IEEE P1722 AVBTP Encapsulations

Version 0.01, 2007-06-24

Alan K. Bartky alan@bartky.net

Bartky Networks www.bartky.net

Send comments to AVBTP@listserv.ieee.org

Notice of copyright release

- **Notice:**
  - This document has been prepared to assist the work of the IEEE P1722 and IEEE 802 Working Groups. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein.
- **Copyright Release to IEEE:**
  - The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE's name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE's sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that this contribution may be made public by the IEEE P1722 Working Group or the IEEE 802 Working Group.

Revision History

Rev	Date	Comments
0.01	2007-06-24	First version using formats and notes based on what probably will start going into the draft P1722 specifications.
0.02	2007-06-27	Added draft proposal for fragmentation for discussion on how to handle large CIP packets broken up into smaller packets for the case of 1394/61883 to AVBTP /61883 interworking. Cleaned up some diagrams to correct areas “grey” or white depending on scope of fields versus the diagram (what the diagram was trying to convey as fields of interest).

References:

- Draft AVBTP over IEEE 802.3 AVB stream data format Version 0.02 (Alan Bartky)
  - http://www.avbtp.org/contributions/avbtp-bartky-proposed-stream-data-format-v0-02-2007-03-27.pdf
- IEEE P1722 AVBTP assumptions (Alan Bartky)
  - http://www.avbtp.org/contributions/avbtp-bartky-assumptions-v0-02-2007-06-27.pdf

Design Assumptions

- AVBTP shall use 802.1AS for time base
- AVBTP shall be able to react to change in 802.1AS time (user changing time of day, change in Grandmaster, etc. (see 802.1AS assumptions from AVB document)).
- 61883 format over AVBTP will support presentation time in the same manner as 1394/61883 using the SYT field and in 24.576 MHz cycle time based on 802.1AS clock.
  - 61883-4 & 61883-7: Source Packet Header format with 0-127 seconds, 0-7999 8 kHz cycles, 0-3072 24.576 MHz sub-cycles.
  - All other 61883 encapsulations: CIP header format with 0-15 8 kHz cycles, 0-3072 24.576 MHz sub-cycles.
- AVBTP 61883 presentation time shall be relative to the 802.1AS clock
  - Adapt 1394 AV/C Function Control Protocol (FCP) for use in 61883 over AVBTP.
  - Allow for Proprietary encapsulations via different subtype
  - Allow for other future expansions via different subtypes.

Encapsulation Assumptions

- Approved by Consensus:
  - For AVBTP stream data frames, MAC Destination Addresses shall always be multicast addresses and shall be unique for the Layer 2 network. This address shall be used for stream identification.
  - For AVBTP stream control frames, MAC Destination Address may be unicast, multicast or broadcast depending on the specification of the usage of each AVBTP control frame.

Encapsulation Assumptions

- Approved by Consensus:
  - All talkers shall always send stream data frames with 1st Ethertype field set to 0x8100 for 802.1 P/Q type.
  - For AVBTP, talkers and controllers are not required to send stream control frames with an 802.1 P/Q tag.
  - All devices must always be able to accept data and control frames with an 802.1 P/Q tag.

Encapsulation Assumptions

- Approved by Consensus:
  - VLAN Identifier (VID), 12 bits:
    - The VID is a VLAN and not a Stream Identifier
    - AVBTP stations must support VLAN ID of zero to send or receive for stream data traffic.
    - AVBTP stations are recommended to support other VLAN IDs, but it is not required.
    - Receiving AVBTP stations not supporting VLANs or if supported and configured for a given set of VLANs shall discard any frames for which it is not a member of the specified VLAN.
  - Canonical Format Indicator (CFI), 1 bit
    - AVBTP will only support CFI of zero.
  - Priority Code Point (PCP), 3 bits:
    - For data streams, AVBTP shall always specify class 5 or class 4 traffic.

Encapsulation Assumptions

- Approved by Consensus:
  - AVBTP subtype field (8 bits):
    - 0x00: “61883 stream type data” (Editor’s note: placeholder name, suggestions welcome)
    - 0x01: “61883 stream type control” (Editor’s note: placeholder name, suggestions welcome)
    - 0x02: AV/C control
    - 0x03-0xFD: Reserved for future protocols
    - 0xFE: Extended OUI type (Vendor specific)
    - 0xFF: Reserved for future version of this standard
  - Subsequent parsing of AVBTP packets shall be based on subtype.

