IEEE P1722 AVBTP assumptions

Version 0.02, 2007-06-27

Alan K. Bartky alan@bartky.net

Bartky Networks www.bartky.net

Changes since previous document in red text.

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
0.00	2007-06-14	First version for comments, created from previous AVBTP presentations and key assumptions from Don Pannell's AVB assumptions presentation
0.01	2007-06-24	Changed due to comments and discussions during June 20, 2007 face to face meeting. Made changes based on my notes and also updated slides from John Nels Fuller. Removed detailed encapsulations and comments/changes thereof to first draft AVBTP specification.
0.02	2007-06-25	Edited in changes based on review during teleconference 2007-06-25 and edited in summary proposals/discussions from Chuck Harrison & Alan Bartky (included reference to new encapsulation presentation with additional info).

References:

- Audio Video Bridging (AVB) Assumptions, Version 6 (Don Pannell)
  - http://www.ieee802.org/1/files/public/docs2007/avb-pannell-assumptions-0607-v6.pdf
- AVBTP Presentation Time (Dave Olsen)
  - http://www.avbtp.org/contributions/avbtp-olsen-presentation-time-0507.pdf
- 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
- AVB/AVBTP layering, management objects and data transfer processing study Draft 0.01 (Alan Bartky)
  - http://www.avbtp.org/contributions/avb-bartky-end-station-study-0507-v1.pdf
- AVBTP encapsulation assumptions Draft 0.02 (Alan Bartky):
  - http://www.avbtp.org/contributions/avb-bartky-encapsulations-v0-02-2007-06-27.pdf

Important Notes from Editor

- **Please review all consensus items and if you do not consent to them, please let the editor and the rest of the team know (i.e. silence implies consensus). We will use our weekly teleconference review any new consensus items.**

Overview

- This document is to collect high level requirements, ideas, concepts, etc. for Audio/Video Bridging Transport Protocol (AVBTP) for use in:
  - Building and verifying Consensus on key items
  - Documenting those key items for work on the main specification and other documents/contributions
  - Using as a checklist to make sure key items are taken care of.

Overview (continued)

- This document is broken up into major sections as follows:
  - General (leader: Alan Bartky?)
  - Encapsulation (leader: Alan Bartky)
  - Timing and Synchronization (leader: Matt Mora)
  - Session Management (leader: John Fuller)
  - Protocol layering and selected options from other protocols (e.g. 802.1AS, 802.1Qat, 802.1Qav, etc.) (leader: Alan Bartky?)
- For each assumption, it will be identified as a item that is:
  - Approved by Consensus
  - Proposal
  - Question
    - Open
    - Closed
  - Work Item

General Assumptions

- Approved by consensus:
  - AVB class 5 together with AVB class 4 cannot use more than 75% of a link's bandwidth
    - The Remaining 25% (or more) is used for Legacy (non-AVB) flows
  - Functional device type names
    - AVBTP will have Talkers, Listeners and Controllers
      - AVBTP will interoperate with AVB 802.1 bridges.
    - A Talker is the source of a stream
    - A Listener is a receiver of a stream
    - A Controller is a device that introduces and manages talkers and listeners, and manages groups of sessions.
    - Any physical device can be any combination of these
  - An AVBTP stream is between one talker and one or more listeners

General Assumptions

- Approved by consensus:
  - AVBTP will adapt the following 1394/61883 type protocols to run in an IEEE 802 environment.
    - 61883-2
    - 61883-4
    - 61883-6
    - 61883-7
    - BT.601 (to become 61883-8)
    - IIDC

General Assumptions

- Proposals:
  - Keep the AVBTP document “simple and pure”. No control/discovery/etc.
  - Keep it simple and close enough to 61883 that bridging to/from the most common forms of 1394 isochronous streams is a straight-forward problem that can easily be done in hardware.
  - AVBTP should be a virtual cable

General Assumptions

- Questions:
  - Open:
    - None at this time.
  - Closed:
    - Will AVBTB have to do any policing or scheduling? Do we need a group to study this, or should we add this to the work of things to do in the Timing/Synchronization team?
      - *No, will be done in 802.1Qav which will handle per stream shaping. AVBTP will refer to it.*

General Assumptions

- Work Items:
  - Agreed to have initial draft for P1722 with initial agreed encapsulation details and high level outlines for other sections in time for July IEEE 802 meeting in San Francisco.
    - Alan to work on this draft.
  - Agreed to start work on annex for Interworking function between 1394/61883 and P1722/61883.
    - Alan to provide high level outline for this annex as part of the initial draft.

