IEEE 1722 Control Format
Generic Image Sensor Transport

2020-07-28, Rev 5 (slide 8~15 added)

Edited by Yong Kim

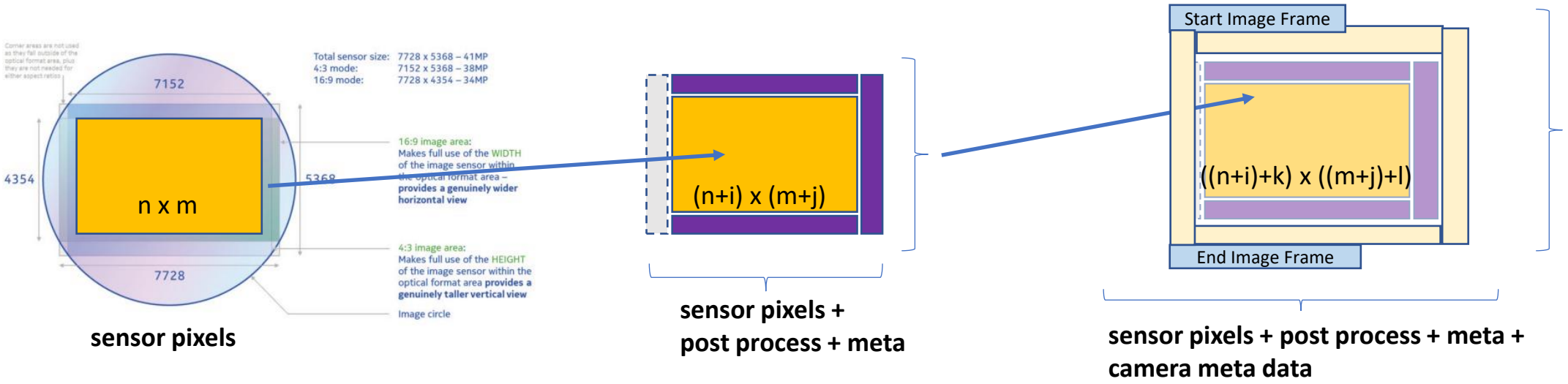
yongkim at axonne dot com

Motivation

- One of the major justifications of automotive Multi-gig Ethernet PHY (IEEE 802.3ch) is network connection of high bandwidth autonomous drive (AD)/advanced driver assistance system (ADAS) sensors, such as uncompressed AD and parking cameras.
- Image sensors outputs n x m matrix of n pixels per line x m lines.
- Image sensors meta-data outputs, often, are embedded in the extra row or columns.
- There is a need to transport image sensor outputs, row by row, over high-speed serial interface; P2P, or ideally real P2MP network, transport protocol that supports time sensitive networking, i.e. IEEE 1722 AVTP.
- Avoid pre-standard and proprietary implementation islands that form once system productions start; Avoid artificial turf defense when these form. There is no intrinsic value to having multiple different but functionally equivalent approaches.
- Avoid compatibility issues, implement transparent transport, i.e. bits in, bits out, layered.
- Stated goal of 1722b – serve the industry and perform necessary revision quickly – fits well with a goal of this proposal.

Cameras used in ADAS and Autonomous Drive

- 1 MP, 1.3MP, 2MP, and 8 MP, often leveraged from consumer electronics R&D, e.g. cell phones.
  - The same image sensors but instead of RGBG (Red, Green, Blue, Green) filter lens, it may use alternate color filters, RCCC (Red, Clear, Clear, Clear), or other combinations that are optimized for machine vision and object detection.
  - Camera sensor wrapped around camera system SoC may outputs to a number of standard interfaces.
  - Many of the ADAS/AD camera system leverages the mobile camera (integrated Sensor + system SoC) R&D.
- Transport Considerations
  - Camera sensors output what it sees in various configurable formats
  - Camera system SoC output what it processes in various configurable formats
  - Transport should just transport bits, bytes. In camera terms, [frame start] [lines]…[lines][frame end].