Encapsulation Assumptions

- Approved by Consensus:
  - AVBTP/61883 stream data “other” field (8 bits):
    - >> least significant bit for Timestamp valid
    - >> other bits reserved.
  - AVBTP/61883 source timestamp field (32 bits)
    - Data field shall express time related to the 802.1AS Global Clock
    - Format TBD (based on decision from 802.1AS)
    - Full resolution target at ~1 second.
  - Stream Data Length Field 16 bits
    - Same as used in 1394/61883 CIP header

Encapsulation Assumptions

- Approved by Consensus:
  - 61883 over AVBTP shall support the same fields using by IEEE 1394 Isochronous data packets
    - Length Field 16 bits
    - Isochronous Data Format (tag) field, 2 bits:
      - Supported by AVBTP:
        - » 00 binary, “data field unformatted” (used by Instrumentation & Industrial Digital Camera (IIDC) 1394 trade association specification)
        - » 01 binary, CIP header is present
      - Not supported by AVBTP:
        - » 10 binary: Reserved by IEEE 1394
        - » 11 binary: Global asynchronous stream packet (GASP) format (Used in 1394 for Serial Bus to Serial Bus bridges)
    - Type code (tcode), 4 bits:
      - Shall be fixed value of 1010 binary (same as 1394 Isochronous packet format)
        - » Set to 0xA on transmit
        - » Ignore on receive

Encapsulation Assumptions

- Approved by Consensus:
  - CIP header 1st quadlet indicator, 2 bits
    - Fixed at 00 binary
  - Data Block Size (DBS), 8 bits
    - Same definition as currently in 61883, size of Data Blocks in Quadlets
      - 0: 256 quadlets
      - 1-255: 1-255 quadlets
  - Quadlet Padding Count (QPC), 3 bits
    - For all types of 61883 as defined today, this field is always zero.

Encapsulation Assumptions

- Approved by Consensus:
  - Source Packet Header (SPH) indicator, 1 bit
    - If one
      - Then AVBTP packet contains 61883-4 or 61883-7 (or future) source packets.
    - If zero
      - Then AVBTP packet does not contain source packets (contains integer number of Data Blocks)
  - Reserved (Rsv), 2 bits
    - Reserved (currently not used by 1394/61883), set to zero, ignore on receive.
  - Data Block Count, 8 bits
    - Sequence number of 1st Data Block in the packet
    - Same meaning as in 61883 over 1394

Encapsulation Assumptions

- Approved by Consensus:
  - CIP header 2nd quadlet indicator, 2 bits
    - Fixed at 10 binary
  - Stream Format, 6 bits
    - Same values as currently defined for 61883
  - Format Dependent Field (FDF), 8 bits if SPH=0, 24 bits if SPH=1
    - Same values as currently defined for 61883
  - SYT field (1394 cycle time based presentation time for SPH field equals 0)
    - Mandatory for use by AVBTP end stations
    - Same as 1394/61883

Encapsulation Assumptions

- Approved by Consensus:
  - CIP header 2nd quadlet
  - Channel ID (0-63)
    - 0-30 & 32-63: originating channel ID from 1394 network.
    - 31: originating source is on AVB network (native AVB)
  - Source ID (0-63)
    - 0-62 originating Source ID from IEEE 1394 network
    - 63 originating source is on AVB network
  - Reserved (2 bits)
    - Same as 1394/61883
      - » Set to 0 on transmit
      - » Ignore on receive.
  - "sy" field (4 bits) (currently used by 1394 for Digital rights management).
    - Same as 1394/61883

Encapsulation Assumptions

- Proposals:
  - For all class 5 traffic, limit maximum transmission unit size in order to limit total transmission time on and 802.3 100 megabit (including preamble and inter-frame gap to 75% of 125µs)
  - For all 61883 type traffic, limit maximum data payload to 256 quadlets (1024 bytes)
    - >> Editor's note: Needs work. Current consensus to break on event boundaries:
      - Data Blocks
      - Source Packets