Timing and Synchronization

- Approved by Consensus:
  - 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 based seconds, cycles and cycle-offset time.
    - 61883-4 & 61883-7: Source Packet Header format with 0-127 seconds, 0-7999 8 kHz cycles, 0-3071 24.576 MHz cycle-offset.
    - All other 61883 encapsulations: CIP header format with 0-15 8 kHz cycles, 0-3071 24.576 MHz cycle-offset.

Timing and Synchronization

- Proposals:
  - One or more independent bridged-1394 domains (domain = set of media applications sharing a common SYT) may be supported over AVBTP, with each domain optionally syntonized to 802.1AS.
    - (Details in backup section of this presentation)
  - Presentation time assumptions/proposals:
    - The Presentation Time has the following purposes:
      - Reconstruction of the media clock
      - Account for link latency
      - Possible Synchronization of streams
      - AVBTP Presentation time is only associated with a single AVBTP stream
    - Ingress time is when the sample is sent to the AVBTP layer
      - On an I2S interface this is a 802.1AS timestamp of the word clock transition for the received sample.
    - Egress time is the Ingress time plus a delay constant

Timing and Synchronization

- Questions:
  - Open:
    - Should this team also work on queuing, policing and scheduling topics?
    - Will we work on a MIB/Management interface definition for this?
    - What will be the relation (if any) between AVBTP 61883 presentation time and the 802.1AS clock?

Timing and Synchronization

- Work Items:
  - Define what is or isn't done by transport layer (i.e. what is done by applications versus the transport layer itself)
  - Design and specify timing/synchronization methods, protocols, formats, etc.
  - Design Timing/Synchronization service interface.
  - Verify timing and synchronization methods are implementable in hardware.
    - *Michael has volunteered to work on this.*

Session Management

- Approved by Consensus:
  - Shall use LLDP(802.1AB), SRP(802.1Qat) as protocols of the Session management protocols and procedures.
  - Provide interface to Zeroconf
  - Adapt 1394 AV/C Function Control Protocol (FCP) for use in 61883 over AVBTP.
  - Architecture will support other protocols in the future besides AV/C, but we will define AV/C first.

Session management

- Approved by Consensus
  - Function Control Protocol is IN
    - AV/C will just be the first command set supported
      - Intention is to not carry 1394 bus resets (use 1394.1 model)
  - Plug Control Registers are IN
    - Some equivalent to Plug Registers for managing stream connections
  - Connection Management Procedures are IN
    - Must reflect our “Plugs”
  - Stream ID Assignment is IN if not defined in 802.1
    - Needed to complete our “Plugs” and CMP
  - IRM emulation is IN for AV/C
  - Service Discovery is IN for each command set supported
    - (i.e. AV/C will recommend Bonjour, but other protocols will be allowed).

Session Management

- Proposals:
  - Support for changing bandwidth reservations while a stream is running
    - I hesitate to say “dynamic” as this should be only an occasional thing.
    - Probably OUT for first AV/C version, but IN for planned follow on work

Session Management

- Questions
  - Open
    - Talker-Listener model vs. Talker-Controller-Listener model in 61883
      - In AVB is it possible to put all the smarts into a controller to make talkers and listeners simple?
      - Still use Controller, but talkers and listeners are somewhat smarter than on 1394
    - Are there other protocols needed at lower layers?
    - Are there other protocols we should provide a service interface to?
    - Will we work on a MIB/Management interface definition for this?
    - How is latency managed by session management?

Session Management

- Work Items:
  - Define transport
    - Protocol/procedure
    - State Machine(s)
    - Service Interface(s)
  - Look into using UDP with AV/C
    - *Matt and Andy to look into this.*
  - Define plugs
  - IRM equivalent?? Mapping of Channel ID??

Protocol layering/options

- Approved by Consensus:
  - Shall directly interface with the following protocols
    - LLDP(802.1AB)
    - SRP(802.1Qat)
    - LLC (802.2), Ethertype option only (no length/DSAP/SSAP/etc. support).
  - Shall require use in the AVBTP end station of:
    - PTP(802.1AS)
    - 802.1Qav (queuing and scheduling)
- Proposals:
  - *None at this time.*

Protocol layering/options

- Questions:
  - Will we define any interface to use PTP via a system time interface (or will all PTP time aspects be outside of the scope of the transport layer and instead part of the application layer)?
  - Will we define a client interface for the AVBTP media transport layer?
- Work Items:
  - Ensure all service interfaces are defined for all layers needed by AVBTP either in the AVBTP document or in other documents.

