IEEE P1722 Synchronization Assumptions

Boilerplate Draft PAR version 0.01

June 20, 2007

Editor: Matthew Xavier Mora, Apple, Inc. mxmora@apple.com

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
0.0	2007-006-19	First version for comments

Goals Review

- Low implementation Cost
  - Low implementation cost
    - but not at the expense of quality
  - Need to span the full range of inexpensive consumer products to high quality professional products
- Plug and Play
  - DVD/PVR into TV
  - Speakers into AVR
  - Audio interface into Personal Computer
  - Digital snake master unit to slave unit
- Auto Configuration (where appropriate)
  - Simple (DVD -> AVR)
  - Complex (multple speakers -> AVR)
  - Live console to Line Array

What is AVB?

- The charter of the TG is to provide the specifications that will allow time-synchronized low latency streaming services through 802 networks.
- We already have high latency Ethernet that aims to reduce jitter by implementing "smoothing" buffers. IMO, we (AVBTP) are implementing a virtual wire between CODECs. It's "virtual" because there are AVB TG defined 802 services that take the place of the wire. One of the warts on IEC61883 is the lack of layering (one can't tell where 61883 stops and 1394 starts). We should strive to not repeat that by confining application-specific problems to application layers. We, AVBTP, should interface to the 802-defined services at our bottom and to higher layer(s) at the top. Simple, constrained, & elegant. -Robert Boatright

What is the target audience?

- **Professional use**
  - **Audio**
    - Live sound
    - Recording studio
    - Broadcast
    - Installations
  - **Video**
    - ???
- **Residential Use**
  - Home theater
  - MPEG stream to TV or AVR
  - Whole house audio
    - Static configuration (address selector set when installed)
    - Dynamic zone configuration
    - Outdoor

Pro Audio Use case

- Recording/Production
  - Audio interfaces
  - Recording desk
  - Outboard DSP Processor
  - Digital Mics
- Live Sound
  - FOH / Monitor desk
  - Amp Rack
  - Speakers
  - Outboard DSP processing
  - Recording feed to DAW
  - Digital Mics
- Broadcast
  - Desk
  - Speakers
- Installations
  - Conference rooms
  - Ball parks

Residential Use case

- Home Theater
  - MPEG stream to TV or AVR
  - PVR to AVR or TV
  - TV direct to Media server
- Whole house audio
  - Multi-zone
    - Static configuration (address selector set when installed)
    - Dynamic zone configuration
    - Outdoor

Device Logic requirement

- Smart devices
  - mLAN A model
    - Full connection management logic
    - Full device discovery and enumeration logic
    - Full stream negotiation and construction
- Dumb Devices
  - AVC Model
    - Requires a CPU to do the connection logic

Stream Sync

- Aggregated stream scenario
  - 5.1 speaker model
    - Master sends one stream with aggregated channel count
    - Each device of the clock domain receives the stream
    - Device is told what channel to render
    - All device start at the same time
- Individual stream per channel scenario
  - Line array model
    - Each device receives its own steam
    - Device renders the stream received no knowledge of other streams
    - All devices start at a different time

What level of Sync is needed?

- Pro/Live audio
  - Sample accurate
    - ~ 22µs @ 44.1khz
    - ~10µs @ 96khz
    - ~ 5µs @ 192khz
- Pro video
  - Video frame sync
    - ~16ms
- Home theater
  - Lip sync
    - ~30ms ?

Synchronization Scope

- This document will define the scope of the synchronization services for IEEE P1722 transport layer.
- May not cover Media application specific synchronization (lip-sync)
- Defines the low layer services that media applications will need to achieve synchronization at the application layer.

Required services

P1722 Synchronization relies on a number of AVB services to make cross device synchronization possible.

- P802.1as - clock synchronization.
- P802.1Qat - stream reservation
- What are the other services?

Definitions

- Media Application
  - Media application (MA) is a process that packetizes and depacketizes the digital media into the P1722 packet format.
    - Multiple Applications can be on one device
    - Which application to use will be determined at stream start up time
    - Listener device needs to have the same application as the talker device to decode the stream
- Presentation time
  - Time relative to the P1722 packet time stamp in the packet that the MA can use for synchronization.
  - Presentation time is MA specific
- Device level synchronization
  - Device level synchronization is when a media sample (audio or video frame) will be presented by the media application to the next interface at the same time relative to Global 802.1as time.
- Global 802.1as Time
  - A clock that is shared across all devices of a AVB cloud. Devices must be locked to the Global 802.1as time to be able to send or receive P1722 streams.