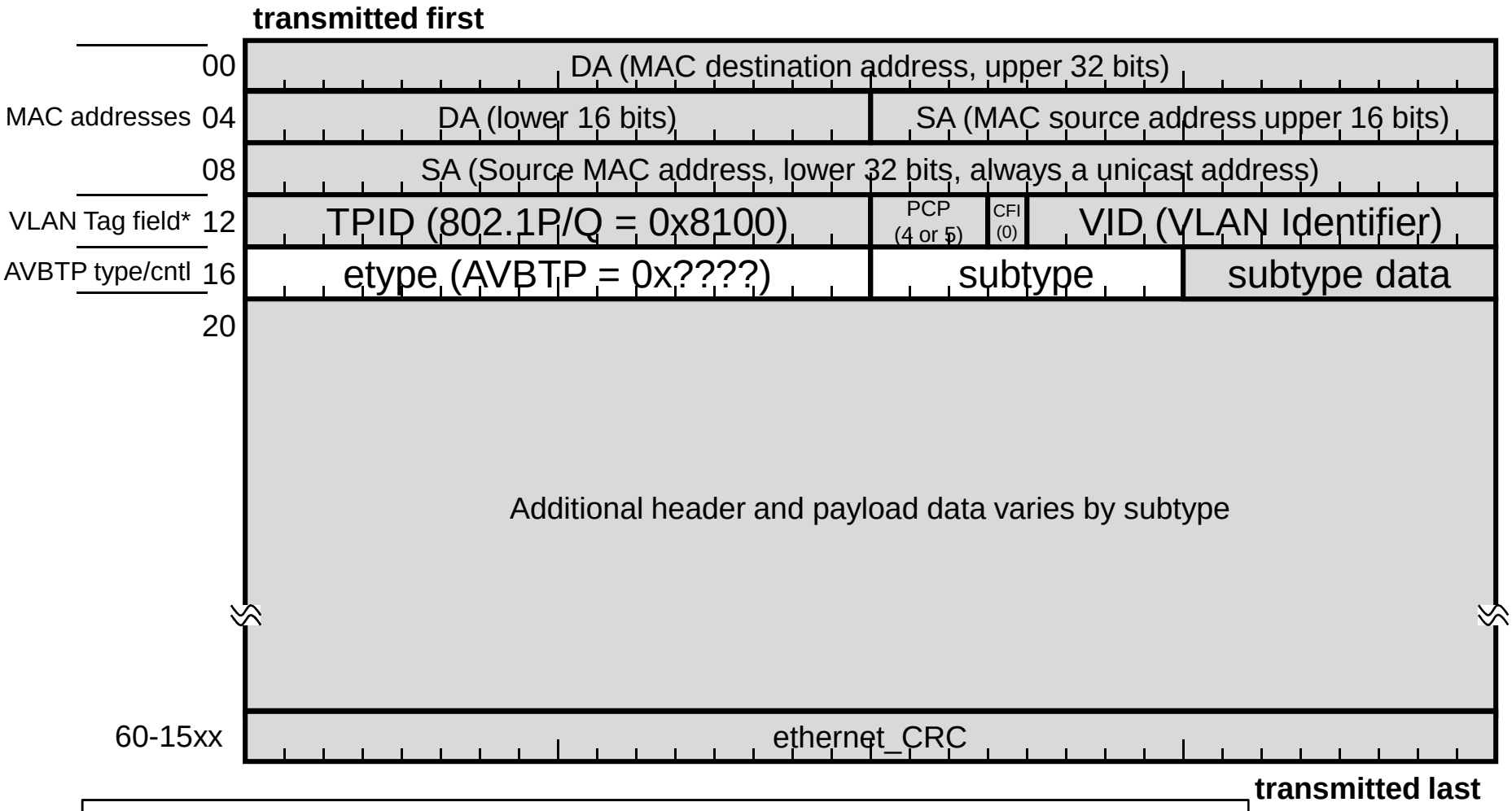
Encapsulation Assumptions

- Questions:
  - Closed
    - Should we standardize the length field for all AVBTP formats?
      - Consensus: No, all data after the subtype field shall be subtype dependent.
    - Is there other control traffic that will need other encapsulation options?
      - *Consensus: Yes, 61883 over AVBTP will need one for stream control and one for AVC. See current encapsulation proposal for details. Other future protocols over AVBTP will need them as well.*

