Network Working Group
Request for Comments: 4326
Category: Standards Track

G. Fairhurst
University of Aberdeen
B. Collini-Nocker
University of Salzburg

December 2005

Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS)

Status of This Memo

Copyright Notice

Abstract

The MPEG-2 Transport Stream (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 a Unidirectional Lightweight Encapsulation (ULE) mechanism for the transport of IPv4 and IPv6 Datagrams and other network protocol packets directly over the ISO MPEG-2 Transport Stream as TS Private Data. ULE specifies a base encapsulation format and supports an extension format that allows it to carry additional header information to assist in network/Receiver processing.

Table of Contents

1. Introduction

This document describes an encapsulation for the transport of IP datagrams, or other network-layer packets, over ISO MPEG-2 Transport Streams [ISO-MPEG2, RFC4259]. The encapsulation satisfies the requirement for a lightweight encapsulation defined in section 4 of [RFC4259]. The basic header provides the required set of protocol fields. Extension headers may also be defined. This header structure is significantly simpler to parse and process [SOOR05] than current alternative methods (e.g., MPE [ETSI-DAT], which builds upon the DSM-CC Table Section syntax [ISO-DSMCC]).

The encapsulation 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 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], and 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), such as Ethernet Frames, IP datagrams, or other network-layer packets, used 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 one or more MPEG-2 Transport Stream (TS) Packets that are sent over a single TS Logical Channel.

The MPEG-2 specification [ISO-MPEG2] requires that conformant TS Multiplexes provide Program Specific Information (PSI) for each stream in the TS Multiplex. Other MPEG-2-based transmission standards may also define Service Information (SI).

A format_identifier value has been registered for ULE [ULE1]. This 32 bit number has a hexadecimal value of 0x554C4531. Transport Streams that utilise the Programme Map Table (PMT) defined in ISO 13818-1 [ISO-MPEG2] and that use the ULE format defined in this document, SHOULD insert a descriptor with this value in the PMT ES_info descriptor loop. ULE Streams may also be identified by the stream_type value of 0x91 [ATSC-REG] in a SI/PSI Table [ISO_MPEG2].

This information may allow Receivers and Re-multiplexors [RFC4259] to locate a specific ULE Stream (i.e., the PID value of the TS Logical

Channel that carries a ULE Stream). The conditions under which this information is required and the format in which it is to be provided are beyond the scope of this document. Addressing and mapping issues for ULE over MPEG-2 are also described in [IPDVB-AR].

2. Conventions Used in This Document

The capitalized key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Other terms used in this document are defined below:

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-MPEG2].

AFC: Adaptation Field Control [ISO-MPEG2]. 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.

b: bit. For example, one byte consists of 8b.

B: Byte. Groups of bytes are represented in Internet byte order.

DSM-CC: Digital Storage Media Command and Control [ISO-DSMCC]. A format for transmission of data and control information in an MPEG-2 Private Section, defined by the ISO MPEG-2 standard.

DVB: Digital Video Broadcast. A framework and set of associated standards published by the European Telecommunications Standards Institute (ETSI) (e.g., [ETSI-DVBC, ETSI-DVBS, ETSI-DVBT]) for the transmission of video, audio, and data using the ISO MPEG-2 Standard [ISO-MPEG2].

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.

End Indicator: A value that indicates to the Receiver that there are no further SNDUs present within the current TS Packet.

LLC: Logical Link Control [ISO-8802-2, IEEE-802.2]. A link-layer protocol defined by the IEEE 802 standard, which follows the Ethernet MAC Header.

MAC: Medium Access Control [IEEE-802.3]. A link-layer protocol defined by the IEEE 802.3 standard (or by Ethernet v2 [DIX]).

MAC Header: The link-layer header of the IEEE 802.3 standard [IEEE-802.3] or Ethernet v2 [DIX]. It consists of a 6B destination address, 6B source address, and 2B Type field (see also NPA, LLC).

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/IEC 13818-1) [ISO-MPEG2], and ITU-T (in H.222 [ITU-H222]).

Next-Header: A Type value indicating an Extension Header.

NPA: Network Point of Attachment. In this document, refers to a 6-byte destination address (resembling an IEEE MAC address) within the MPEG-2 transmission network that is used to identify individual Receivers or groups of Receivers.

Packing Threshold: A period of time an Encapsulator is willing to defer transmission of a partially filled TS-Packet to accumulate more SNDUs, rather than use Padding. After the Packet Threshold period, the Encapsulator uses Padding to send the partially filled TS-Packet.

Padding: A method that fills the remaining unused bytes in a TS Packet payload using the specific pattern of 0xFF.

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

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

PES: Packetized Elementary Steam [ISO-MPEG2]. A format of MPEG-2 TS packet payload usually used for video or audio information.

PID: Packet Identifier [ISO-MPEG2]. A 13-bit field carried in the header of TS Packets. This is used to identify the TS Logical Channel to which a TS Packet belongs [ISO-MPEG2]. The TS Packets

forming the parts of a Table Section, PES, or other Payload Unit must all carry the same PID value. The all-zeros PID 0x0000 as well as other PID values are reserved for specific PSI/SI Tables [ISO-MPEG2]. The all-ones PID value 0x1FFF indicates a Null TS Packet introduced to maintain a constant bit rate of a TS Multiplex. There is no required relationship between the PID values used for TS Logical Channels transmitted using different TS Multiplexes.

PP: Payload Pointer [ISO-MPEG2]. An optional one-byte pointer that directly follows the 4-byte TS Packet header. It contains the number of bytes that follow the Payload Pointer, up to the start of the first Payload Unit (counted from the first byte of the TS Packet payload field, and excluding the PP field itself). 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, Table Sections, and ULE. It is not present in TS Logical Channels that use the PES-format.

Private Section: A syntactic structure constructed in accordance with Table 2-30 of [ISO-MPEG2]. The structure may be used to identify private information (i.e., not defined by [ISO-MPEG2]) relating to one or more elementary streams, or a specific MPEG-2 program, or the entire Transport Stream. Other Standards bodies, e.g., ETSI, ATSC, have defined sets of table structures using the private_section structure. A Private Section is transmitted as a sequence of TS Packets using a TS Logical Channel. A TS Logical Channel may carry sections from more than one set of tables.

PSI: Program Specific Information [ISO-MPEG2]. Tables used to convey information about the service carried in a TS Multiplex. The information is carried in one of four specifically identified Table Sections defined by MPEG-2 [ISO-MPEG2]. See also SI Table.

PU: Payload Unit.

PUSI: Payload_Unit_Start_Indicator [ISO-MPEG2]. 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).

Receiver: Equipment that processes the signal from a TS Multiplex and performs filtering and forwarding of encapsulated PDUs to the network-layer service (or bridging module when operating at the link layer).

SI Table: Service Information Table [ISO-MPEG2]. In this document, this term describes a table that is defined by another standards body to convey information about the services carried in a TS Multiplex. A Table may consist of one or more Table Sections; however, all sections of a particular SI Table must be carried over a single TS Logical Channel [ISO-MPEG2].

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

Table Section: A Payload Unit carrying all or part of an SI or PSI Table [ISO-MPEG2].

TS: Transport Stream [ISO-MPEG2], a method of transmission at the MPEG-2 level using TS Packets; it represents layer 2 of the ISO/OSI reference model. See also TS Logical Channel and TS Multiplex.

TS Header: The 4-byte header of a TS Packet [ISO-MPEG2]. Each 188B TS Packet incorporates a 4B header with the following fields (those referenced within this document are marked with *):

Field Length (in bits)	Name/Purpose
8b	Synchronisation pattern equal to 0x47
*1b	Transport Error Indicator
*1b	Payload Unit Start Indicator (PUSI)
1b	Transport Priority
*13b	Packet IDentifier (PID)
2b	Transport Scrambling Control
*2b	Adaptation Field Control (AFC)
*4b	Continuity Counter (CC)

If the PUSI bit is set to a value of 1, there is one additional field following the TS packet header:

*8b	Payload Pointer (PP)

TS Logical Channel: Transport Stream Logical Channel. In this document, this term identifies a channel at the MPEG-2 level [ISO-MPEG2]. It exists at level 2 of the ISO/OSI reference model. All packets sent over a TS Logical Channel carry the same PID value (this value is unique within a specific TS Multiplex). The term "Stream" is defined in MPEG-2 [ISO-MPEG2] to describe the content carried by a specific TS Logical Channel (see ULE Stream). Some PID values are reserved (by MPEG-2) for specific signalling. Other standards (e.g., ATSC, DVB) also reserve specific PID values.

