How Many Stream Time Stamps are Required?

IEEE 1722a Meeting

Aug 2013 – San Jose, CA

Don Pannell
Marvell Semiconductor
dpannell@marvell.com

Overview

- Draft 5 Page 33 Line 25 states (section 8.2.8):
  - "All AVTP Audio format AVTPDU's shall set **tv** (avtp_timestamp_valid) field to a one (1) and contain valid data in the **avtp_timestamp** field."
- While this appears to be a simplification of designs (i.e., there are no options), this isn't necessarily "simpler" to build cost effectively:
  - Old Section 6 48KHz Audio generates 6000 **ts**/sec
  - New Section 8 48KHz Audio generates 8000 **ts**/sec!
  - Can be 192000 **ts**/sec for 48 KHz 1 sample/frame!!!
- How many **ts**/sec do we really need?

Goal

- Require enough **ts**/sec (time stamps per second) for the target use cases and markets
- Propose to change Draft 5 Page 33 Line 25 to be:
  - "At a maximum time spacing of ??? uSec AVTP Audio format AVTPDU's shall set **tv** (avtp_timestamp_valid) field to a one (1) and contain valid data in the **avtp_timestamp** field.
- What time spacing amount is needed?
  - A 1000 frame/sec stream can't do better than a time spacing of 1 mSec between valid **ts**'s (and that is with every **ts** being valid!)

Reality

- In AVTP the avtp_timestamp is compared to IEEE 802.1AS time in the device to re-create the stream's media clock (if a Clock Reference Stream) is not be used.
- IEEE 802.1AS time is updated by its protocol no faster than 8 times a second
- To stay locked, we really only need 16 **ts**/sec or 1 every 62.5 mSec!
  - 2x the IEEE 802.1AS update rate is used due to Nyquist
- But there are other use cases that require more:

Use Case 1

- Connecting to a stream already in process
  - Assume Listener is already locked to 802.1AS time
  - Before stream connection the Listener already knows the target media frequency from SRP – so its PLL/VCO is powered up and set to this target
  - As the stream starts being received by the Listener, its needs to lock/adjust its media clock to the stream's **ts**'s
  - This is like the case shown in Media Clock Stabilization Time – Part 2 by Czekaj as it's a PPM shift
    - http://grouper.ieee.org/groups/1722/contributions/2013/Media%20Clock%20Stabilization%20Part%20Two_one.pdf
  - This fine tuning process can't start until 2 **ts** are valid

Use Case 2

- Starting a stream from a dead start
  - Listener may or may not be locked to 802.1AS time
  - Before the stream is sent to the Listener it already knows the target media frequency from SRP – but its PLL/VCO is powered down saving power!
  - As the stream starts being received by the Listener, its needs to power up its media clock & lock to the stream's **ts**'s
  - This is worse than the previous case as the media PLL was powered down
  - But the power up of the PLL can start on the 1st received frame of the stream even if the **ts** is not valid!

Physics

- Adjusting a clock takes time (PPM adjustments) & needs a minimum of 2 **ts**'s to start
  - Clocks may take typically 10 to 20 uSec to adjust
- Starting a clock takes more time (powering it up) & and needs a "power-up" indication (i.e., receiving the 1st frame, **ts** valid or not) and then it performs the Adjusting a clock function above
  - Clocks may take typically 50 to 100 uSec
- These times are not going to effect the number of **ts** that are needed by much at all
- But if an AMP is powered down that will take longer

Human Perception

- How much time can elapse before something is ‘perceived’ to not be instantaneous?
  - Video industry has used 500mSec to change channels
  - Audio industry has used 10mSec to play an instrument
  - Switch de-bounce requirements is in the 1-20mSec range (time delay before a ‘button’ is registered)

Our Current Worst Case

- At our longest 1000 uSec FQTSS Observation Interval we get no more than 1000 **ts**/sec or 1 **ts** every 1 mSec.
  - In this case it takes 2 mSec worst cast to get two valid **ts**'s (assuming all frames have **ts**'s)
  - Then the 20 uSec time to adjust the PLL
  - This is within the perception of instantaneous for Audio
- But our longest Transmit Interval is 1088 uSec so this is 919 **ts**/sec or 1 **ts** every 1.088 mSec
  - This is for 44.1 kHz & its derivatives
  - This works out to be 2.176 mSec to get 2 **ts**'s

Proposal

- Propose to change Draft 5 Page 33 Line 25 to be:
  - "At a maximum time spacing of 1.1 mSec AVTP Audio format AVTPDU's shall set **tv** (avtp_timestamp_valid) field to a one (1) and contain valid data in the **avtp_timestamp** field.
- This works with all FQTSS Observation Intervals
  - 125 uSec to 1000 uSec
- An implementation can still choose to provide **ts**'s on every frame – but doing so will not provide for any added perceptible performance as everything locks within 2.2 mSec

Summary

- AVB started out as Residential Ethernet
- We don't want to prevent the ability to create low cost consumer/automotive AVB solutions while at the same time we want to support high end professional solutions
- Requiring 8000 valid **ts**/sec for 48 KHz audio is 33% more **ts** overhead than IEEE 1722
- The proposed 919 **ts**/sec is 14.7 oversampled vs. what is needed to keep the clocks locked
- The proposed solution works for all applications:
  - Consumer, Automotive and Professional