Encapsulation Assumptions

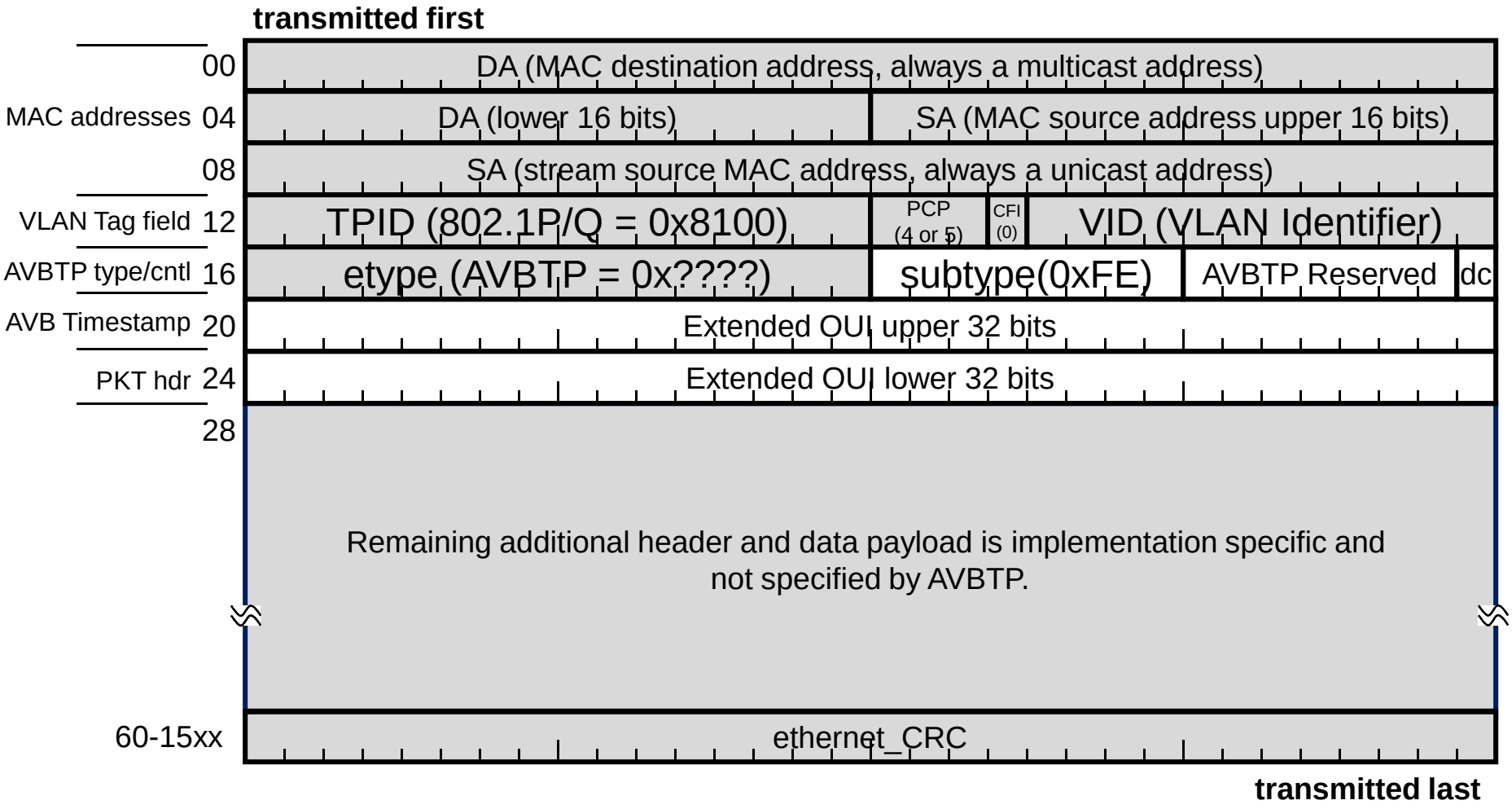
- Work Items:
  - Need to come up with format to allow proprietary encapsulations (define subtype and any fields we deem necessary to ensure consistency)
    - >> Alan to come up with initial proposal for 64 bit Extended OUI and subtype of 0xFE