TS Multiplex: In this document, this term defines a set of MPEG-2 TS Logical Channels sent over a single lower-layer connection. This may be a common physical link (i.e., a transmission at a specified symbol rate, FEC setting, and transmission frequency) or an encapsulation provided by another protocol layer (e.g., Ethernet, or RTP over IP). The same TS Logical Channel may be repeated over more than one TS Multiplex (possibly associated with a different PID value) [RFC4259]; 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-MPEG2]. 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 TS Logical Channels.

ULE Stream: An MPEG-2 TS Logical Channel that carries only ULE encapsulated PDUs. ULE Streams may be identified by definition of a stream_type in SI/PSI [ISO-MPEG2].

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 either by being placed in the payload of a single TS Packet, or, if required, by being 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), a practice sometimes known as Packing. All TS Packets comprising an 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. A PUSI field with a value of 1 indicates the start of at least one Payload Unit (SNDU) within the TS Packet payload. The semantics of the PUSI bit are defined for PES and PSI packets [ISO-MPEG2]; for private data, its use is not defined in the MPEG-2 Standard. Although ULE uses private data, the operation follows that of PSI packets. Hence, the following PUSI values are defined:

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

1: The TS Packet contains the start of an 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 intended to start another Payload Unit, a stuffing procedure (known as Padding) fills the remainder of the TS Packet payload with bytes with a value 0xFF [ISO-MPEG2].

A Receiver processing MPEG-2 Table Sections that receives a value of 0xFF in the first byte of a Table Section (table_Id) interprets this as Padding/Stuffing 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. This adaptation field may extend the TS Packet Header to carry timing and synchronisation information and may also be used to include stuffing bytes before a TS Packet payload. 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'. For TS Logical Channels supporting ULE, Receivers MUST discard TS Packets that carry other AFC values.

4. SNDU Format

PDUs are encapsulated using ULE to form an SNDU. (Each SNDU is an MPEG-2 Payload Unit.) The encapsulation format to be used for PDUs is illustrated below:

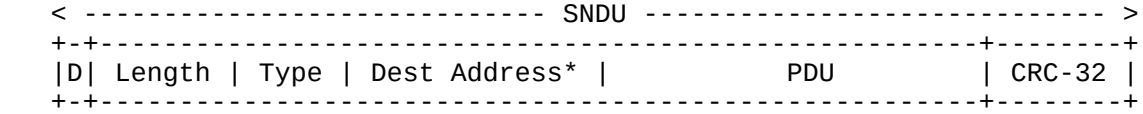


Figure 1: SNDU Encapsulation (* optional Destination Address)

All multi-byte values in ULE (including the Length/End Indicator (4.2,4.3), Type (4.4), Destination Address (4.5), and Extension Headers (5)) are transmitted in network byte order (most significant byte first). The most significant bit of each byte is placed in the left-most position of the 8-bit field. Appendix A provides informative examples of usage.

4.1. Destination Address Absent (D) Field

The most significant bit of the Length field carries the value of the Destination Address Absent 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.

An End Indicator (4.3) MUST be sent with a D-bit value of 1. Other SNDUs MAY be sent with a D-bit value of 0 or 1. The default method SHOULD use a D-bit value of 0 (see section 4.5).

4.2. Length Field

A 15-bit value that indicates the length, in bytes, of the SNDU counted from the byte following the Type field of the SNDU base header (figure 9) up to and including the CRC. This Length includes the size of any extension headers that may be present (section 5). Note the special case described in section 4.3.

4.3. End Indicator

When the first two bytes following an SNDU have the value 0xFFFF, this denotes an End Indicator (i.e., all ones length combined with a D-bit value of 1). This indicates to the Receiver that there are no further SNDUs 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

The 16-bit Type field indicates the type of payload carried in an SNDU, or the presence of a Next-Header. 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 Ethernet v2 Specification [DIX]. After specification of IEEE 802.3 [IEEE-802.3, ISO-8802-2], the set of EtherTypes less than 1536 (0x0600) assumed the role of a length indicator. Ethernet receivers use this feature to discriminate LLC format frames. Hence, any IEEE EtherType < 1536 indicates an LLC frame, and the actual value indicates the length of the LLC frame.

There is a potential ambiguous case 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 5.2).

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 less than 1536 in decimal. These Type field values are IANA assigned (see section 4.4.1) and indicate the Next-Header.

The second set of ULE Type field values comprise the set of values greater than or equal to 1536 in decimal. In ULE, this 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: Next-Header Type Fields

The first part of the Type space corresponds to the values 0 to 1535 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 in this document:

0x0000: Test SNDU (see section 5.1)
0x0001: Bridged Frame (see section 5.2)
0x0100: Extension-Padding (see section 5.3)

The remaining values within the first part of the Type space are reserved for Next-Header values allocated by the IANA.

4.4.2. Type 2: EtherType Compatible Type Fields

The second part of the Type space corresponds to the values between 0x600 (1536 decimal) and 0xFFFF. This set of type assignments follows DIX/IEEE assignments (but excludes use of this field as a frame length indicator). All assignments in this space MUST use the values defined for IANA EtherType. The following two Type values are used as examples (taken from the IANA EtherTypes registry):

0x0800: IPv4 Payload (see section 4.7.2)
0x86DD: IPv6 Payload (see section 4.7.3)

4.5. SNDU Destination Address Field

The SNDU Destination Address Field is optional (see section 4.1). This field MUST be carried (i.e., D=0) for IP unicast packets destined to routers that are sent using shared links (i.e., where the same link connects multiple Receivers). 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, or a bridged MAC 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 an SNDU Destination Address field (i.e., D=0), a Network Point of Attachment (NPA) field directly follows the fourth byte of the SNDU header. NPA destination addresses are 6 Byte 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 an 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 that this SNDU is to be delivered to all Receivers.

IPv4 packets carrying an IPv4 subnetwork broadcast address need to be delivered to all systems with the same network prefix. When a SNDU Destination Address is present (D=0), the value MUST be set to the NPA link broadcast address (0xFF:FF:FF:FF:FF:FF).

When the PDU is an IP multicast packet and an SNDU Destination Address is present (D=0), the IP group destination address of the multicast packet MUST be mapped to the multicast SNDU Destination Address (following the method used to generate a destination MAC address in Ethernet). The method for mapping IPv4 multicast addresses is specified in [RFC1112]. The method for mapping IPv6 multicast addresses is specified in [RFC2464].

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. Examples where this polynomial is also employed include Ethernet, DSM-CC section syntax [ISO-DSMCC], and AAL5 [ITU-3563]. This is a 32-bit value calculated according to the generator polynomial represented 0x104C11DB7 in hexadecimal:

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

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 holding the CRC-32), and places this in the CRC Field. In ULE, the bytes are processed in order of increasing position within the SNDU; the order of processing bits is NOT reversed. This use resembles, but is different from that in SCTP [RFC3309].

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. Appendix B provides an example of an Encapsulated PDU that includes the computed CRC-32 value.

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 Encapsulator and/or the Receiver. It may also detect the presence of uncorrected errors from the physical link (however, these may also be detected by other means, e.g., section 7.3).

4.7. Description of SNDU Formats

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

The following SNDU Formats are defined here:

End Indicator: The Receiver should enter the Idle State (4.7.1).
IPv4 SNDU: The payload is a complete IPv4 datagram (4.7.2).
IPv6 SNDU: The payload is a complete IPv6 datagram (4.7.3).
Test SNDU: The payload will be discarded by the Receiver (5.1).
Bridged SNDU: The payload carries a bridged MAC frame (5.2).

Other formats may be defined through relevant assignments in the IEEE and IANA registries.

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.

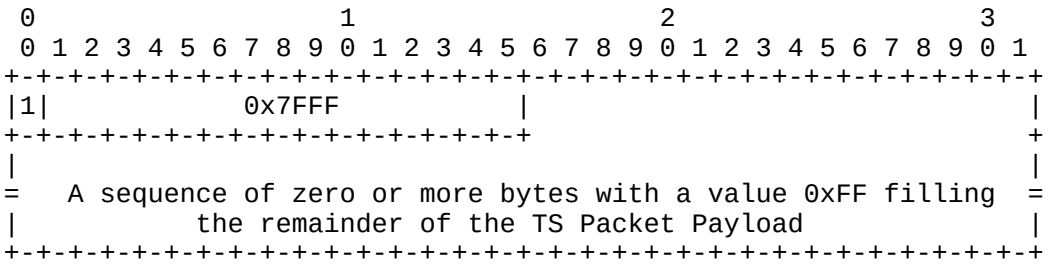


Figure 2: Format for a ULE End Indicator

4.7.2. IPv4 SNDU Encapsulation

IPv4 datagrams are directly 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).