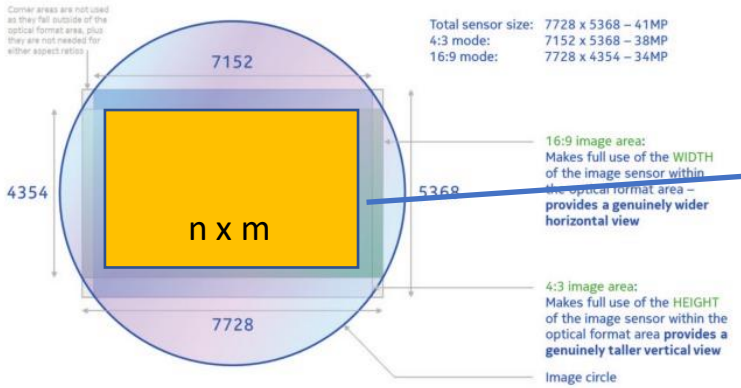


sensor pixels

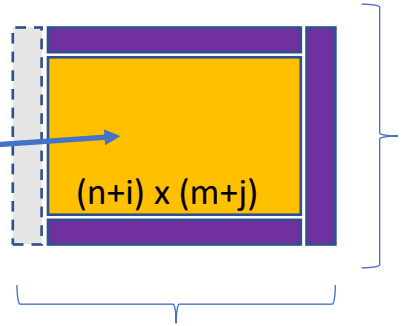
sensor pixels + post process + meta

sensor pixels + post process + meta + camera meta data

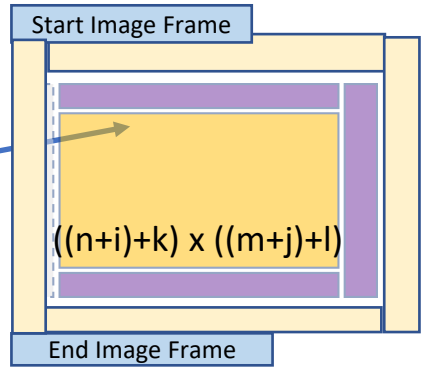
1722b to transport image data - image sizes



sensor pixels



sensor pixels +
post process + meta



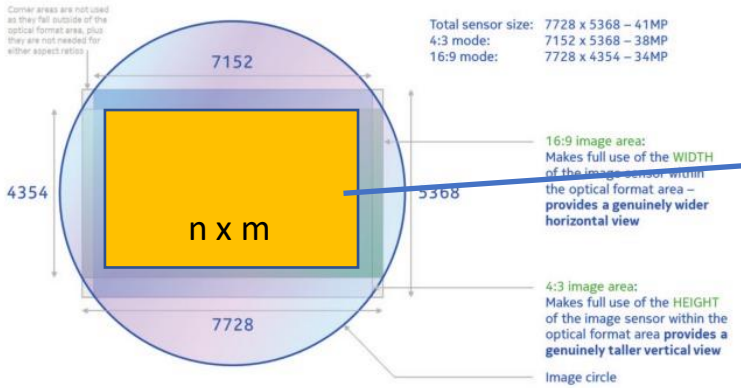
sensor pixels + post process + meta +
camera meta data

- Image sensor + lens + filter
- Just accept n x m pixels of some resolution
- No need to know, don't care about content (from transport perspective)

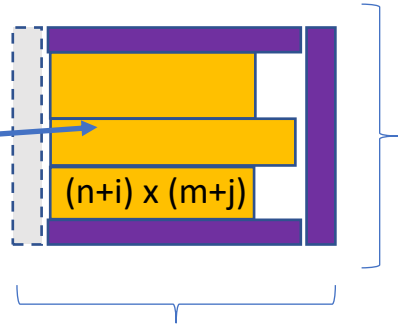
- Pixels has meaning (color space, depth, HDR processed, etc. whatever).
- Added pixel space may be added for meta data that increases n x m to be a bit bigger.
- No need to know, don't care about content (from transport perspective)

- Transparent transport of the previous content.
  - Start Image Frame
  - Line by line
  - End Image Frame.
  - May have added metadata but not distinguishable.

