P1722 Presentation Time

Craig Gunther
(cgunther@harman.com)
October, 2007

P1722 Presentation Time

Topics

- Recommendations
- Existing PT Definition
- Proposed PT Definition
- 61883-6 Audio Format (example)
- P1722 A/V Network Requirements
- 802.1AS GrandMaster Changes
- 1394/AVB Gateway
- Unanswered Questions

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 2

I will be using audio streams (61883-6) as an example stream format for this presentation.

Recommendations

1. Redefine avbtp_timestamp as nanosecond based presentation time (0-4.3 second range)
2. Utilize existing SYT_INTERVAL on 61883-6 audio packets
3. By default, Class A Presentation Time Offset will be 2ms from ingress time
4. When Listener notices a large Presentation Time mismatch it should free-run for several packets (this is a GrandMaster change)
5. When Presentation Time mismatch is small Listener should adjust frequency
6. Talker’s cannot change the Presentation Time Offset of a running stream

Zero fill the SYT field. Also, strip the Source Packet Header if it comes in from a 1394-to-AVB gateway/portal.

Make sure we understand frequency changes vs wall time changes

Existing PT Definition

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 4

First let's talk about what is currently defined in P1722 and what I perceive as problems.

Existing PT Definition

(16883 SYT-based Presentation Time)

- 16-bit SYT on Part 2, 3, 5 & 6
  - 2ms maximum Presentation Time Offset
  - Class B traffic allows 10ms over 7 hops (1.4ms/hop)
    - 2ms only works for 2 hops
- 24.576MHz Cycle Time clock
  - ~40.7ns accuracy
  - Clock conversions imply jitter
- Cycle Master
  - Every node must now be Cycle Master capable
  - Must define negotiation process
    - Another PTP-like problem to solve
    - What if a better one comes along?
    - What if current one disconnects?
- Requires periodic CYCLE_START packet
  - Are there latency issues to address here?
  - Cross timestamp approach still requires some type of Cycle Master
- Only applicable to 16883-based encapsulation
  - Do we really need to develop a new scheme for every encapsulation

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 5

What I don't like about 61883 SYT based timing:

•2ms limitation – see next slide for more details

•Cycle Master (somebody has to beat the drum)

•Cycle Start or XTS periodic packets

  •Are there latency issues?

  •Negotiating who is master

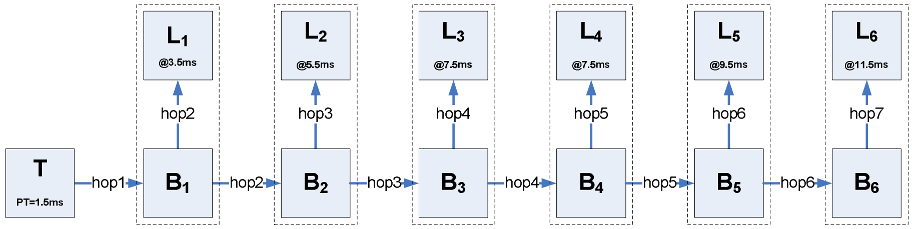
Is it really worth the overhead of creating the 8kHz cycle with associated sub cycles that only give 40.7ns accuracy?

What if PTP Grand Master is also SYT Cycle Master? When GM goes away how long will it take to get a new clean Cycle Time going?

We could redefine SYT format (nanosecond instead of Cycle Time) – This requires a double conversion for 1394-to-AVB-to-1394.

Existing PT Definition

(Problems with 16-bit/2ms Presentation Time)



- Assume Class 5 stream - 10ms for 7 hops (1.4ms per hop)
- Assume 16-bit Presentation timestamp is set to 1.5ms
  - 16-bits allows a maximum presentation time of 2ms, then wrapping occurs
  - 1.5ms will wrap and match at 3.5ms, 5.5ms, 7.5ms, etc
- Packet arrival times
  - L_1 arrival time @ 2.8ms, 16-bit timestamp will wrap and match at 3.5ms, then be played
  - L_2 arrival time @ 4.2ms, 16-bit timestamp will wrap twice and match at 5.5ms, then be played
  - L_3 arrival time @ 5.6ms, 16-bit timestamp will wrap three times and match at 7.5ms, then be played
  - L_4 arrival time @ 7.0ms, 16-bit timestamp will wrap three times and match at 7.5ms, then be played
- Since 16-bits of timestamp accuracy does not allow a node to determine if the packet has arrived late, all nodes will play the packet and the audio will sound very strange indeed

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 6

I don’t really want to spend a lot of time on this slide, other than to say it shows the problems associated with a 16-bit (2ms) Presentation Time.