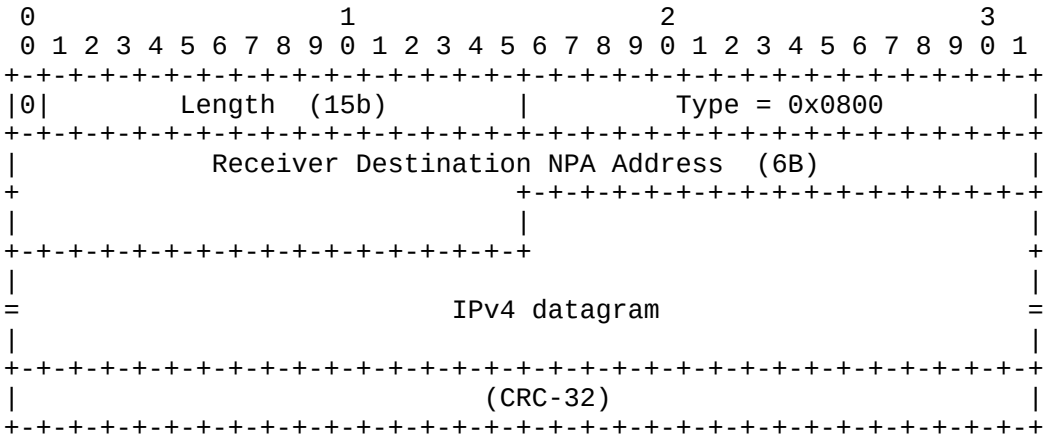


Figure 3: SNDU Format for an IPv4 Datagram using L2 filtering (D=0)

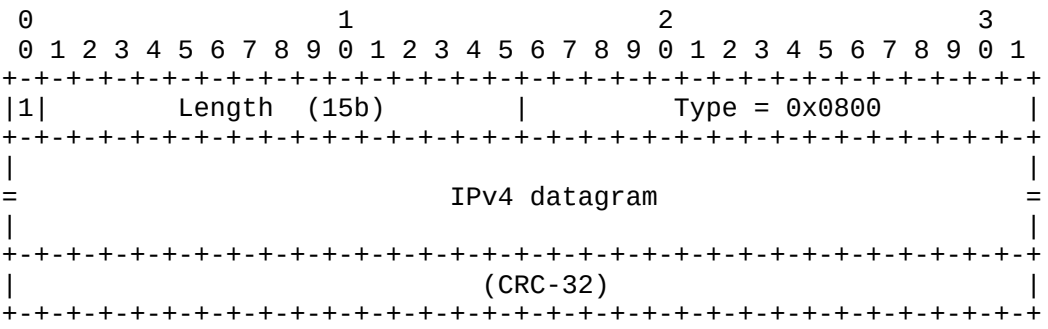


Figure 4: SNDU Format for an IPv4 Datagram using L3 filtering (D=1)

4.7.3. IPv6 SNDU Encapsulation

IPv6 datagrams are directly 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.

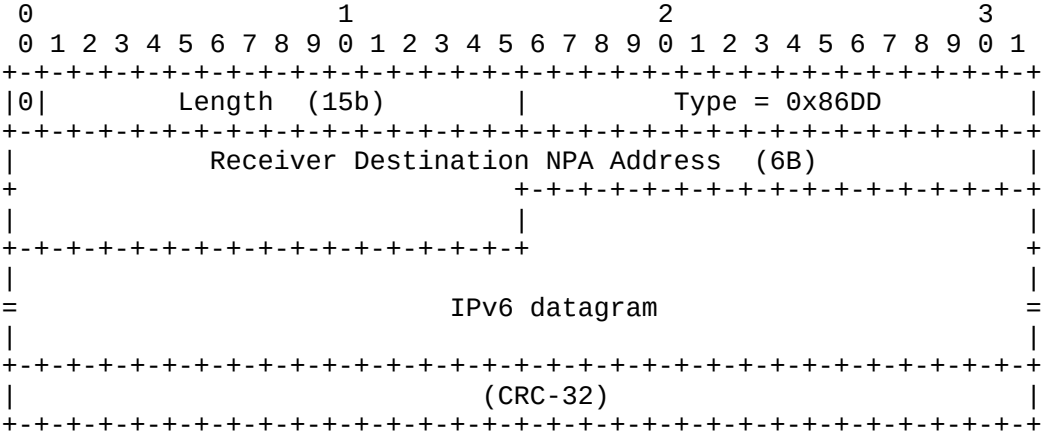


Figure 5: SNDU Format for an IPv6 Datagram using L2 filtering (D=0)

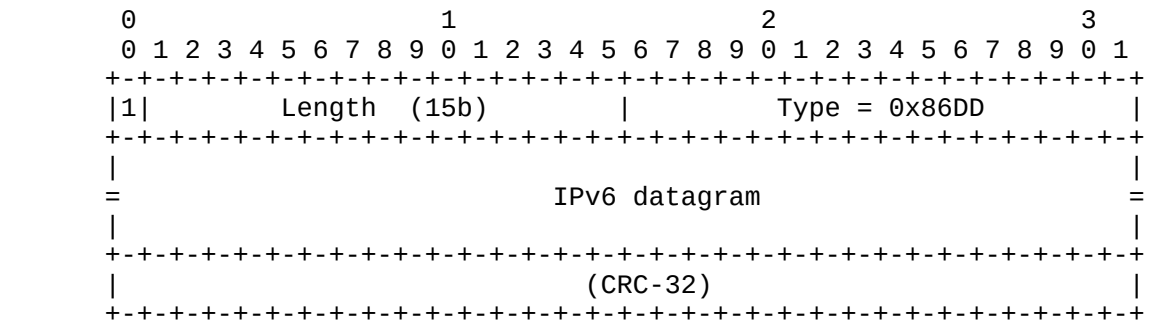


Figure 6: SNDU Format for an IPv6 Datagram using L3 filtering (D=1)

5. Extension Headers

This section describes an extension format for the ULE encapsulation. In ULE, a Type field value less than 1536 decimal indicates an Extension Header. These values are assigned from a separate IANA registry defined for ULE.

The use of a single Type/Next-Header field simplifies processing and eliminates the need to maintain multiple IANA registries. The cost is that each Extension Header requires at least 2 bytes. This is justified, on the basis of simplified processing and maintaining a simple lightweight header for the common case when no extensions are present.

A ULE Extension Header is identified by a 16-bit value in the Type field. This field is organised as a 5-bit zero prefix, a 3-bit H-LEN field, and an 8-bit H-Type field, as follows:

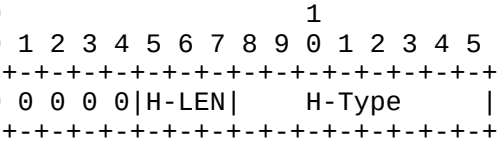


Figure 7: Structure of ULE Next-Header Field

The H-LEN Assignment is described below:

0 Indicates a Mandatory Extension Header
1 Indicates an Optional Extension Header of length 2B (Type only)
2 Indicates an Optional Extension Header of length 4B (Type + 2B)
3 Indicates an Optional Extension Header of length 6B (Type + 4B)
4 Indicates an Optional Extension Header of length 8B (Type + 6B)
5 Indicates an Optional Extension Header of length 10B (Type + 8B)

>=6 The combined H-LEN and H-TYPE values indicate the EtherType of a PDU that directly follows this Type field.

The H-LEN value indicates the total number of bytes in an Optional Extension Header (including the 2B Type field).

An H-LEN value of zero indicates a Mandatory Extension Header. Each Mandatory Extension Header has a pre-defined length that is not communicated in the H-LEN field. No additional limit is placed on the maximum length of a Mandatory Extension Header. A Mandatory Extension Header MAY modify the format or encoding of the enclosed PDU (e.g., to perform encryption and/or compression).

The H-Type is a one-byte field that is either one of 256 Mandatory Header Extensions or one of 256 Optional Header Extensions. The set of currently permitted values for both types of Extension Headers are defined by an IANA Registry (section 15). Registry values for Optional Extensions are specified in the form H=1 (i.e., a decimal number in the range 256-511), but may be used with an H-Length value in the range 1-5 (see example in section 5.3).

Two examples of Extension Headers are the Test SNDU and the use of Extension-Padding. The Test SNDU Mandatory Extension Header results in the entire PDU's being discarded. The Extension-Padding Optional Extension Header results in the following (if any) option header being ignored (i.e., a total of H-LEN 16-bit words).