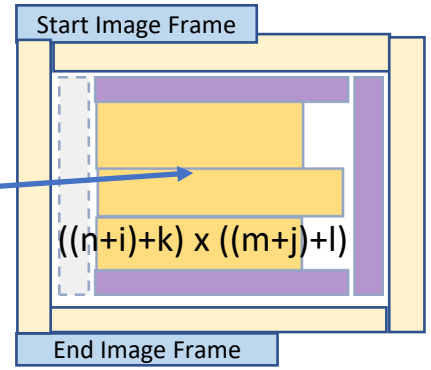
1722b to transport image data - discuss



sensor pixels



sensor pixels + post process + meta



sensor pixels + post process + meta + camera meta data

- Image sensor + lens + filter
- Just accept n x m pixels of some resolution
- No need to know, don't care about content (from transport perspective)

- Each image is on its own.
- Frame, Line, .. Line, Frame
- Don't assume fixed image size/depth.
- Each image type is separate stream (e.g allow for interspersed raw and HDR images, or dual image sensors onto single transport.)

- Each frame is on its own.
- FS, 'embedded', … , line, …, embedded', FE
- Don't assume fixed image size/depth.
- Transparent transport of packet header – issues?

1722b to transport image data – requirements

Background

- Video Stream format is optimized for rendering device, e.g. TV
  - Presentation Time ➔ renderer's use.
- Example: Brake actuator needs presentation time – currently in time-sensitive header.
- ACF format allows for input data timestamp, e.g. Lin message time stamp, camera shutter.
  - ACF time ➔ sensor input's use.
- Example: Image sensor needs input time – currently only in ACF

Some Requirements for potential 1722 transport.

- Format with input time (e.g. as in existing ACF sub-type message timestamp)
- Raw video format that supports variable length lines (i.e multiple rectangles in a single frame).

1722b to transport image data – considerations

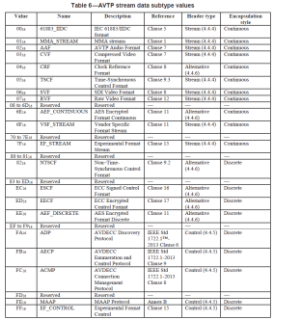
- Raw image : e.g. not i.e., 4:4:4, 4:4:4:4, …,
- ‘Improved image’: e.g. 8b(4:2:2), 10b(4:4:2), 12b(4:4:4), 16b(5:6:5)…
- Resolutions: e.g. 1MP, 1.3MP, 2MP, 4MP, 8MP,…
- Line Lengths:

pixel/line	+8		12 bit/pixel in bytes	16 bit/pixel in bytes
1280	1288		1932	2576
1920	1928		2892	3856
2560	2568		3852	5136
3840	3848		5772	7696

- Should allow for transparent transport of a line of pixel bytes in multiple Ethernet frames and reduce max latency for other TSN flows w/o use of preemption.
- Be aware of robustness of CRC32 and payload – avoid use of jumbo.
- For those who loves linear math and error functions, see referenced in: http://www.ieee802.org/3/bj/public/jul12/cideciyan_01_0712.pdf

1722 Transport sub-type considerations - detail

AVTPDU Subtype Summary (4.4.3 & Table 6)



- Header Type: [Stream | Alternative | Control]
  - **Stream**: stream_id, avtp_[presentation]timestamp, etc ← TS ACF (TSCF)
  - **Alternative**: **no** stream_id, **no** avtp_[presentation]ts, etc. ← NTS ACF (NTSCF)
  - **Control**: stream_id, **no** avtp_ts, etc ← 1722.1, MAAP, etc.