Draft AVBTP packet

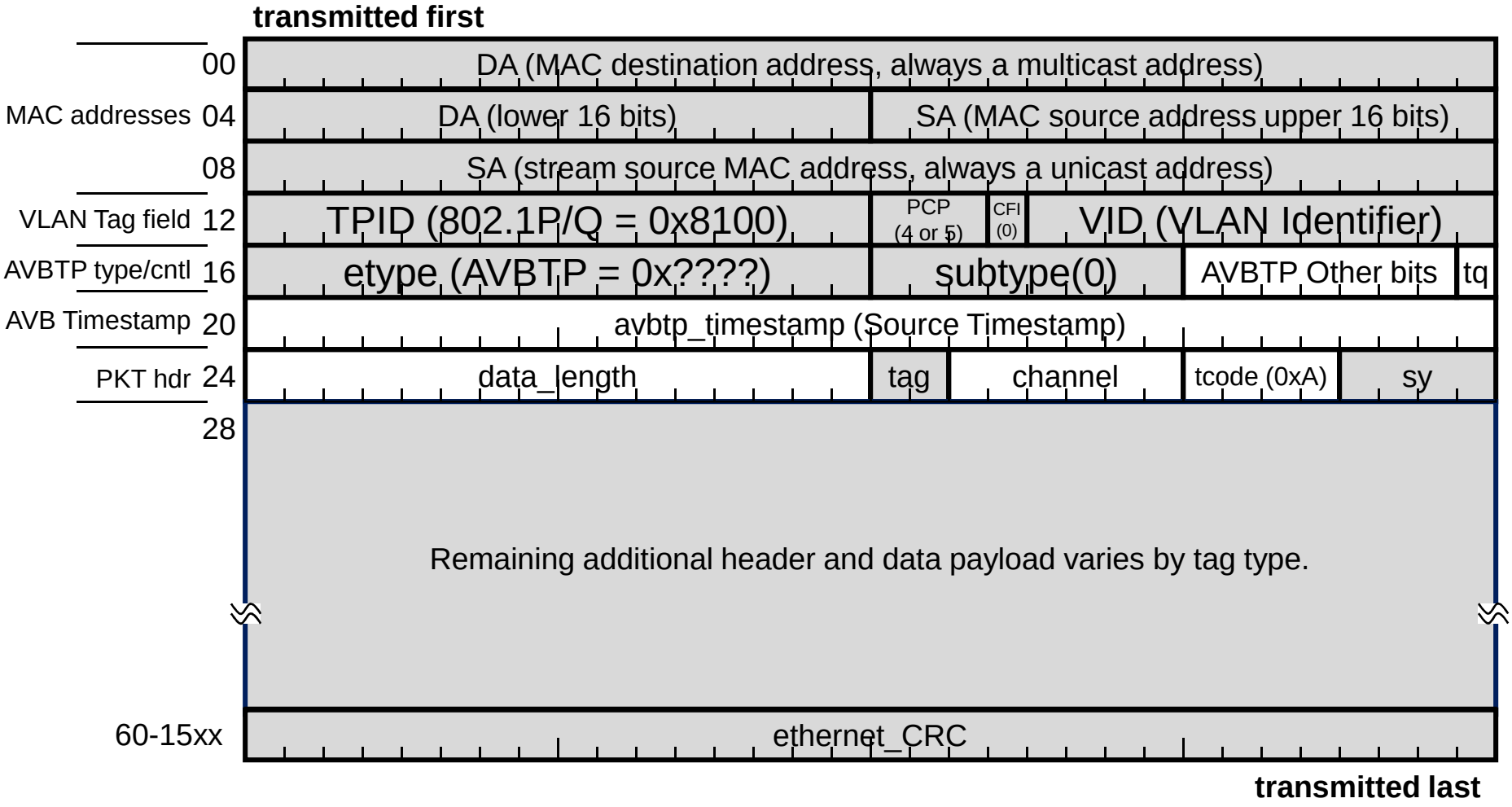


***Note: VLAN Tag field is mandatory for some subtypes and optional for others**

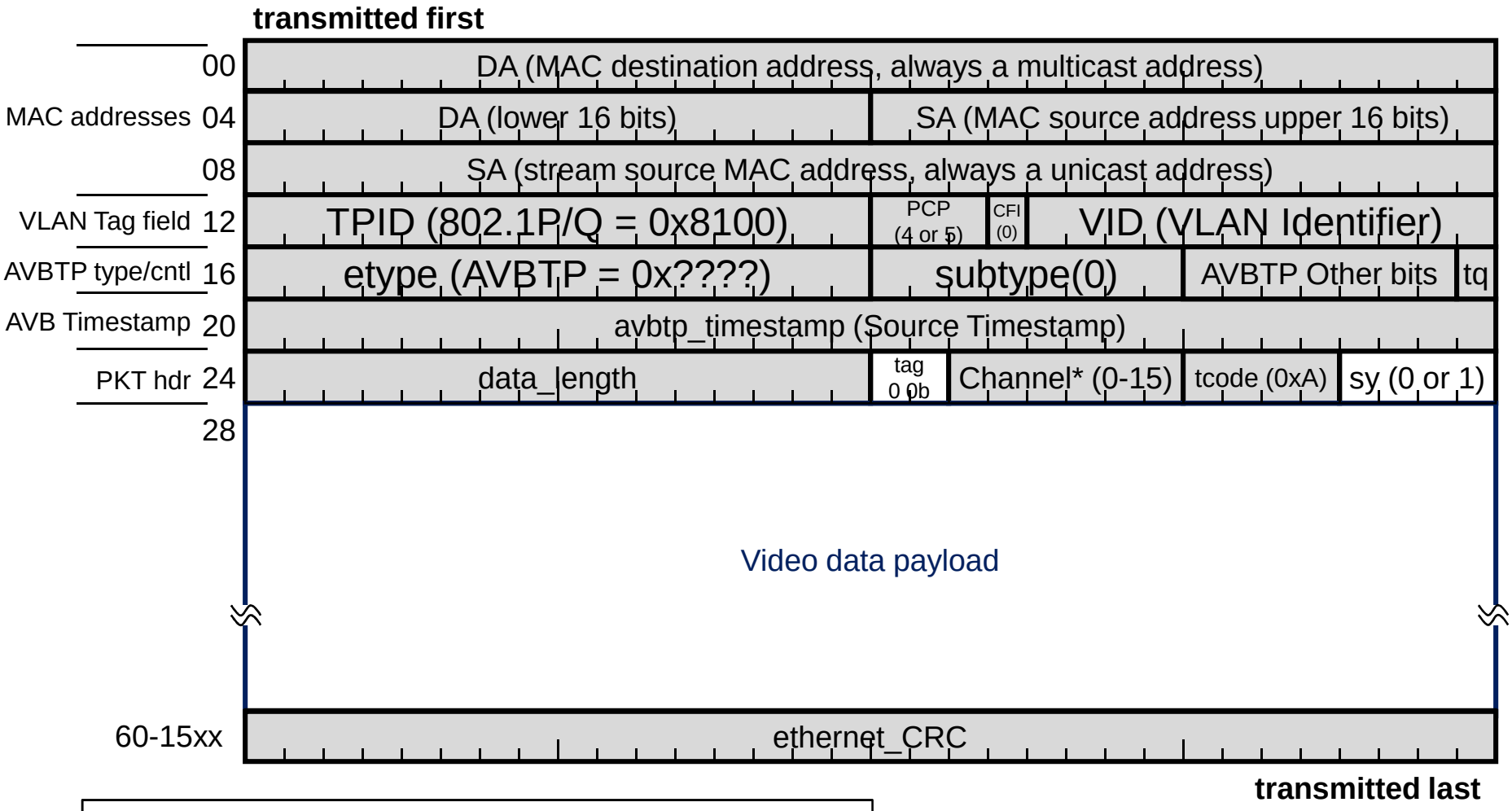