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

- 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)
  - When fragmenting from large CIP packets on 1394 nets to AVBTP/802 nets.
    - Option 1: fragment on "event boundaries" of Data Blocks or 61883-4/7 Source Packets.
      - Editor's proposal in http://www.avbtp.org/contributions/avbtp-bartky-encapsulation-v0-02-2007-06-27.pdf using currently defined format and adding first fragment and last fragment indication bits in AVBTP "Other bits" field.
      - Editor's note: Does not handle IIDC packets as they don't have CIP headers.
    - Option 2: Create new AVBTP header fields & protocol to do arbitrary fragmentation and reassembly of CIP/IIDC packets
    - >>Comment from team: Probably will also have to reassemble them if going back to 1394 in the same packet.

Encapsulation Assumptions

- Proposals:
  - Create “bare bones” AVBTP transport for 61883
    - (details in backup section)
  - A distinct AVBTP packet type is defined for cross-timestamps
    - (details in backup section).

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
      - » First cut proposal in: http://www.avbtp.org/contributions/avbtp-bartky-encapsulation-v0-02-2007-06-27.pdf

Backup

C. Harrison proposal details

- Proposal
  - One or more independent bridged-1394 domains (domain = set of media applications sharing a common SYT) may be supported over AVBTP, with each domain optionally syntonized to 802.1AS.
- [Additional information]
  - Note 1: As a special case, there may be a single bridged 1394 domain syntonized to 802.1AS global clock: this is the "single global synchronization source" mode as discussed on a previous call. This mode is the default.
  - Note 2: Streams originating in one 1394 domain and delivered to a different domain may require timestamp conversion (typically achieved through cross-correlating with the 802.1AS global timescale)
  - In 1394 systems, SYT time is distributed thru periodic cycle start packets.

C. Harrison proposal details

- [Additional information (continued)]
  - It is proposed that in AVBTP systems SYT time be distributed within a bridged-1394 domain by periodic SYT-to-802.1AS cross-timestamp packets (this is a new AVBTP frame type, which need not, in principle, use reserved Class 4 or Class 5 bandwidth).
  - The sender and/or receiver use these cross-timestamps to maintain synchronized SYT registers in their respective devices.
  - Within each domain a single station is elected as the "cycle master" to issue these packets to other stations in the domain.

C. Harrison proposal details

- Proposal
  - Create “Bare bones” AVBTP media transport for 61883/CIP
- Additional/Requirements, we need to say:
  - (1) here is how to encapsulate a small CIP packet into an Ethernet frame
  - (2) here is how to fragment a big CIP packet into several Ethernet frames, and how to reassemble them into the original CIP
  - (3) a sender of media data behaves as follows:
    - (a) create a sequence of CIPs in accordance with an accepted part of 61883.
    - (b) process the CIP according to (1) or (2) to create Ethernet frames
    - (c) schedule transmission of the frames in accordance with the stream parameters and 802.1Qav
  - (4) a receiver of media data behaves as follows:
    - (a) decapsulate and reassemble the frames according to (1) and (2)
    - (b) process the CIP in accordance with the relevant part of 61883

C. Harrison proposal details

- Proposal
  - A distinct AVBTP packet type is defined for cross-timestamps.
- Additional information:
  - The proposal is intended to support at least the following use cases:
    - (1) A 61883 function wishes to synchronize its SYT register to a 61883 cycle master function communicating over an AVB network, using 802.1AS global clock reference.
    - (2) An arbitrary media application wishes to synchronize its media clock with a media clock master function communicating over an AVB network. This function is intended for non-61883 applications and may also be used in combination with 61883 -- e.g. to obtain timing granularity finer than 24.576 MHz ticks.