The general format for an SNDU with Extension Headers is:

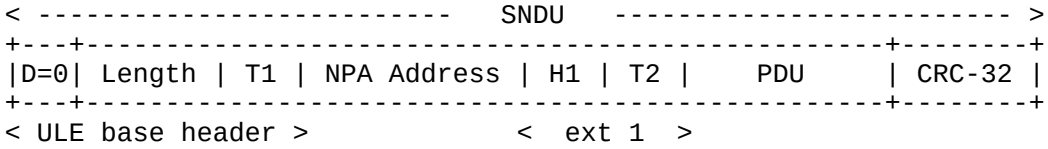


Figure 8: SNDU Encapsulation with one Extension Header (for D=0)

Where:
D is the ULE D_bit (in this example D=0; however, NPA addresses may also be omitted when using Extension Headers).
T1 is the base header Type field. In this case, specifying a Next-Header value.
H1 is a set of fields defined for header type T1. There may be 0 or more bytes of information for a specific ULE Extension Header.
T2 is the Type field of the next header, or an EtherType > 1535 B indicating the type of the PDU being carried.

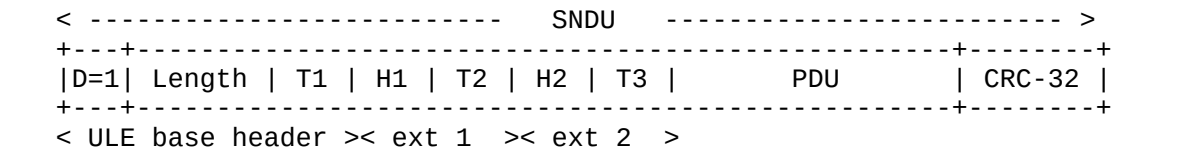


Figure 9: SNDU Encapsulation with two Extension Headers (D=1)

Using this method, several Extension Headers MAY be chained in series. Figure 12 shows an SNDU including two Extension Headers. In the example, the values of T1 and T2 are both less than 1536 decimal. Each indicates the presence of an Extension Header, rather than a directly following PDU. T3 has a value > 1535 indicating the EtherType of the PDU being carried. Although an SNDU may contain an arbitrary number of consecutive Extension Headers, it is not expected that SNDUs will generally carry a large number of extensions.

5.1. Test SNDU

A Test SNDU (Figure 10) is a Mandatory Extension Header of Type 1. This header must be the final (or only) extension header specified in the header chain of an SNDU. The structure of the Data portion of this SNDU is not defined by this document. Receivers MAY record reception in a log file, but MUST then discard any Test SNDUs. The D-bit MAY be set in a TEST SNDU.

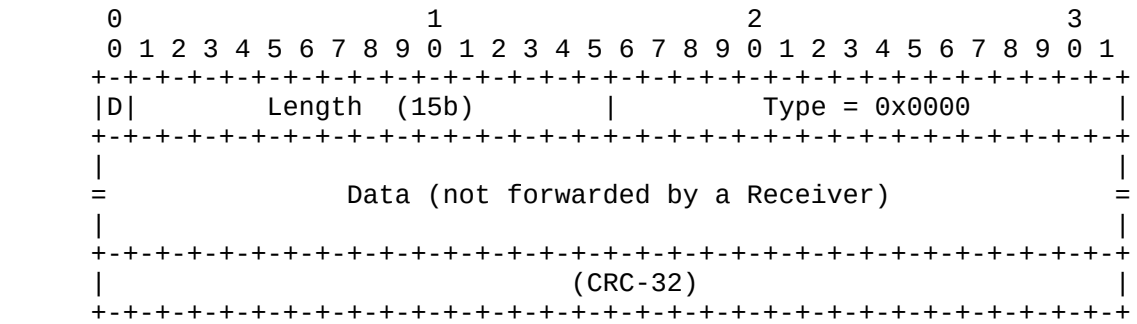


Figure 10: SNDU Format for a Test SNDU

5.2. Bridged Frame SNDU Encapsulation

A bridged SNDU is a Mandatory Extension Header of Type 1. It MUST be the final (or only) extension header specified in the header chain of an SNDU. The payload includes MAC address and EtherType [DIX] or LLC Length [ISO-8802-2] fields together with the contents of a bridged MAC frame. The SNDU has the format shown in Figures 11 and 12.

When an NPA address is specified (D=0), Receivers MUST discard all SNDUs that carry an NPA destination address that does NOT match their own NPA address (or a broadcast/multicast address); the payload of the remaining SNDUs are processed by the bridging rules that follow. An SNDU without an NPA address (D=1) results in a Receiver performing bridging processing on the payload of all received SNDUs.

An Encapsulator MAY also use this encapsulation format to directly communicate network protocol packets that require the LLC encapsulation [IEEE-802.2, ISO-8802-2]. To do this, it constructs an SNDU with a Bridge Extension Header containing the intended destination MAC address, the MAC source address of the Encapsulator, and the LLC-Length. The PDU comprises an LLC header followed by the required payload. The Encapsulator MAY choose to suppress the NPA address (see 4.5).

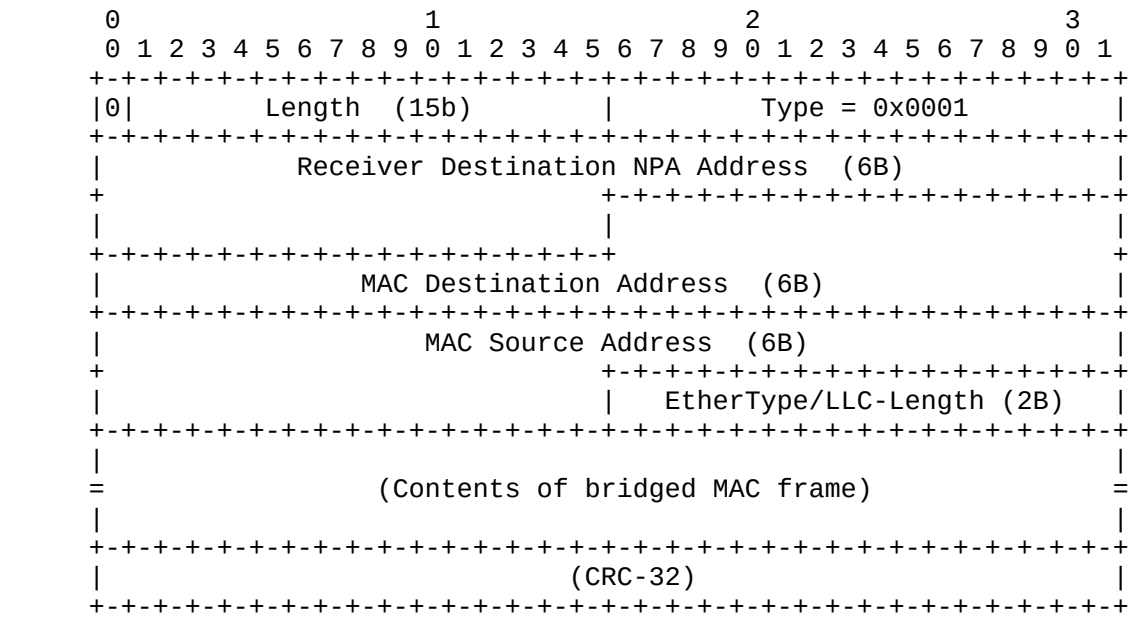


Figure 11: SNDU Format for a Bridged Payload (D=0)

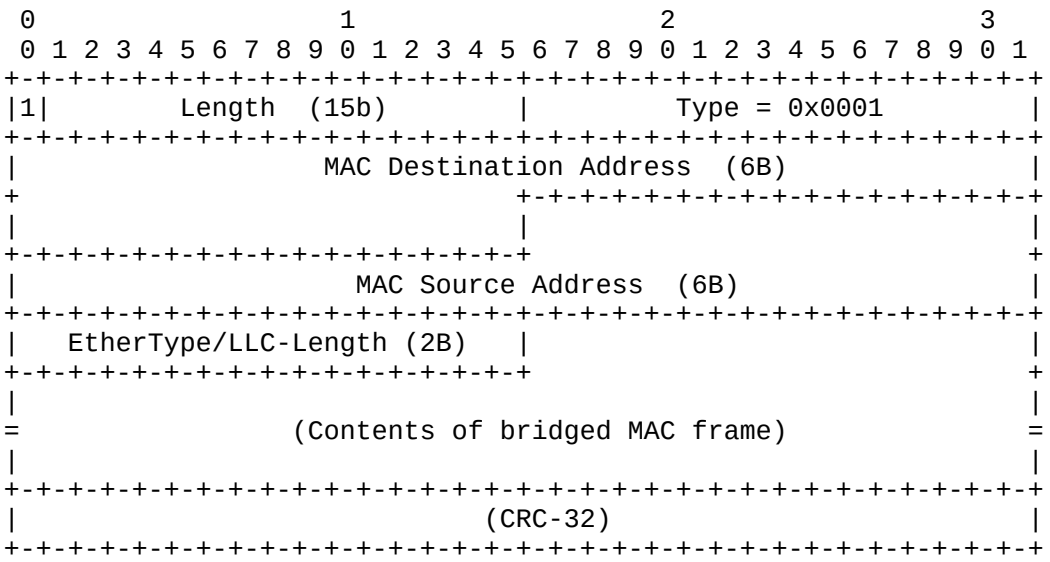


Figure 12: SNDU Format for a Bridged Payload (D=1)

The EtherType/LLC-Length field of a frame is defined according to IEEE 802.3 [IEEE-802.2] (see section 5).