Draft AVBTP Proprietary format



Draft AVBTP Subtype 0, 61883/IIDC data packet

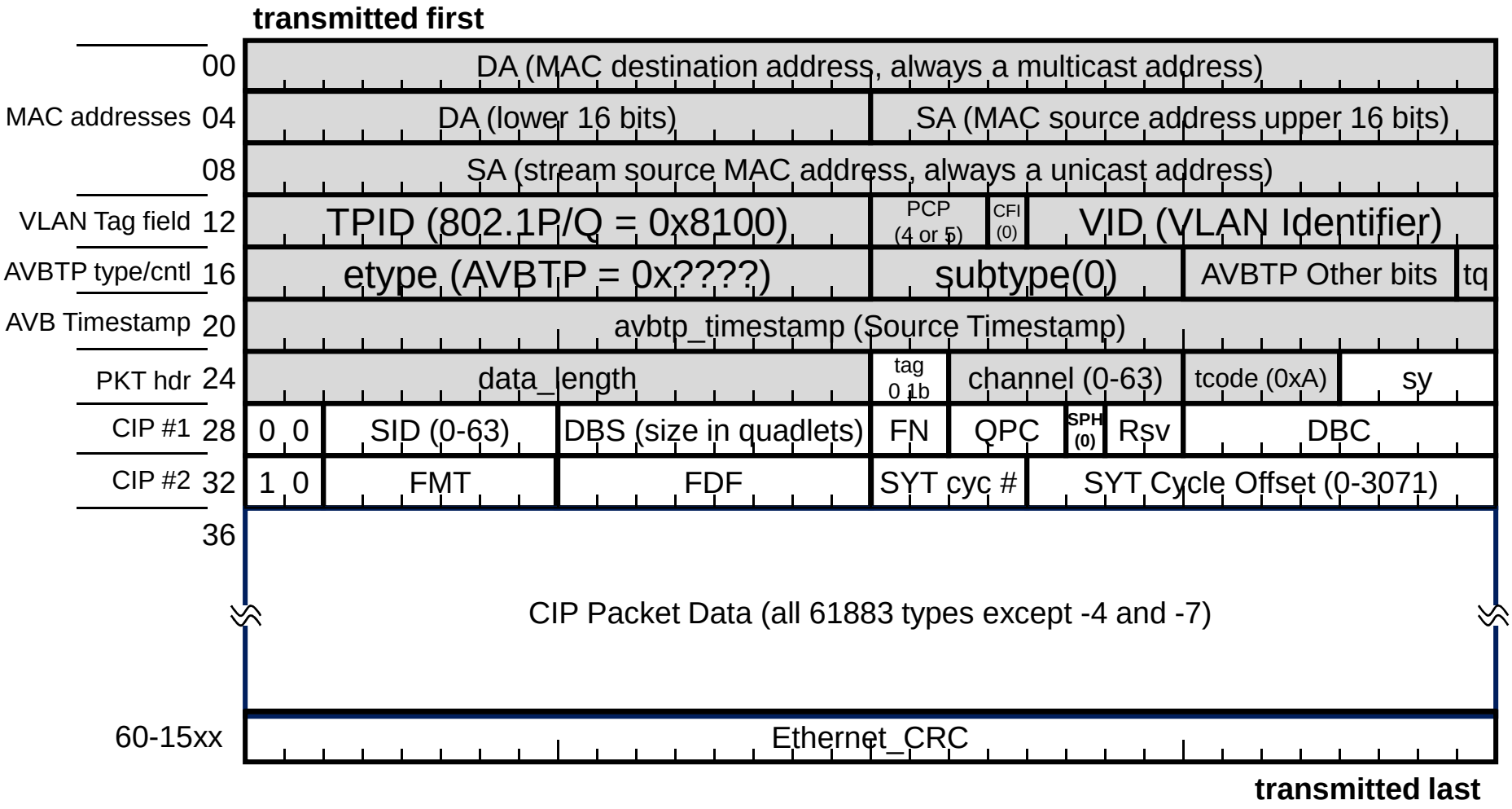


Draft AVBTP IIDC Stream Data packet

