----+				
PUSI=1	*	*		

+-----+-----+-----+-----+-----+-----+				
HDR A183 ... A366				
+-----+-----+-----+-----+-----+-----+				
PUSI=0				
+-----+-----+-----+-----+-----+-----+				
HDR A367 ... A550				
+-----+-----+-----+-----+-----+-----+				
PUSI=0				
	PP=181	CRC for A	SNDU Length	
+-----+-----+-----+-----+-----+-----+				
HDR 0xB5 A551 ... A731 B000 B001				
+-----+-----+-----+-----+-----+-----+				
PUSI=1	*		*	

+-----+-----+-----+-----+-----+-----+				
HDR B002 ... B186				
+-----+-----+-----+-----+-----+-----+				
PUSI=0				
		End Indicator	Stuffing Bytes	
+-----+-----+-----+-----+-----+-----+				
HDR B187 ... B283 0xFF 0xFF ... 0xFF				
+-----+-----+-----+-----+-----+-----+				
PUSI=0				

Example A.4: Packing of SNDUs

SNDU A is 200 bytes

SNDU B is 60 bytes

SNDU C is 60 bytes

The sequence comprises two TS Packets:

[illegible]