In this special case, the Mandatory Extension Header format may be interpreted as either an EtherType [DIX] or an LLC Length field, specified by IEEE 802 [IEEE-802.3] rather than as a value assigned in the ULE Next-Header Registry maintained by the IANA.

The MAC addresses in the frame being bridged SHOULD be assigned according to the rules specified by the IEEE and 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 from the NPA addresses carried in the SNDU header.

A frame Type < 1536 for a bridged frame 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 SHOULD be removed prior to forwarding. This requires the sender to be aware of such Ethernet padding (e.g., [DIX, IEEE-802.3]).

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 [DIX, IEEE-802.3]). 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 calculating/verifying 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).

5.3. Extension-Padding Optional Extension Header

The Extension-Padding Optional Extension Header is specified by an IANA-assigned H-Type value of 0x100. As in other Optional Extensions, the total length of the extension is indicated by the H-LEN field (specified in 16-bit words). The extension field is formed of a group of one to five 16-bit fields.

For this specific option, only the last 16-bit word has an assigned value; the sender SHOULD set the remaining values to 0x0000. The last 16-bit field forms the Next-Header Type field. A Receiver MUST interpret the Type field, but MUST ignore any other fields of this Extension Header.

6. 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 13). 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 [RFC4259].

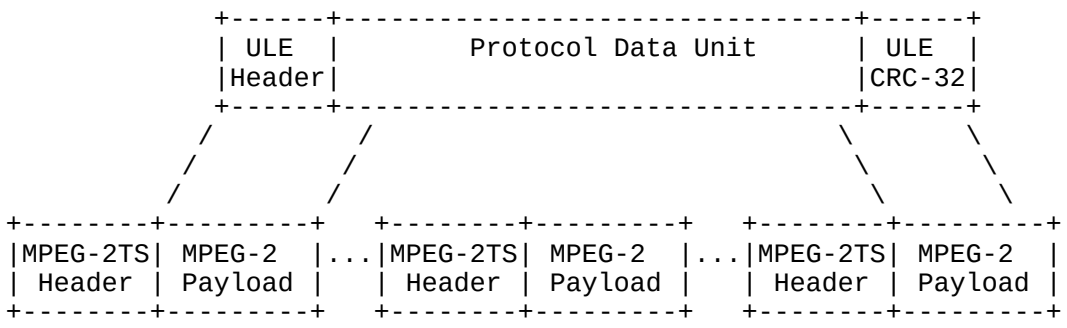


Figure 13: Encapsulation of an SNDU into a series of TS Packets

6.1. 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 an 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 that the SNDU starts immediately after the Payload Pointer in the TS Packet payload.

The Encapsulation MUST ensure that all TS Packets set the MPEG-2 Continuity Counter carried in the TS Packet header, according to [ISO-MPEG2]. This value MUST be incremented by one (modulo 16) for each successive TS Packet containing a fragment/complete SNDU sent using the same TS Logical Channel.

An Encapsulator MAY decide not to send another SNDU immediately, even if space is available in a partially filled TS Packet. This procedure is known as Padding (Figure 14). The End Indicator 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.

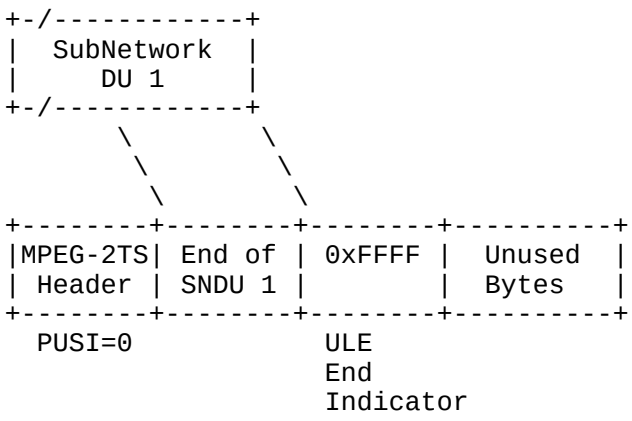


Figure 14: A TS Packet carrying the end of SNDU 1, followed by an End Indicator

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 6.2). This is called Packing (Figure 15).

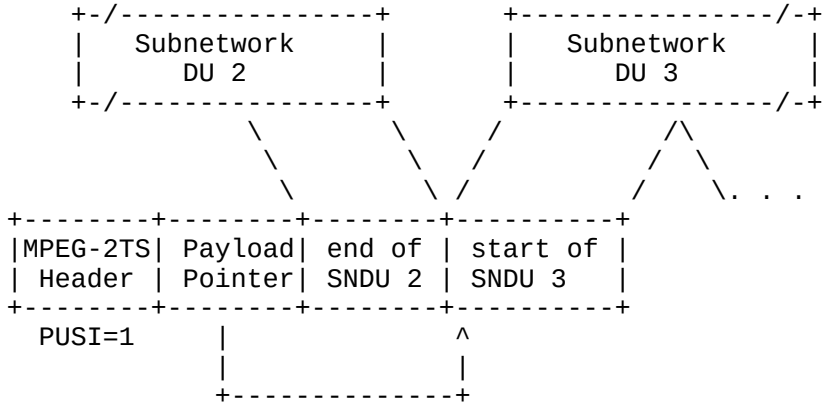


Figure 15: A TS Packet with the end of SNDU 2, followed by SNDU 3

6.2. 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 [ISO-MPEG2] require that the header of this new TS Packet carry a PUSI value of 1 followed by a Payload Pointer value of 0x00.)

(ii) If the TS Packet carrying the final part of an 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 followed by a Payload Pointer value of 0x00.)

(iii) If the TS Packet carrying the final part of an SNDU has exactly two bytes of unused payload, and the PUSI was NOT already set, the Encapsulator MUST place the value 0xFFFF in these final two bytes, providing an End Indicator (section 4.3), 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 followed by 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 then start transmission of the next SNDU in a new TS Packet. (The standard rules [ISO-MPEG2] require that the header of this new TS Packet carry a PUSI value of 1 and a Payload Pointer value of 0x00.)

(v) If at least two bytes are available for SNDU 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. When the Encapsulator packs further SNDUs into a TS Packet where the PUSI has

NOT already been set, the PUSI MUST be updated (set to 1), and an 8-bit Payload Pointer MUST be inserted in the first byte directly following the TS Packet header. (This reduces the size of the TS Packet payload field that is available for data by one byte.) The value of the Payload Pointer 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, known as the Packing Threshold, MUST be bounded and SHOULD be configurable in the Encapsulator. If sufficient additional PDUs are NOT received to complete the TS Packet within the Packing Threshold, the Encapsulator MUST insert an End Indicator (using rule iv).

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

When an 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 (using rule iv).

7. Receiver Processing

A Receiver tunes to a specific TS Multiplex carrying a ULE Stream 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 MUST be 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 181, 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).

7.1. Idle State

After initialisation or errors, 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, upon which it then enters the Reassembly State. Figure 16 outlines these state transitions:

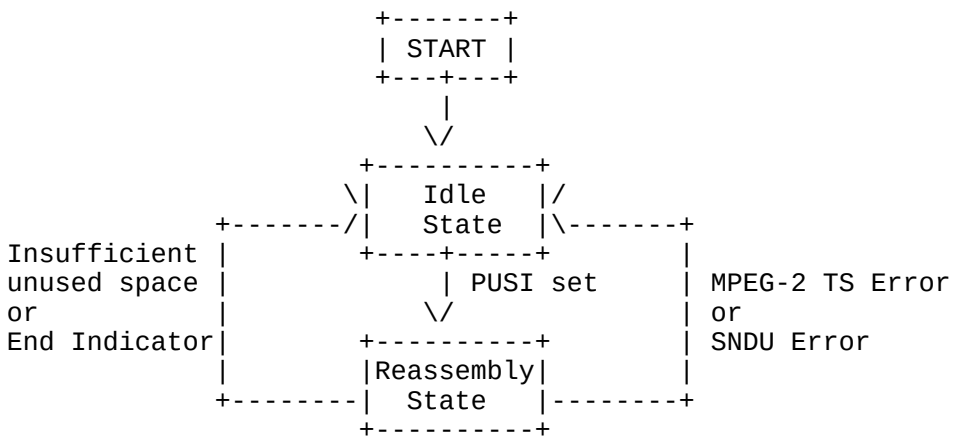


Figure 16: Receiver state transitions

7.1.1. 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. Following a loss of synchronisation, values between 0 and 181 are permitted, in which case the Receiver MUST discard the number of bytes indicated by the Payload Pointer (counted from the first byte of the TS Packet payload field, and excluding the PP field itself), before leaving the Idle State. It then enters the Reassembly State, and starts reassembly of a new SNDU at this point.