C. Harrison proposal details

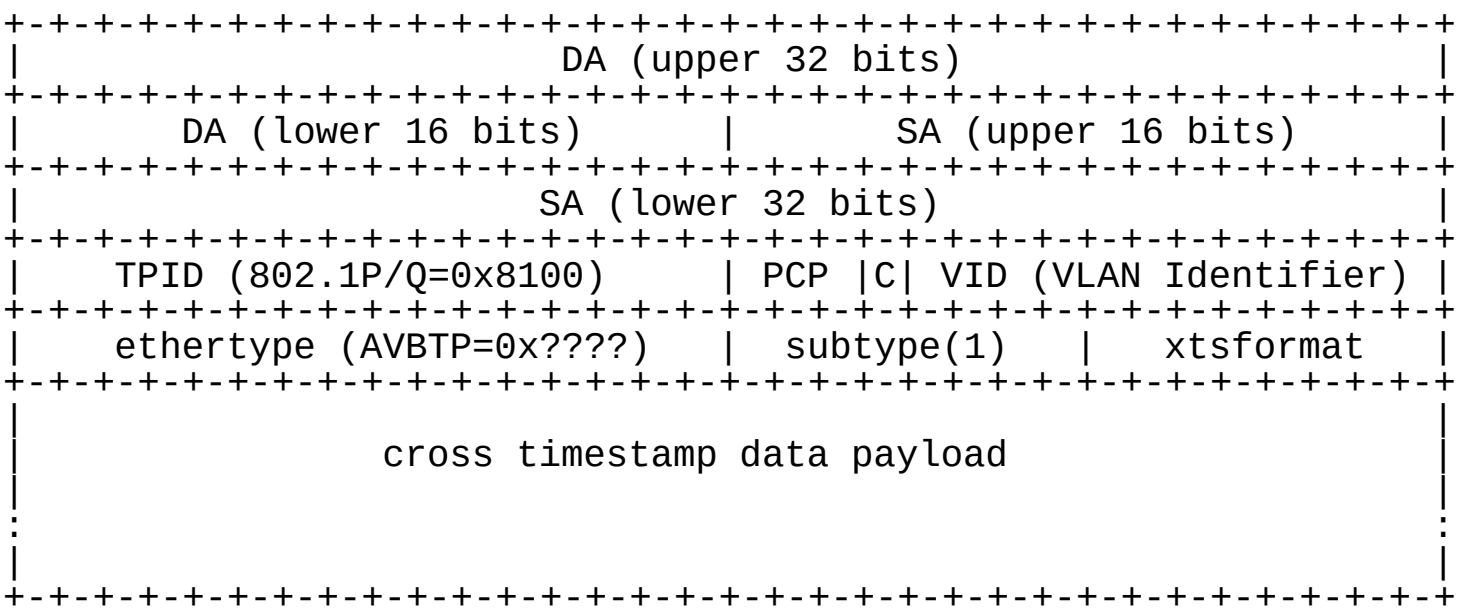
- Additional information (continued):
  - The sync-slave functions subscribe to the sync-master service through a session management protocol.
  - The sync-master service may be co-located with a talker, a listener, or a controller.
  - The cross-timestamp (XST) packets may be distributed by unicast or multicast MAC addresses.
  - The XST packets will typically be sent by best-effort Ethernet service but may be sent over reserved bandwidth

C. Harrison proposal details

- Additional information (continued):
  - In order to support situations in which an application involves several media clocks there is provision for time-base (TB) identifiers.
    - The scope of these identifiers is within a session containing a single sync-master MAC address and a single sync-slave MAC address.
    - TB identifier zero is reserved as 802.1AS global time.
  - In order to support rapid settling of media clocks at the beginning of a streaming session, the protocol provides an optional high-precision rate field
  - In order to efficiently support use case (1) above (SYT register synchronization) two special packet formats are defined
    - One with full resolution timestamps
    - One with reduced-length timestamps for periodic updating.