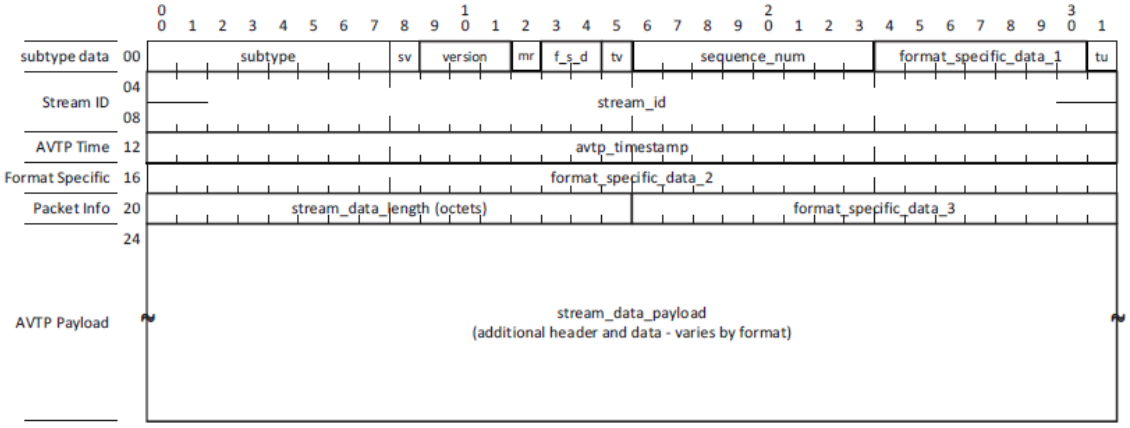


Figure 9—AVTPDU common stream header

Audio, Video, TS Control Format

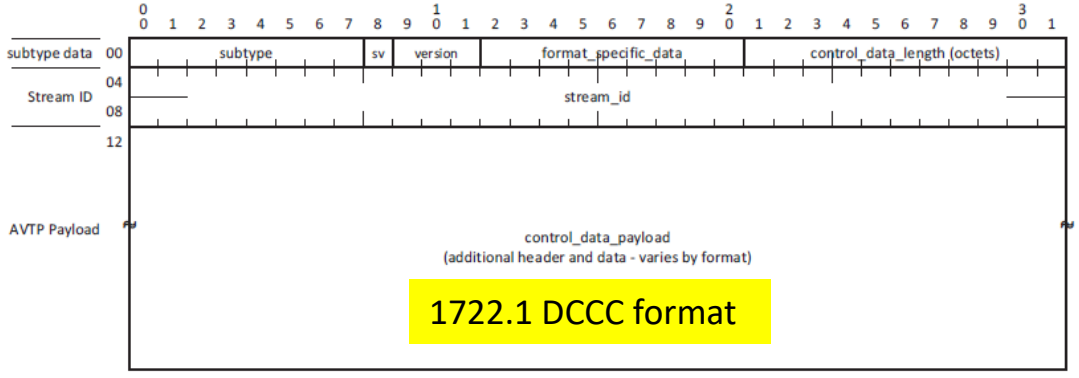


Figure 11—AVTPDU common control header

1722.1 DCCC format

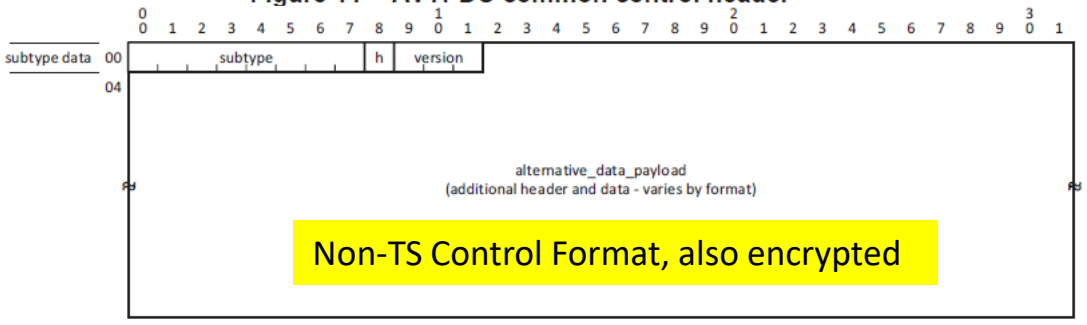


Figure 13—Alternative AVTPDU header

Non-TS Control Format, also encrypted

AVTP TS and Non-TS Control Format subtypes has “origination” time-stamps, differentiating “presentation” time-stamps.

IEEE 1722 Raw Video Sub-type – summary

- Excluding IEC 61883/IIDC format, Compressed Video Format (CVF), SDI Video Format (SVF), leaves just RAW Video Format (RVF)
  - IEC61883/IIDC format has color space and resolution – but camera use case no relevance to IEC61883.
  - Compressed Video Format has MJPEG, H.264, JPEG2000 (we should add H.265 in 1722b) -- mild relevancy to compressed (e.g. some Parking camera)
  - SDI Video Format has fixed raster structure (user space available but not good enough).
- RAW Video Format
  - Stream_id – G
  - Avtp_[presentation]timestamp – NM (not meaningful)
  - Pixel depth – 0x0, 0x5~0xE (reserved), 0xF (user) – total of 11 resv + 1– NG but ok to not use.
  - Pixel format – 0x5, 0xC~0xE (reserved), 0xF (user) – total of 4 resv + 1 – NG but ok to not use.
  - Frame rate –16 bit field (20 used, 1 user, rest reserved).
  - Color space - 0x0, 0xA~0xE (reserved), 0xF (user) – total of 6 resv + 1 – NM (not meaningful)
  - Num_lines – 4 bit -- # of whole lines in this PDU – N/A (not applicable)
  - Line number & total line – 16 bit (good for 64K lines x 113K across for 9x16) – G
  - Active Pixels – 16 bits – fixed for the frame (every line) – NG
  - i_seq_num – sequence of frames for partial lines – G
  - Flags: ap, f, ef, evt, pd, I – extend to other respective fields valid.