7.2. 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 and SHOULD be recorded as an SNDU length error.

If the Length of the Current SNDU is greater than 4, the Receiver 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 (see also 7.2.1). When the Current SNDU length equals the value of the Length field, the Receiver MUST calculate and verify the CRC value (see 4.6). SNDUs that contain an invalid CRC value MUST be discarded. Mismatch of the CRC is an error event and SHOULD be recorded as a CRC error. The underlying physical-layer processing (e.g., forward error correction coding) often results in patterns of errors, rather than single bit errors, so the Receiver needs to be robust to arbitrary patterns of corruption to the TS Packet and payload, including potential corruption of the PUSI, PP, and SNDU Length fields. Therefore, a Receiver SHOULD discard the remaining TS Packet payload (if any) following a CRC mismatch and return to the Idle State.

When the Destination Address is present (D=0), 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, 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 Extension Headers). The SNDU payload is then passed to the next protocol layer specified. An SNDU with an unknown Type value < 1536 MUST be discarded. This error event SHOULD be recorded as an 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 whether 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. (This is provided that the TS Packet has a

PUSI value of 1, see 7.2.1; otherwise, the Receiver has detected a delimiting error and MUST discard all remaining bytes in the TS Packet payload and transitions to the Idle State.)

7.2.1. Reassembly Payload Pointer Checking

A Receiver that has partially received an SNDU (in the Current SNDU buffer) MUST check the PUSI value in the header of all subsequent TS Packets with the same PID (i.e., same TS Logical Channel). 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.

7.3. Other Error Conditions

The Receiver SHOULD check the MPEG-2 Transport Error Indicator carried in the TS Packet header [ISO-MPEG2]. 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 MUST check the MPEG-2 Continuity Counter carried in the TS Packet header [ISO-MPEG2]. If two (or more) successive TS Packets within the same TS Logical Channel carry the same Continuity Counter value, the duplicate TS Packets MUST be silently discarded. If the received value is NOT identical to that in the previous TS Packet, and it 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.

Note that an MPEG2-2 Transmission network is permitted to carry duplicate TS Packets [ISO-MPEG2], which are normally detected by the MPEG-2 Continuity Counter. A Receiver that does not perform the above Continuity Counter check would accept duplicate copies of TS Packets to the reassembly procedure. In most cases, the SNDU CRC-32 integrity check will result in discard of these SNDUs, leading to unexpected PDU loss; however, in some cases, duplicate PDUs (fitting into one TS Packet) could pass undetected to the next layer protocol.

8. Summary

This document defines a Unidirectional Lightweight Encapsulation (ULE) that performs 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.

ULE also provides an Extension Header format and defines an associated IANA registry for efficient and flexible support of both mandatory and optional SNDU headers. This allows for future extension of the protocol, while providing backwards compatibility with existing implementations. In particular, Optional Extension Headers may safely be ignored by Receivers that do not implement them, or choose not to process them.

9. Acknowledgements

This document is based on a previous document 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; in particular, the many comments received from Art Allison, Carstsen Borman, Patrick Cipiere, Wolgang Fritsche, Hilmar Linder, Alain Ritoux, and William Stanislaus. Alain also provided the original examples of usage.

10. Security Considerations

The security considerations for ULE resemble those that arise when the existing Multi-Protocol Encapsulation (MPE) is used. ULE does not add specific new threats that will impact the security of the general Internet.

There is a known security issue with un-initialised stuffing bytes. In ULE, these bytes are set to 0xFF (normal practice in MPEG-2).

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 direct encapsulation of IPv4/IPv6 in ULE, this is avoided by including only one SNDU Length

Field. However, this issue still arises in bridged LLC frames, and frames with a LLC Length greater than the SNDU payload size MUST be discarded, and an SNDU payload length error SHOULD be recorded.

In the future, a ULE Mandatory Extension Header may be used to define a method to perform link encryption of the SNDU payload. This is as an additional security mechanism to IP-, transport-, or application-layer security, not a replacement [RFC4259]. The approach is generic and decouples the encapsulation from future security extensions. The operation provides functions that resemble those currently used with the MPE encapsulation.

Additional security control fields may be provided as part of this link encryption Extension Header, e.g., to associate an SNDU with one of a set of Security Association (SA) parameters. As a part of the encryption process, it may also be desirable to authenticate some or all of the SNDU headers. The method of encryption and the way in which keys are exchanged is beyond the scope of this specification, as are the definition of the SA format and that of the related encryption keys.

11. IANA Considerations

The IANA has created the ULE Next-Header Type field registry as defined in this document.

ULE Next-Header registry

This registry allocates Next-Header values within the range 0-511 (decimal). For each allocated value, it also specifies the set of allowed H-LEN values (see section 5). In combination, these define a set of allowed values in the range 0-1535 for the first part of the ULE Type space (see section 4.4.1).

11.1. IANA Guidelines

The following contains the IANA guidelines for management of the ULE Next-Header registry. This registry allocates values 0-511 decimal (0x0000-0x01FF, hexadecimal). It MUST NOT allocate values greater than 0x01FF (decimal).

It subdivides the Next-Header registry in the following way:

1) 0-255 (decimal) IANA-assigned values, indicating Mandatory Extension Headers (or link-dependent Type fields) for ULE, requiring expert review leading to prior issue of an IETF RFC. This specification MUST define the value and the name associated with the Extension Header, together with the procedure for

processing the Extension Header. It MUST also define the need for the Mandatory Extension and the intended use. The size of the Extension Header MUST be specified.

Assignments have been made in this document, and registered by IANA:

Type	Name	Reference
0:	Test-SNDU	Section 5.1
1:	Bridged-SNDU	Section 5.2

2) 256-511 (decimal) IANA-assigned values, indicating Optional Extension Headers for ULE, requiring expert review leading to prior issue of an IETF RFC. This specification MUST define the value and the name associated with the Extension Header, together with the procedure for processing the Extension Header. The entry MUST specify the range of allowable H-LEN values that are permitted (in the range 1-5). It MUST also define the need for the Optional Extension and the intended use.

Assignments have been made in this document, and registered by IANA:

Type	Name	H-LEN	Reference
256:	Extension-Padding	1-5	Section 5.3

12. References

12.1. Normative References

[ISO-MPEG2] IS 13818-1, "Information technology -- Generic coding of moving pictures and associated audio information -- Part 1: Systems", International Standards Organisation (ISO), 2000.

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

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.

[ULE1] Registration for format_identifier ULE1, SMPTE Registration Authority, LLC, http://www.smpte-ra.org/ule1.html.

12.2. Informative References

[IPDVB-AR] Fairhurst, G. and M-J. Montpetit, "Address Resolution for IP datagrams over MPEG-2 Networks", Work in Progress, September 2005.

[ATSC] A/53, "ATSC Digital Television Standard", Advanced Television Systems Committee (ATSC), Doc. A/53 Rev.C, 2004

[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/65B, "Program and System Information Protocol for Terrestrial Broadcast and Cable", Advanced Television Systems Committee (ATSC), Doc. A/65B, 2003.

[ATSC-REG] ATSC "Code Point Registry" www.atsc.org/standards/Code_Point_Registry.pdf.

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

[DIX] Digital Equipment Corp, Intel Corp, Xerox Corp, "Ethernet Local Area Network Specification" Version 2.0, November 1982.

[ETSI-DAT] EN 301 192, "Specifications for Data Broadcasting", European Telecommunications Standards Institute (ETSI), 2004.

[ETSI-DVBC] EN 300 800, "Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV)", European Telecommunications Standards Institute (ETSI), 1998.

[ETSI-DVBS] EN 300 421, "Digital Video Broadcasting (DVB); Modulation and Coding for DBS satellite systems at 11/12 GHz", European Telecommunications Standards Institute (ETSI), 1997.