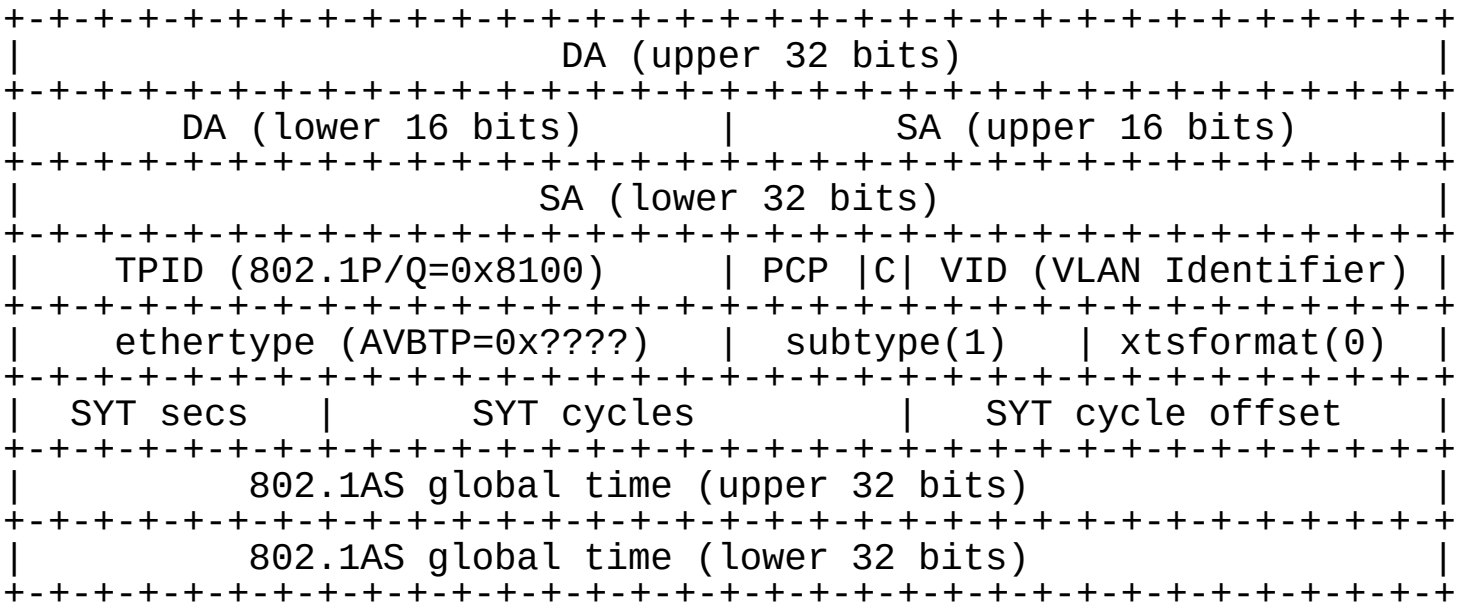
C. Harrison Proposal details

- Proposed AVBTP cross-timestamp (XTS) packet header (AVBTP subtype = 1)