Excellent template for the generic image sensor transport over AVTP Control Formats (ACF) [that supports origination timestamp]

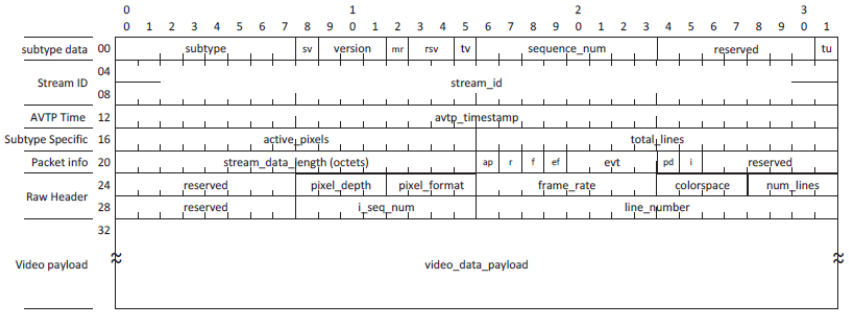


Figure 78—Raw Video PDU Format

AVTP Control Format Summary (9, Table 22)

Table 22—ACF message types

Value	Name	Description	Subclause
00_{16}	ACF_FLEXRAY	FlexRay™ message	9.4.2
01_{16}	ACF_CAN	Controller Area Network (CAN)/CAN with Flexible Data-Rate (CAN FD) message	9.4.3
02_{16}	ACF_CAN_BRIEF	Abbreviated CAN/CAN FD message	9.4.4
03_{16}	ACF_LIN	LIN® message	9.4.5
04_{16}	ACF_MOST	MOST® message	9.4.6
05_{16}	ACF_GPC	General purpose control message	9.4.7
06_{16}	ACF_SERIAL	Serial port message	9.4.8
07_{16}	ACF_PARALLEL	Parallel port message	9.4.9
08_{16}	ACF_SENSOR	Analog sensor message	9.4.10
09_{16}	ACF_SENSOR_BRIEF	Abbreviated sensor message	9.4.11
$0A_{16}$	ACF_AECP	IEEE Std 1722.1 AECP message	9.4.12
$0B_{16}$	ACF_ANCILLARY	Video ancillary data message	9.4.13
0C to 77_{16}	Reserved	Reserved	—
78 to $7F_{16}$	ACF_USER	User-defined ACF message	—

- ACF Subtypes: [Stream | Alternative]
  - **Stream**: stream_id, avtp_[presentation]timestamp, etc ← TS ACF (TSCF)
  - **Alternative**: **no** stream_id, **no** avtp_[presentation]ts, etc. ← NTS ACF (NTSCF)
- Alternative Subtype: “Steering input – origination timestamp”,
  Stream Subtype: “Wheel angle actuation – presentation timestamp”.
  … but presentation timestamp could instead be in the alternative & ACF msg type.