[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), 2004.

[ETSI-RCS] ETSI 301 790, "Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems", European Telecommunications Standards Institute (ETSI), 2005.

[IEEE-802.2] IEEE 802.2, "Local and metropolitan area networks- Specific requirements Part 2: Logical Link Control", IEEE Computer Society, (also ISO/IEC 8802-2), 1998.

[IEEE-802.3] IEEE 802.3, "Local and metropolitan area networks- Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", IEEE Computer Society, (also ISO/IEC 8802-3), 2002.

[ISO-DSMCC] IS 13818-6, "Information technology -- Generic coding of moving pictures and associated audio information -- Part 6: Extensions for DSM-CC", International Standards Organisation (ISO), 1998.

[ITU-H222] H.222.0, "Information technology - Generic coding of moving pictures and associated audio information: Systems", International Telecommunication Union, (ITU-T), 1995.

[ITU-3563] I.363.5, "B-ISDN ATM Adaptation Layer specification: Type 5 AAL", International Telecommunication Union, (ITU-T), 1996.

[ISO-8802-2] ISO/IEC 8802.2, "Logical Link Control", International Standards Organisation (ISO), 1998.

[RFC3077] Duros, E., Dabbous, W., Izumiyama, H., Fujii, N., and Y. Zhang, "A Link-Layer Tunneling Mechanism for Unidirectional Links", RFC 3077, March 2001.

[RFC3309] Stone, J., Stewart, R., and D. Otis, "Stream Control Transmission Protocol (SCTP) Checksum Change", RFC 3309, September 2002.

[RFC4259] Montpetit, M.-J., Fairhurst, G., Clausen, H., Collini-Nocker, B., and H. Linder, "A Framework for Transmission of IP Datagrams over MPEG-2 Networks", RFC 4259, November 2005.

[SOOR05] M. Sooriyabandara, G. Fairhurst, A. Ang, B. Collini-Nocker, H. Linder, W. Stering "A Lightweight Encapsulation Protocol for IP over MPEG-2 Networks: Design, Implementation and Analysis", Computer Networks 48 p5-19, 2005.

Appendix A: SNDU Packing Examples

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-MPEG2]. The specification of ULE is provided in the body of this document.