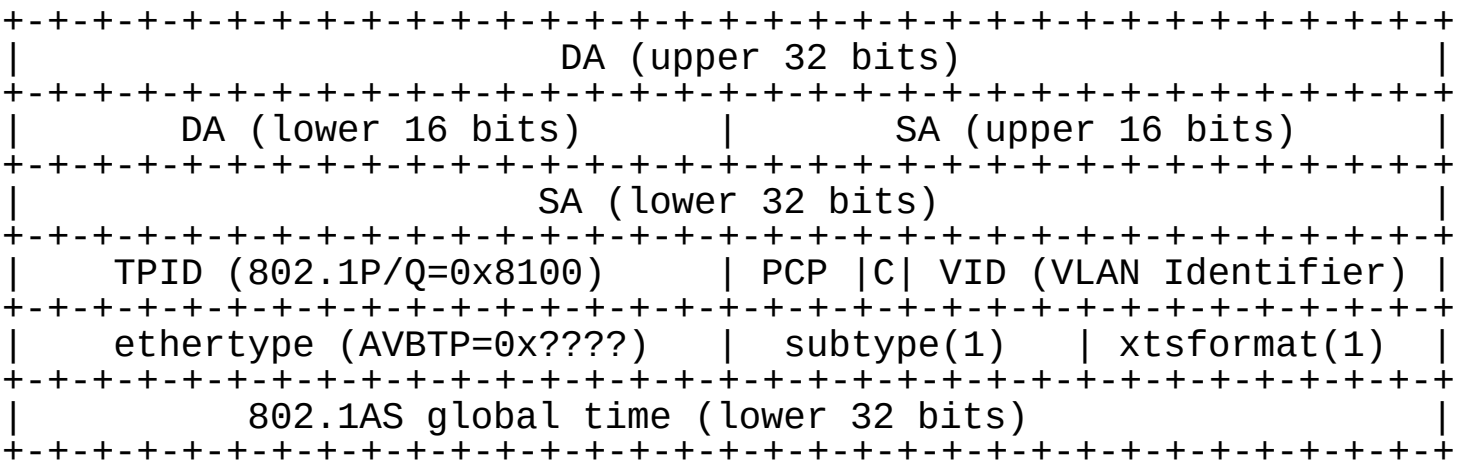
C. Harrison Proposal details

- Proposed AVBTP cross-timestamp (XTS) packet: full SYT format



C. Harrison Proposal details

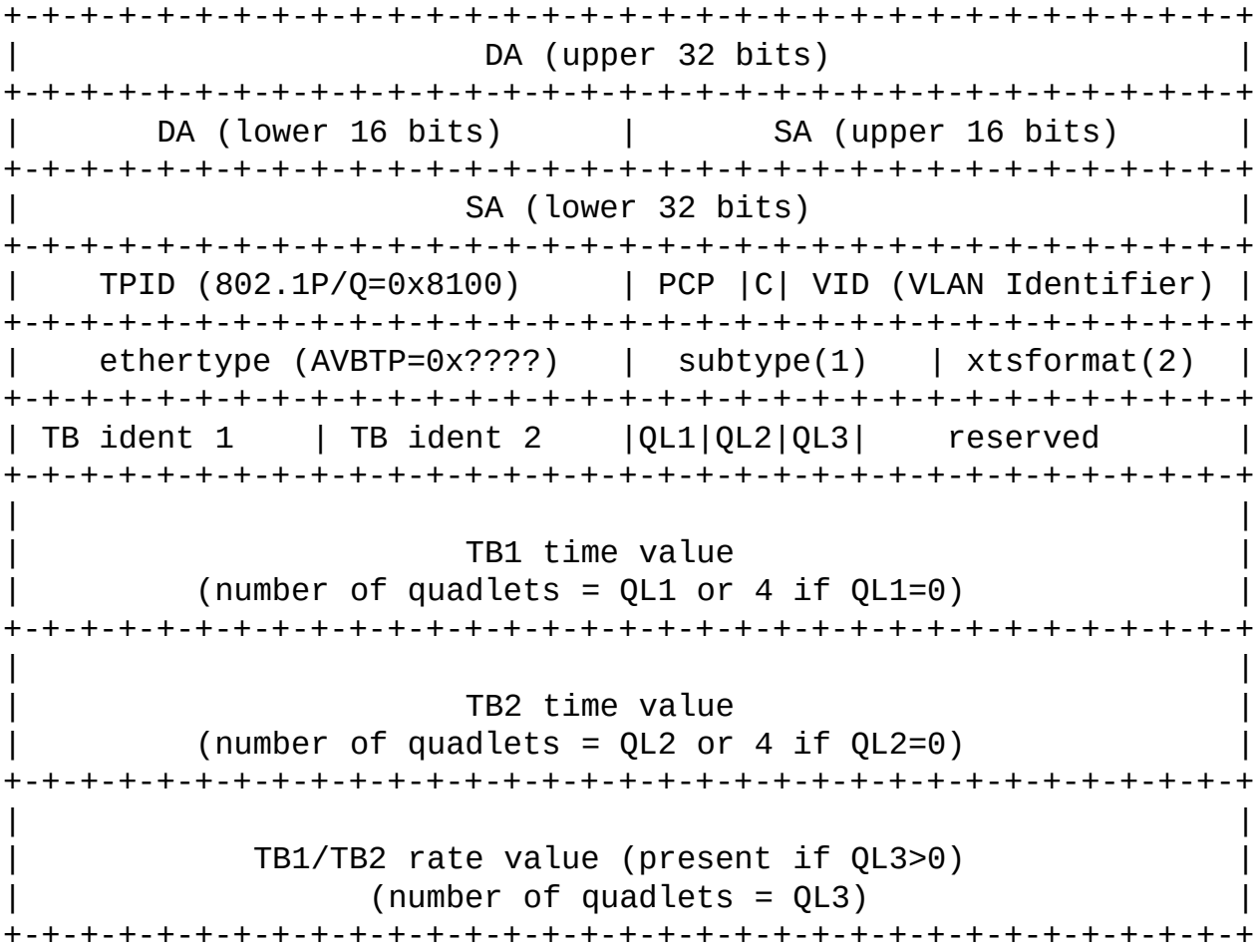
- Proposed AVBTP cross-timestamp (XTS) packet: SYT update format



global timestamp applies to an integer millisecond cycle count (i.e. cycle offset = 0, (cycle_count) mod 8 = 0) within 0.25 second of frame transmission time.