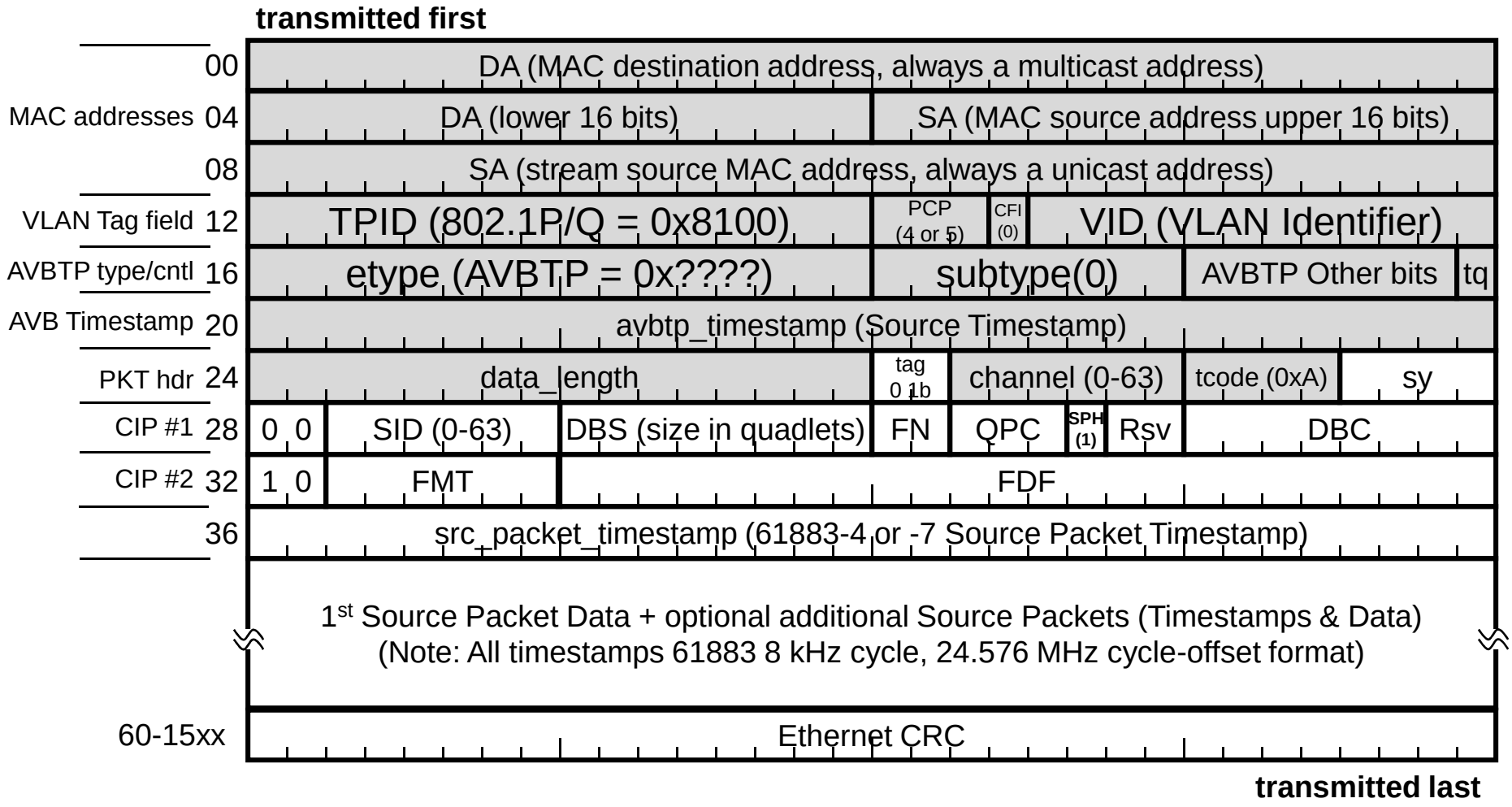


***Note: Current standard for IIDC restricts channel ID**

Draft AVBTP CIP Stream Data packet, SPH(0)