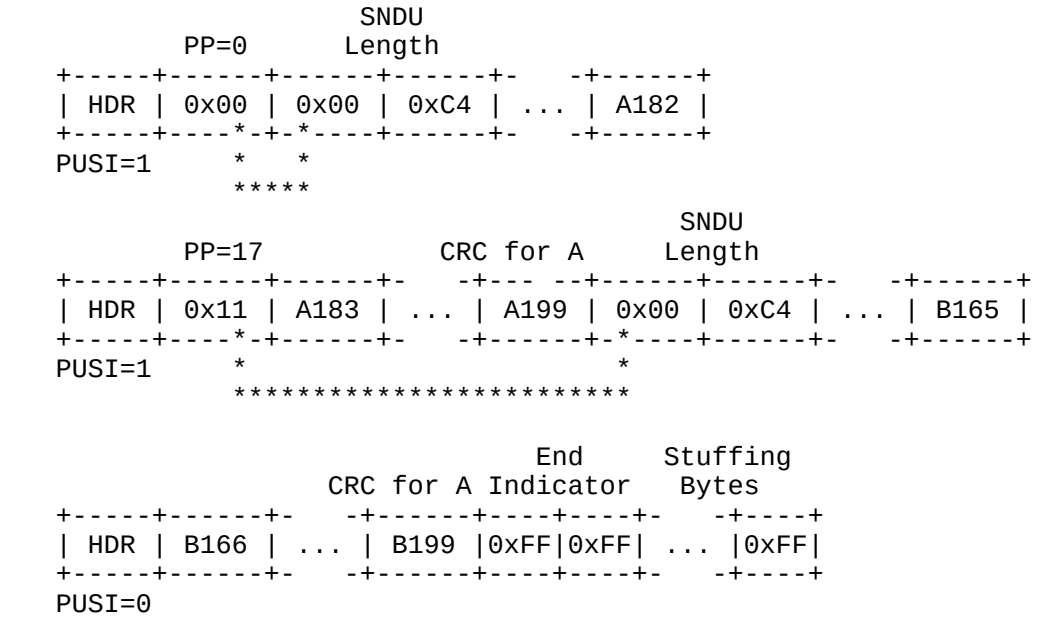
The key below is provided for the following examples.

```
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 the ULE destination NPA address)
SNDU B is 200 bytes (including the ULE destination NPA address)

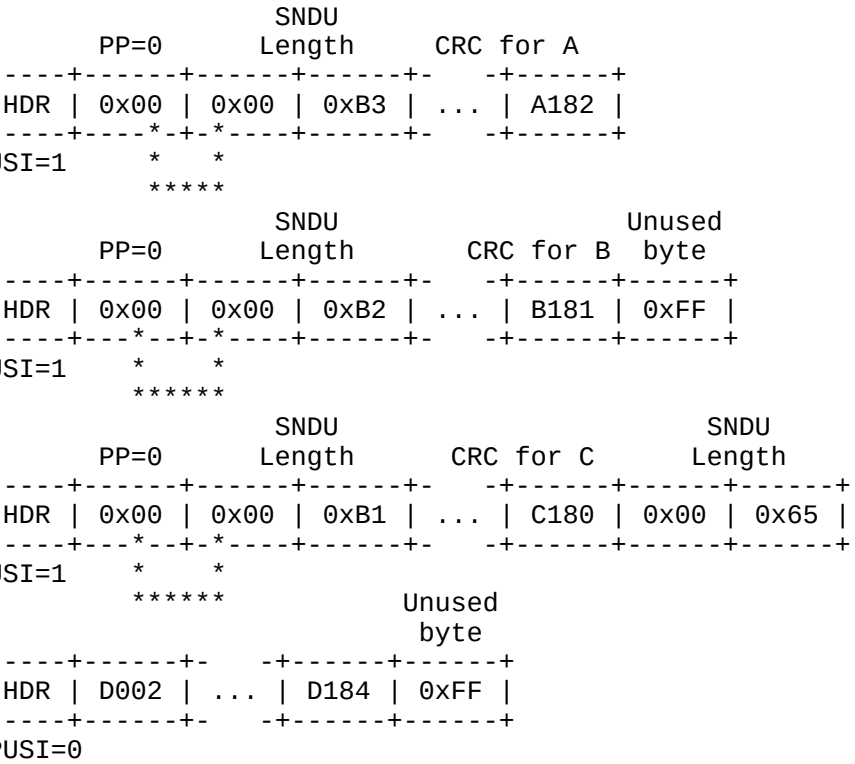
The sequence comprises 3 TS Packets:



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

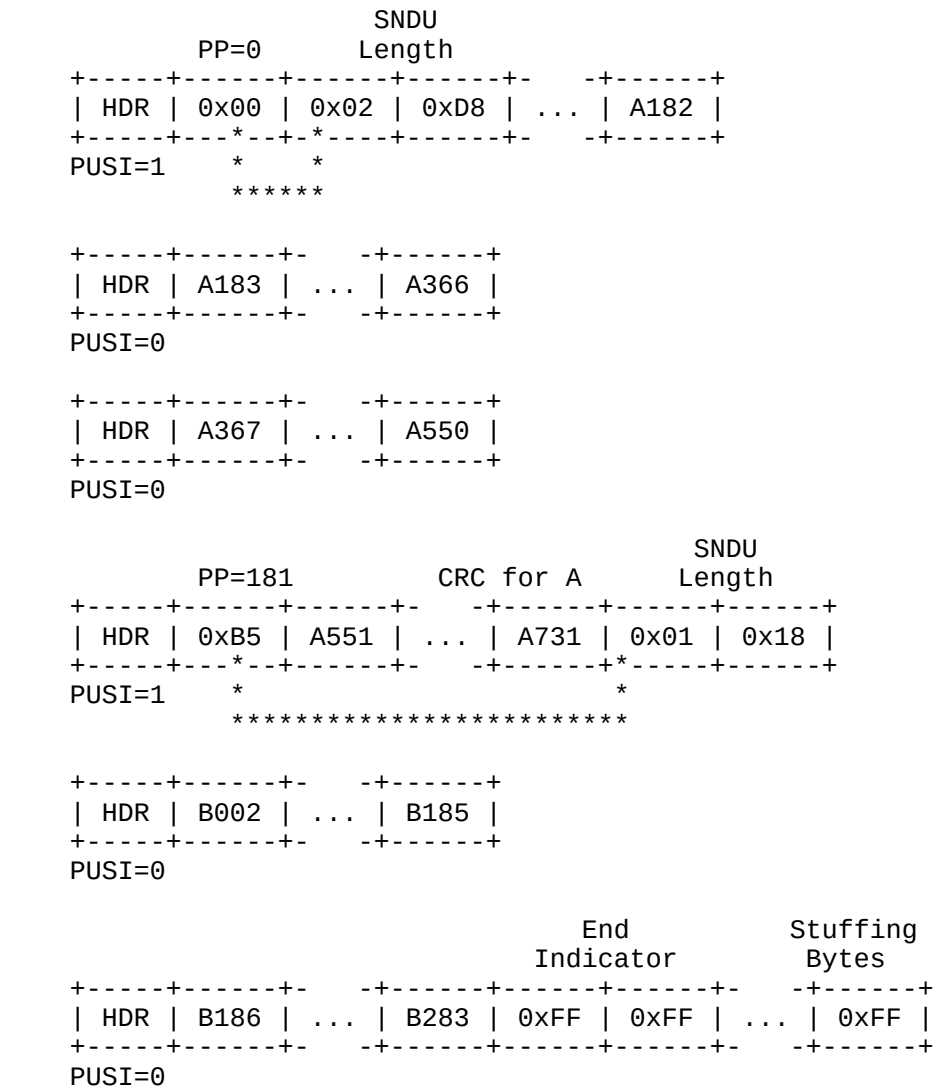
The sequence comprises 4 TS Packets:



Example A.3: Large SNDUs

SNDU A is 732 bytes
SNDU B is 284 bytes

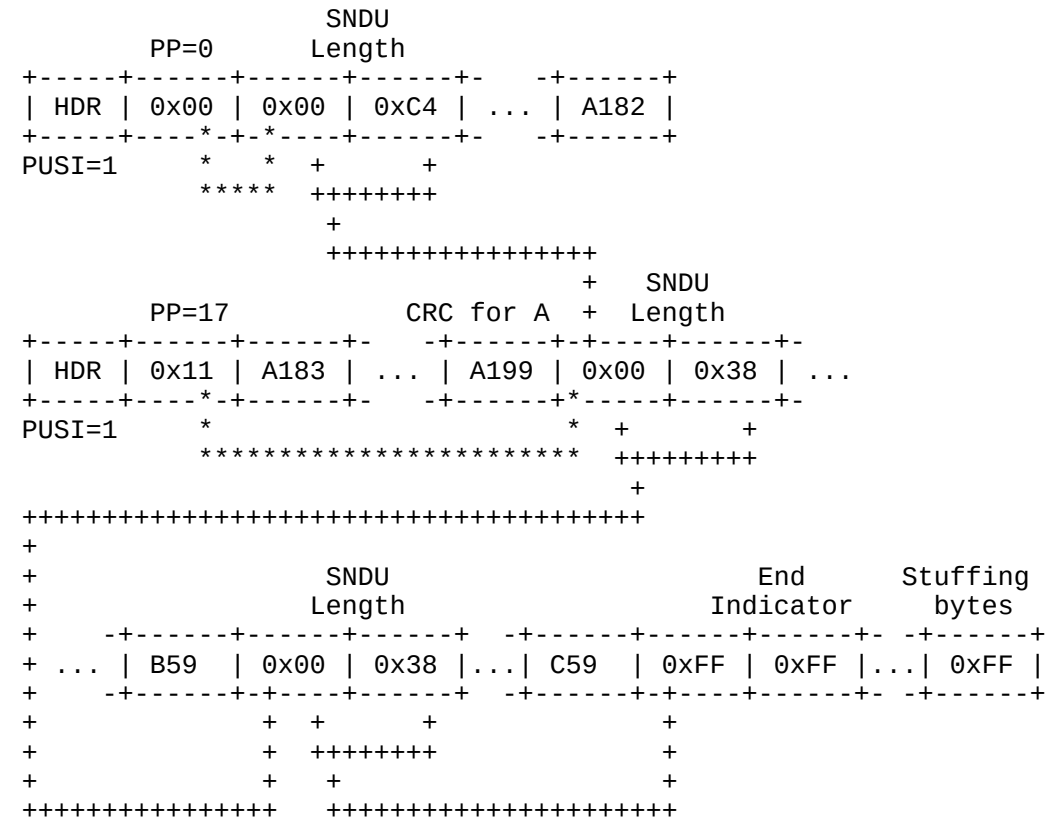
The sequence comprises 6 TS Packets:


SNDU
PP=0 Length
+-----+------+------+------+- -+------+
| HDR | 0x00 | 0x02 | 0xD8 | ... | A182 |
+-----+---*--+-*----+------+- -+------+
PUSI=1 * *
******
+-----+------+- -+------+
| HDR | A183 | ... | A366 |
+-----+------+- -+------+
PUSI=0
+-----+------+- -+------+
| HDR | A367 | ... | A550 |
+-----+------+- -+------+
PUSI=0
SNDU
PP=181 CRC for A Length
+-----+------+------+- -+------+------+------+
| HDR | 0xB5 | A551 | ... | A731 | 0x01 | 0x18 |
+-----+---*--+------+- -+------+*-----+------+
PUSI=1 * *
****
+-----+------+- -+------+
| HDR | B002 | ... | B185 |
+-----+------+- -+------+
PUSI=0
End Stuffing
Indicator Bytes
+-----+------+- -+------+------+------+- -+------+
| HDR | B186 | ... | B283 | 0xFF | 0xFF | ... | 0xFF |
+-----+------+- -+------+------+------+- -+------+
PUSI=0

Example A.4: Illustration of SNDU Length field

SNDU A is 200 bytes
SNDU B is 60 bytes
SNDU C is 60 bytes

The sequence comprises two TS Packets:

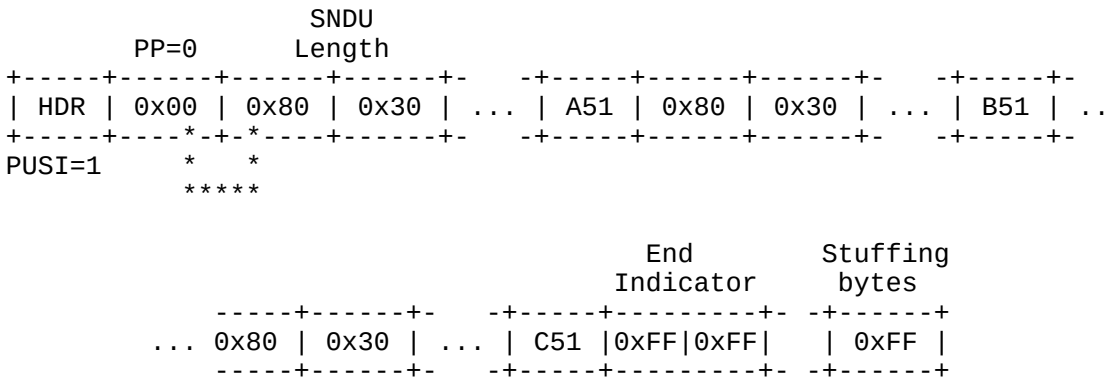


*** TS Packet Payload Pointer (PP)
+++ ULE Length Indicator

Example A.5: Three 44B PDUs.

SNDU A is 52 bytes (no ULE destination NPA address) SNDU B is 52 bytes (no ULE destination NPA address) SNDU C is 52 bytes (no ULE destination NPA address)

The sequence comprises 1 TS Packet:



Appendix B: SNDU Encapsulation

An example of ULE encapsulation carrying an ICMPv6 packet generated by ping6.

```
ULE SNDU Length  :            63 decimal
D-bit value  :                0 (NPA destination address present)
ULE Protocol Type :           0x86dd (IPv6)
Destination ULE NPA Address : 00:01:02:03:04:05
ULE CRC32 :                   0x7c171763

Source IPv6 :                 2001:DB8:3008:1965::1
Destination IPv6 :            2001:DB8:2509:1962::2

SNDU contents (including CRC-32):

0000: 00 3f 86 dd 00 01 02 03 04 05 60 00 00 00 00 0d
0016: 3a 40 20 01 0d b8 30 08 19 65 00 00 00 00 00 00
0032: 00 01 20 01 0d b8 25 09 19 62 00 00 00 00 00 00
0048: 00 02 80 00 9d 8c 06 38 00 04 00 00 00 00 00 7c
0064: 17 17 63
```

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
Department of Scientific Computing
University of Salzburg
Jakob Haringer Str. 2
5020 Salzburg
Austria

EMail: bnocker@cosy.sbg.ac.at
Web: http://www.scicomp.sbg.ac.at/

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.