C. Harrison Proposal details

- Proposed AVBTP cross-timestamp (XTS) packet: generic format

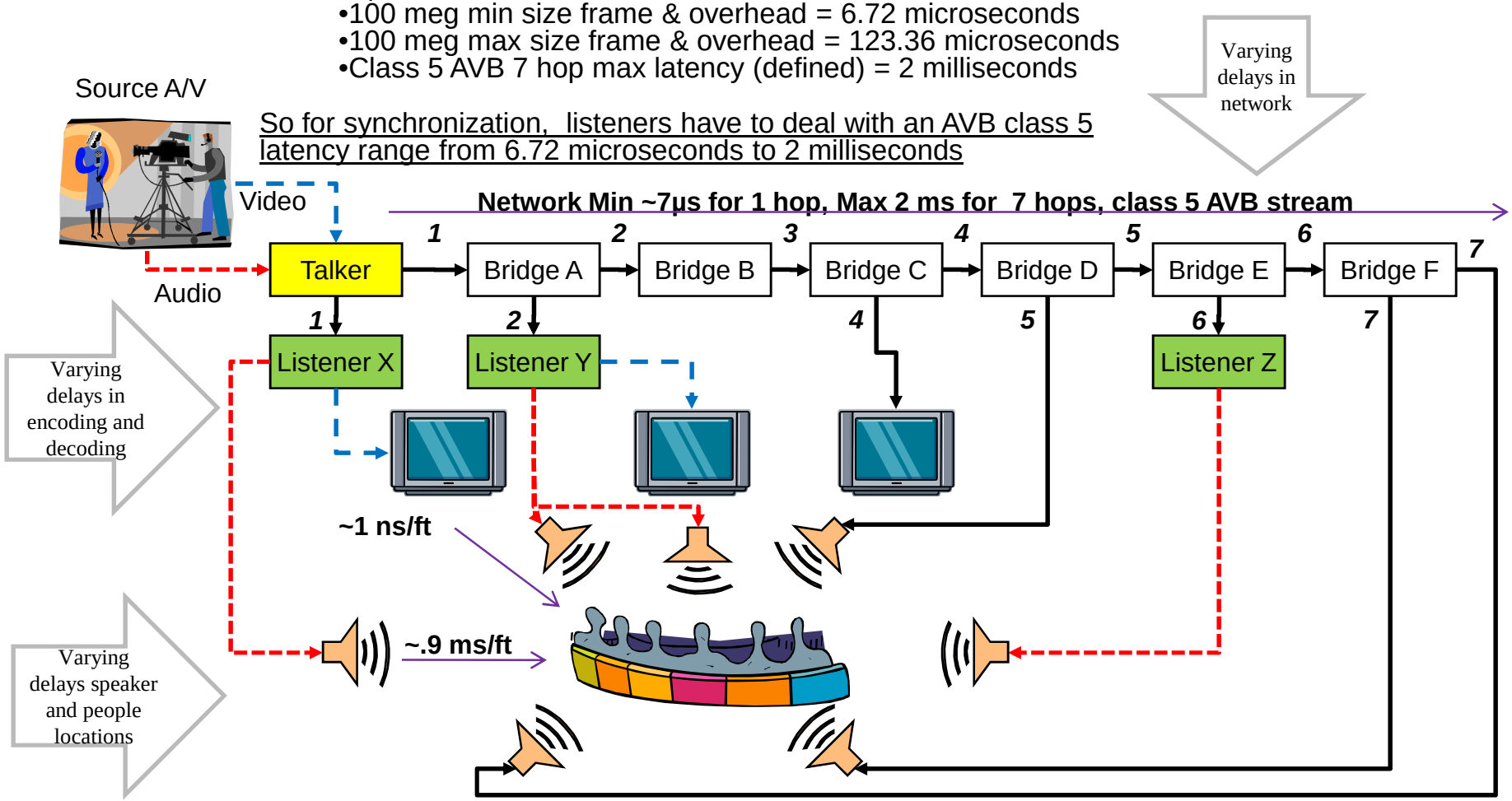


Mixed bridged and P2P AVB

Misc time info:

- Speed of light = ~ 1 nanosecond / foot
- Speed of sound = ~ .9 milliseconds / foot
- 100 meg min size frame & overhead = 6.72 microseconds
- 100 meg max size frame & overhead = 123.36 microseconds
- Class 5 AVB 7 hop max latency (defined) = 2 milliseconds

So for synchronization, listeners have to deal with an AVB class 5 latency range from 6.72 microseconds to 2 milliseconds



AV end station layering