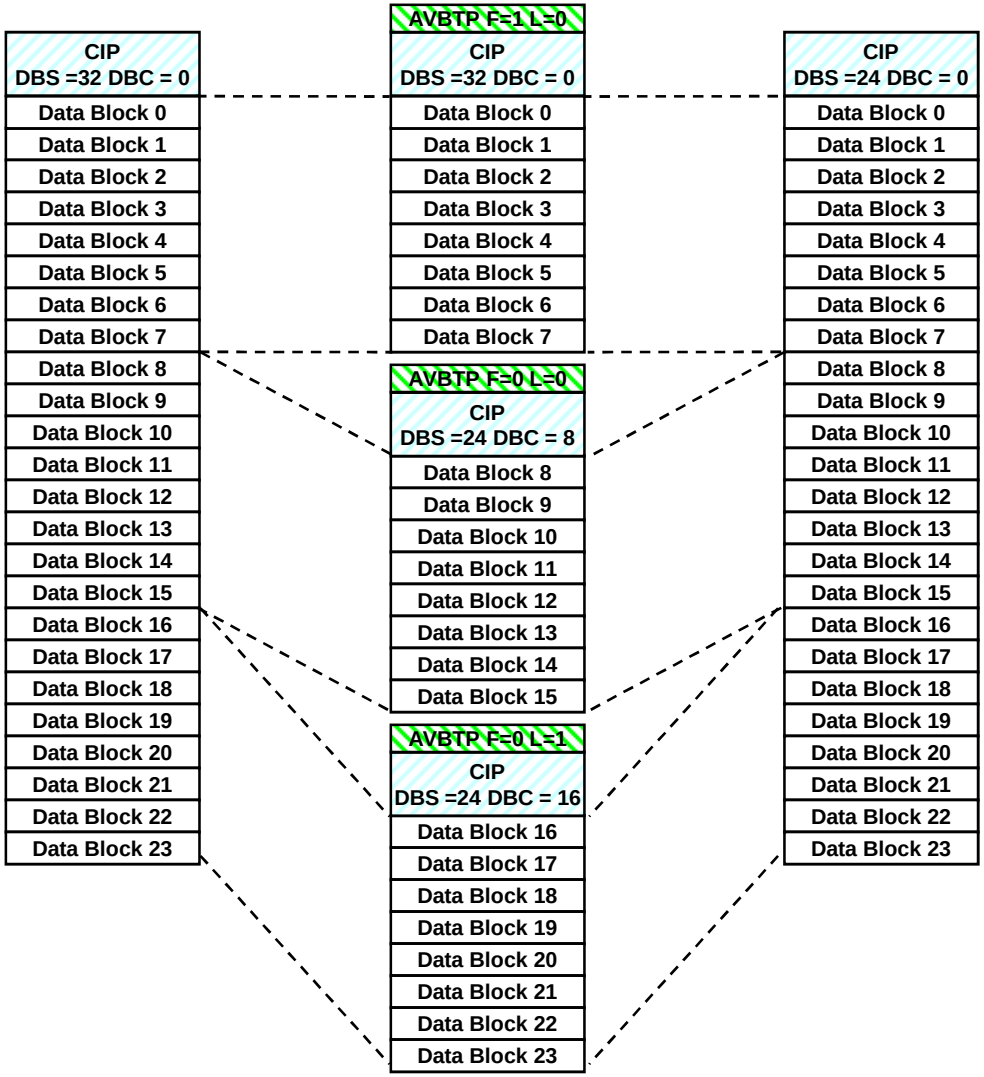


Draft AVBTP CIP Stream Data packet, SPH(1)



Example Data Block Fragmentation

- Example fragmentation based on Data Blocks
- Use case, 61883-6, 192 kHz audio, 32 channels.
  - (24 samples per 8 kHz cycle) * (4 bytes per sample) * 32 channels = 3072 total bytes contained in 24 Data Blocks each 32 quadlets long.
- Need to break into 3 Ethernet Frames
  - First Frame: F bit = 1, L bit = 0, DBC = 0, SYT field copied from source.
  - Middle Frame: F bit =0, L bit = 0, DBC = 8, SYT field set to all ones.
  - Last Frame: F bit = 0, L bit = 1, DBC = 16, SYT field set to all ones
- Reassembly if needed is straight forward:
  - DBS, DBC, SYT copied from packet with F bit = 1,
  - Length calculated by adding lengths of all fragments minus CIP headers of intermediate or last packets.
  - Lost fragments detected by sequence number mismatch (just like for normal streams)



Example Source Packet Fragmentation

- Example fragmentation based on Source Packets
- Use case, 300 megabits/second 422P@HL MPEG-2 (1920x1080 @ 30Hz)
  - ~ 25 MPEG packets per 8 kHz cycle
  - 188 byte packets with 1 quadlet header(192 bytes per Source packet), 8 x 6 quadlet Data Blocks per source packer as per 61883-4 specification.
  - SYT field in source packet headers, not in CIP header.
- Need to break into 4 Ethernet Frames
  - First Frame: First Packet (F) bit = 1, Last Packet (L) bit = 0, DBC = 0
  - 2 Middle Frames: F bit =0, L bit = 0, DBC = 42 & 84
  - Last Frame: F bit = 0, L bit = 1, DBC = 126
- Reassembly if needed is straight forward:
  - DBS & DBC copied from packet with F bit = 1,
  - Length calculated by adding lengths of all fragments minus CIP headers of intermediate or last packets.
  - Lost fragments detected by DBC sequence number mismatch (just like for normal streams)