Note that the dashed outlines gives an example of daisy-chained devices.

Proposed PT Definition

Proposed PT Definition

- P1722 currently defines avbtp_timestamp (32-bit Source Timestamp) and 61883 SYT field (16-bit and 25-bit)
- Change avbtp_timestamp to be Presentation Time
  - 32-bit field with nanosecond accuracy
  - 0 to 4.3 second range
  - No clock conversions
- Ignore SYT field
  - Not used by AVB nodes
  - 1394/AVB gateways
    - Leave as-is (handy for 1394-to-AVB-to-1394)
    - Also convert to/from avbtp_timestamp AVB Presentation Time

61883-6 Audio Format (example)

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 9

I will use the 61883-6 Audio/Music Data Transmission Protocol as an example to explain the proposed P1722 Presentation Time approach.

61883-6 Audio Format

(AM824 Multi-bit linear audio format)

- Based on 61883 Part 6:

  Audio and music data transmission protocol
- AM824 Multi-bit linear audio format (clause 8.2.3)
  - 32-bit data
    - 8-bit Label (value = 0x40)
      - ASI1 = 00 (Raw audio from/to CODEC)
      - ASI2 = 00 (24-bit)
    - 24-bit audio sample
- Presentation time stamp every 8^{th} sample implies SYT_INTERVAL=8 (Table 22)

61883-6 Audio Format

(CIP header details)

EOH	form	SID	DBS	FN	QPC	SPH	rsv	DBC
0	0	1 1 1 1 1 1		0 0	0 0 0	0		

EOH	form	FMT	FDF	SYT (16-bit SPH=0)
1	0	0 1 0 0 0 0	0 0 EVT N S F C	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

- SID = 63 (specifies that originating source is AVB)
- DBS = 1 * (# of channels)
- FN = 00b (no fragments)
- QPC = 000b (no FN, therefore no padding)
- SPH = 1b (using Source Packet Header, i.e. 25-bit SYT field)
- rsv = 00b
- DBC = Zero based, monotonically increasing, block number of first Data Block in the packet **(unique per packet)**
- FMT = 0x10 (61883-6 Audio & Music)
- FDF = 0x02
  - FDF.EVT = 00b (Basic AM824 encoding)
  - FDF.N = 0 (Use default SFC table)
  - FDF.SFC = 010b (48kHz, SYT_INTERVAL=8)
- SYT = 0x00 (61883 Presentation Time will be ignored by AVB)

Green highlight implies unique values per packet. All other values remain constant from packet-to-packet.

61883-6 Audio Format

(48kHz audio sampling rate example)

- A Class A isochronous packet is generated every 8kHz
- CIP Header DBS considerations
  - Number of datablocks, but not the size, can change per packet
    - 44.1kHz stereo audio (DBS=2, L+R)
      - 5-6-5-6-5-6-5-6 samples
      - DBS = 10-12-10-12-10-12
    - 48kHz stereo audio (DBS=2, L+R)
      - Some “accordion” action: 6-6-6-5-6-6-7-6-6
      - DBS = 12-12-12-10-12-12-14-12-12
  - Note that DBS is predefined in everything but 61883-6
- Assume 48kHz audio sampling rate
  - 6 samples every 8kHz
  - Single timestamp in each 61883-6 packet
  - Which of the 6 samples does the timestamp relate to?
  - How is the timestamp interpreted?

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 12

Timestamping first sample is okay as long as you have no underlying clock problems (e.g. GM change). Timestamping every eighth sample allows much easier frequencey adjustmnet calculations.