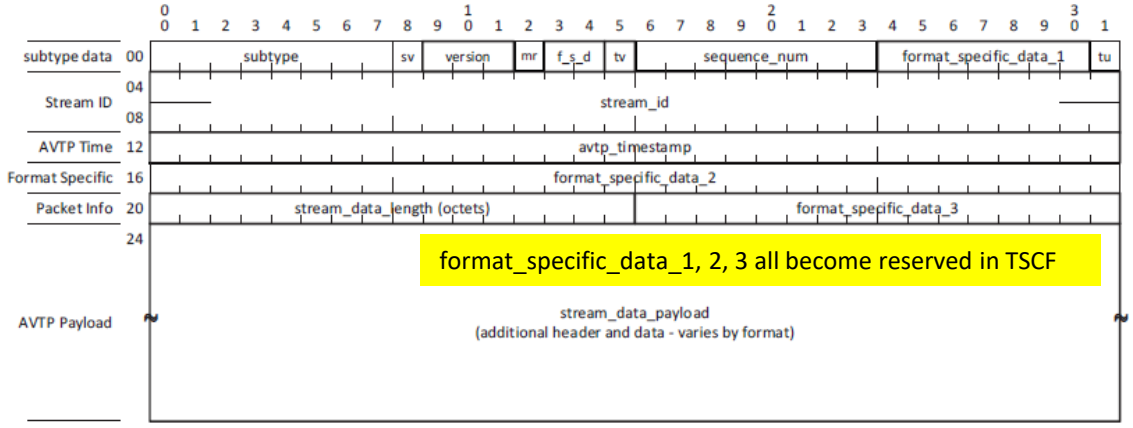


Figure 9—AVTPDU common stream header

Audio, Video, TS Control Format

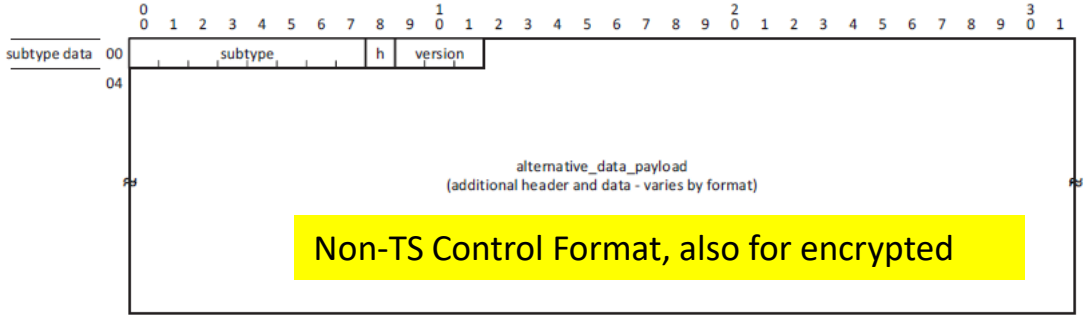


Figure 13—Alternative AVTPDU header

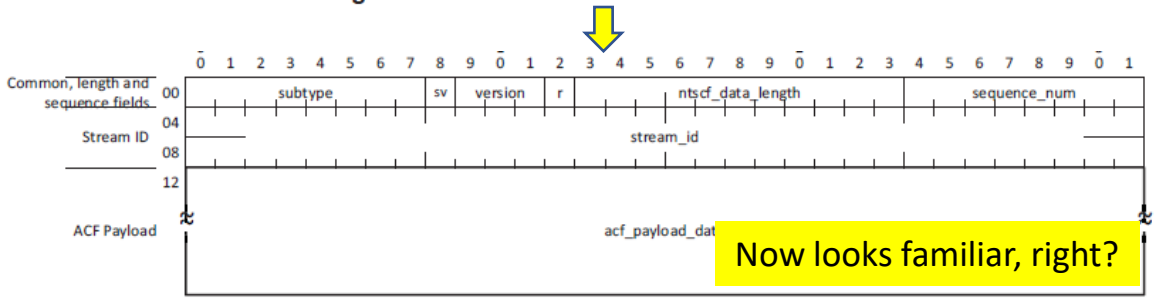


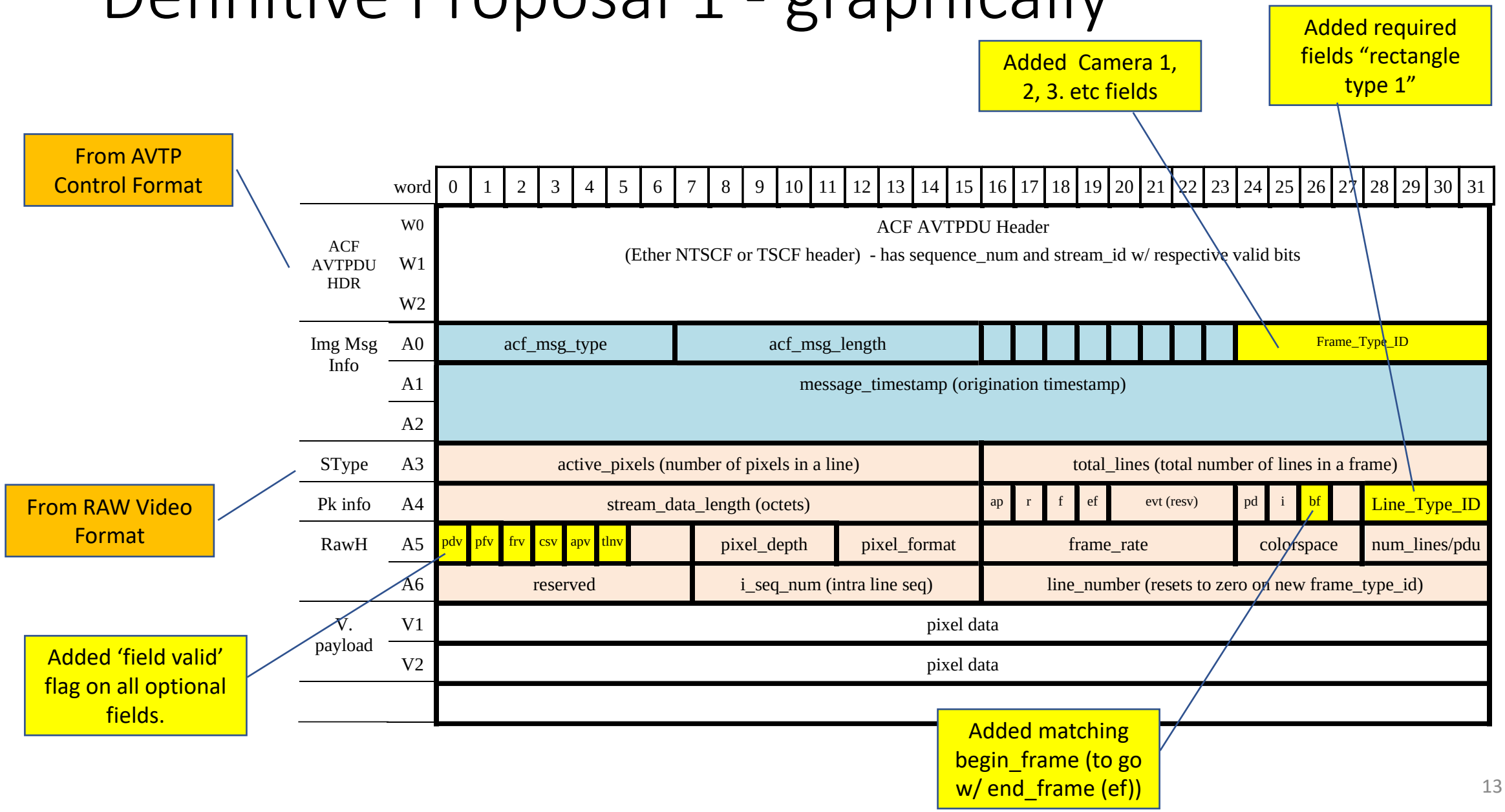
Figure 50—Non–Time-Synchronous Control Stream PDU Format

AVTP TS and Non-TS Control Format subtypes has “origination” time-stamps, differentiating “presentation” time-stamps.

Definitive Proposal – toward agreement

- Subtype: ACF – NTSCF. Discuss whether to allow TSCF.
- Indicate: Frame Start, Frame End, Line Start (implicit), Line # (w/ its valid flag), Line Sequence #, Line End (implicit).
  - Indicate: Frame Type (if multiple frame types are interspersed in a same avtp stream), for example single stream that may serve multiple inter-related camera outputs on single avtp stream.
  - Indicate: Line Type (“rectangle type 1”, “rectangle type 3”, etc)
- Indicate: Frame Origination Time Stamp – must be valid on Frame Start. May increment and vary for each of the lines (for non-global shutter imagers (rare))
- Color space, pixel depth, pixel format, -- copy from RAW format but with valid flag. -- allowed and optional -- further definition may not add further value. But allow them.
- Controversial?
  - Proposal: Do NOT allow ACF_payload concatenation of more ACF_payload and must be the 1st ACF_payload_data. For ease of implementation and optimum MTU size consideration.
  - Counter Proposal: If ACF_payload concatenation is not allowed (TLV fashion), then why NOT use RAW as the basis and add origination timestamp and may any other required modifications?
  - Discuss.