61883-6 Audio Format
(Timestamp Index)

- The SYT_INTERVAL for 48kHz audio specifies that every 8th sample is timestamped (e.g. sample 1,9,17,etc)
  $S_1S_2S_3S_4S_5S_6$ $S_7S_8S_9S_{10}S_{11}S_{12}$ $S_{13}S_{14}S_{15}S_{16}S_{17}S_{18}$ $S_{19}S_{20}S_{21}S_{22}S_{23}S_{24}$ $S_{25}S_{26}S_{27}S_{28}S_{29}S_{30}$
- Therefore the 1st packet timestamp is associated with the 1st sample in that packet
- The 2nd packet timestamp is associated with the 3rd sample in that packet
- The 3rd packet timestamp is associated with the 5th sample in that packet
- The 4th packet timestamp is not associated with any samples in that packet
  - Note: P1722 can mark this timestamp as invalid (*tv* bit)
- The 5th packet timestamp is associated with the 1st sample in that packet

61883-6 Audio Format
(Timestamp Index)

- How can a device know which sample is timestamped in the current packet?
  - SYT_INTERVAL is encoded in FDF.SFC bits
  - DBC identifies the Data Block Count of the first Data Block in the packet
  - Calculate the 0-based Sample Index like this:

    mod((SYT_INTERVAL – mod(DBC, SYT_INTERVAL)), SYT_INTERVAL)

- SYT_INTERVALs (8,16,32) are designed to make it easy to calculate time for a single sample

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 14

SYT_INTERVAL calculations:

(CurrentPresentationTime – PreviousPresentationTime) / 8 = (CPT – PPT) >> 3

SYT_INTERVALS are purposely defined as binary numbers that make it easy to divide (8,16,32) to derive single sample times.

61883 Timestamp Intervals
(Sample Index)

Packet #	DBC	X = mod(DBC,SYT_INTERVAL)	Sample Index mod((SYT_INTERVAL-X),SYT_INTERVAL
1	0	0	0
2	6	6	2
3	12	4	4
4	18	2	**6**
5	24	0	0
6	30	6	2

The table above shows an example of the Sample Index calculations. Note in packet #4 the Sample Index is 6, but the samples only run from 0 to 5. This means the timestamp in packet #4 does not apply to any of the samples in that packet. In fact, it is the same timestamp that will be reported in packet #5.

P1722 A/V Network Requirements

P1722 A/V Network Requirements

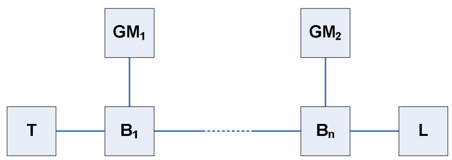
- AVB defines low latency Class A traffic with a maximum latency of 2ms over 7 hops
- AVB defines higher latency Class B traffic with a maximum latency of 10ms over 7 hops
- Listener should be capable of synchronized Class A delivery of audio data anywhere from 1 hop to 7 hops away (i.e. nearest node may have to buffer 2ms of audio streaming data)
- By default, Presentation Time Offset will be 2ms from ingress time

Adding a default presentation time will make it easy to build Consumer grade gear that will sound right! Very handy when sending video to a screen and audio to a set of 7.1 speakers that might be hooked in a daisy chain (lots of hops).

802.1AS GrandMaster Changes

802.1AS GrandMaster Changes
(Introduction)

- Consider Presentation Time as a “highly preferred” time, but not a “mandatory” time
- 61883 DBC is much more useful than just a monotonically increasing number
- GrandMaster change can affect Talker and Listener in different order:



The designer’s of 61883 have actually put some effort into the design of the protocol – let’s use it! There are subtle niceties behind DBC and SYT_INTERVAL.

Talker’s and Listener’s can make no guesses about what is happening with a GM change when Presentation Times vary. The GM change could be forward or backward in time. The Listener could see the new GM before the Talker, or vice versa. That means a Listener can make no valid assumptions about Presentation Time mismatches.

802.1AS GrandMaster Changes
(Scenario)

- Listener media clock PLL will be locked to current PTP GM via DBC and Presentation Time stamps (PTS) in packet
- Listener could calculate an average sample rate using DBC and PTS
- During GM change Listener will see PTS that don't match what it computes via DBC and the calculated average sample rate
  - There may also be a notification from PTP layer (will be at least 3 seconds late!)
- Listener should free-run for several packets (length of time relates to AS settling time)
- Eventually GM change will reach both Talker and Listener, and "correctly" time stamped packets will arrive
- When PTS is "close enough" to Listener PTP time the Listener can start fine tuning PLL frequency (Listener should not make any assumptions about frequency until this time)
- The previous statement implies that Talker cannot change Presentation Time Offset of an existing stream
- Missing DBC sequences will signify lost samples and can be handled appropriately without affecting frequency

Listener should not make assumptions about what the Talker and the network are doing. There are examples where a Listener can be too smart and start trying to adjust for Presentation Time variations to quickly; the result is that it makes changes in exactly the wrong direction. For example, if the Talker's Presentation Times start to stretch out does that mean the stream sample rate has actually slowed down? Or, does it mean the clock the Talker is listening to is faster than the clock the Listener is listening to? This is why the Listener needs to wait a certain amount of time to make sure the underlying PTP clock is stable.

Delta between Packet Presentation Time and Listener calculated Presentation Time:

•if Delta > X then freerun and wait for PTP synchronization

•if Delta < X then frequency adjust

•if Delta > X for some long time then we need to handle a special event. What could cause this?

•If we also passed Presentation Time Offset we might be able to allow the Talkers to change it on the fly. These problems should rarely happen. How much packet overhead is worth the saved calculations?

1394/AVB Gateway

1394/AVB Gateway
(1394-to-AVB)

- Convert SYT field to AVB Presentation Time
- Leave SYT field intact – AVB ignores it
- Exchange cross-timestamp packets with other 1394/AVB Gateways
- *Could* strip the 32-bit SPH to save a quadlet
  - Not really worth while
  - Would introduce jitter on 1394-to-AVB-to-1394
  - AVB Listener ignores SYT field

Push SYT Cycle Time knowledge/processing into 1394/AVB gateways, not into the individual Listeners.

1394/AVB Gateway
(AVB-to-1394)

- If SID=63 (AVB Talker)
  - Convert AVB Presentation Time to SYT field
  - Possible problems with 2ms SYT field on Part 2, 3, 5 & 6
- Exchange cross-timestamp packets with other 1394/AVB Gateways
- *Possibly* recreate SPH if 1394-to-AVB gateway stripped it when putting 1394 packet onto the AVB network
  - If SID <> 63
  - And SPH = 1
- Larger range of AVB Presentation Time Offsets could require buffering in gateway

Possible conversion problems since AVB allows 0-4.3s range, 16-bit SYT allows 0-2ms range, 25-bit SYT allows 0-1s range. May require AVB-to-1394 to have buffering.

Unanswered Questions

Unanswered Questions

1. What about playing samples if presentation time has already passed?
   - What if it only happens once?
   - What if it consistently happens?
   - Consumer only?
   - Visual indication if samples are discarded?
2. What about presentation time that is so far in the future that the node can’t buffer it?

18 October 2007 (v2) (with modifications from 18Oct07 meeting) 25

1. Think about consumer gear that is on hop to far away. If it discards the packets then no sound would come out. Consumers might interpret this as a bad speaker or bad bridge. I can imagine a scenario where a consumer buys a new AVB speaker (and bridge) so they can listen to the TV news in an exercise room next to the family room. They plug in the bridge and speaker and get no sound because the network latency is to great to support the Presentation Time. The consumer assumes the speaker is bad and exchanges it for a new one. The new one doesn’t work either, so the consumer exchanges the bridge for a new one. Still no luck, so the cable must be bad. Nope, swapping the cable doesn’t solve the problem. Now what? Maybe the existing bridge I plugged into to add me new speaker was bad.

MJT feels that late samples should NOT be played. If we do this then we definitely need some type of notification that is obvious to an unskilled consumer.

Thank you