Definitive Proposal 1 - graphically



Definitive Proposal 2 - graphically

From AVTP Control Format

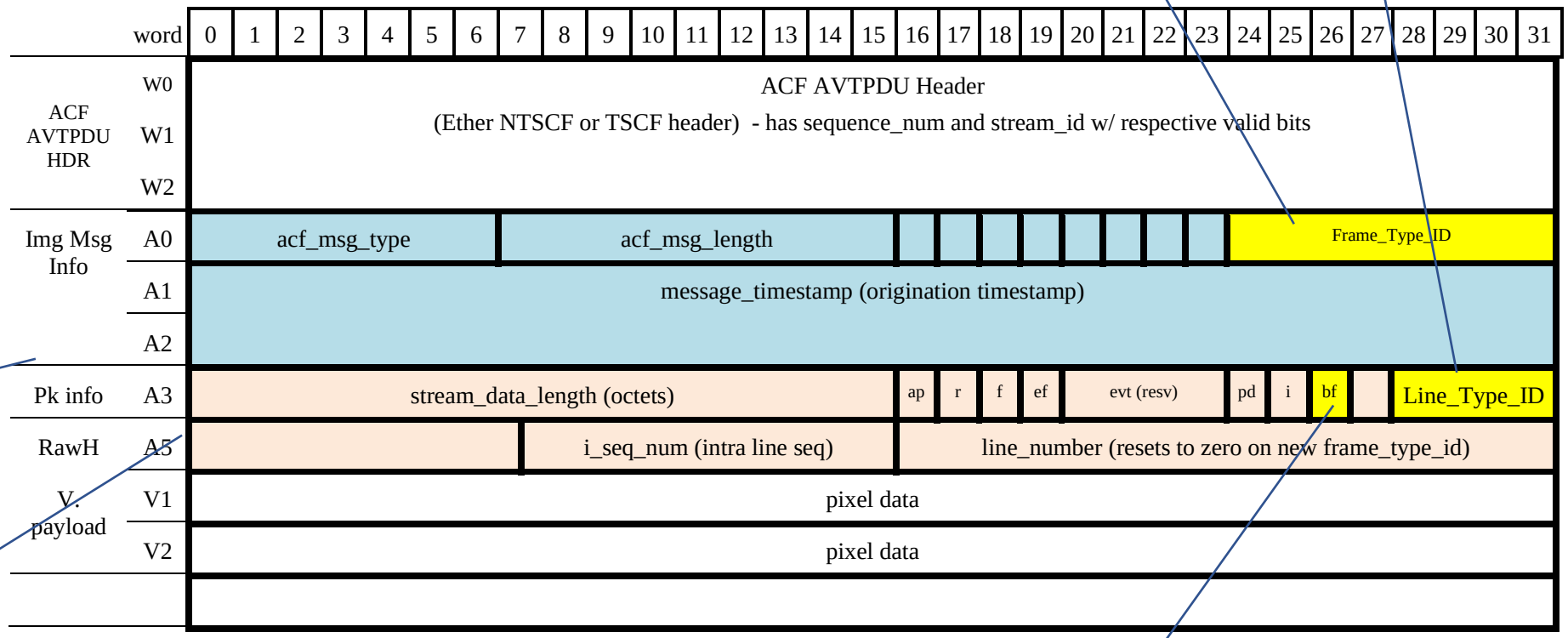
Started with RAW Video Format

Deleted all optional fields (deleted two 32 bit words)

Added Camera 1, 2, 3. etc fields

Added required fields “rectangle type 1”

Added matching begin_frame (to go w/ end_frame (ef))



Discuss and build consensus

- AVTP timestamp always has been used as a presentation timestamp. Changing this may be confusing.
- AVTP origination timestamp had been introduced for automotive (and industrial) sensor data over AVTP. Therefore AVTP Control Formats were defined.
- RAW Video format has many relevant fields, but some are not appropriate (as in number of pixels per line shall not change in a frame).
- Introduce a new image sensor format that parallels RAW Video format but under AVTP Control Format, where it is easy to be found (sensors).
- Discuss -