.

Media Applications

- P1722 packet is like a base class for sending digital media streams. It does not define how to send or receive audio or video, it only specifies how the packets should formatted on the wire, and what the fields of the packet are used for.
- Media applications are what format digital media streams into P1722 packets that can be sent through reserved steams in an AVB cloud.
- Common Media application will be defined for audio and video streams.
- Custom media applications can be created and still be considered P1722 compliant.
- 61883 and RTP are media applications that will be defined in other documents.

Synchronization Disruptions

- Loss of Grand master?
- Loss of media clock
  - External word clock loss
  - Digital stream loss (SPDIF gets unplugged)
  - Source unplugged
- Equivalent to 1394 bus reset?
  - Is there such a thing in AVB?
- How is redundancy handled?
  - Can AVB be looped like 1394B for redundancy?

When to Sync?

- A P1772 stream should be synchronized after stream reservation and creation?
- Stream synchronization should take less than ???ms.
- MA defines when to sync?
  - Audio would need to always send clock
  - Video only syncs when media is actually started
  - Are sync only streams allowed?

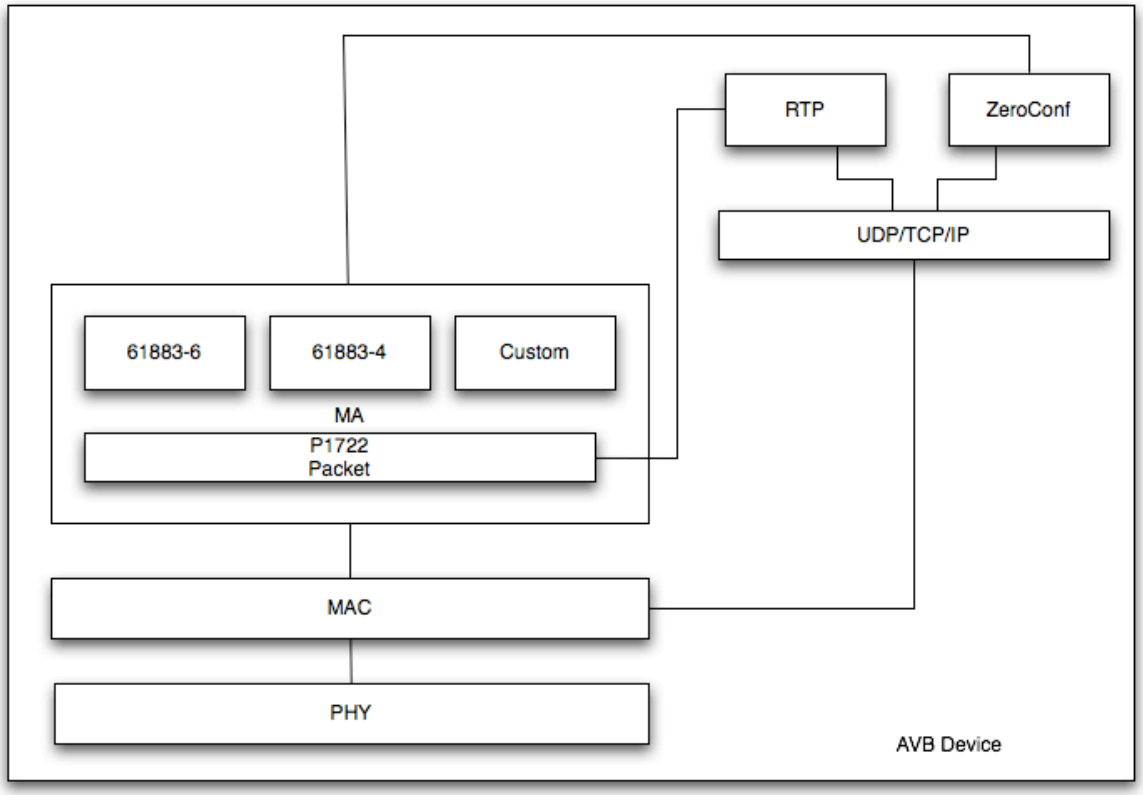
Sync Model Push or Pull?

- Matt wrote:
  - Pull model to me means that the destination asks the source to speed up or slow down sending data at the rate requested by the destination usually by separate rate control commands. In USB I believe they call this adaptive clocking. The source does not receive a "clock" from an external source, it speeds up or slows down based on commands. If no commands are received it sends data at the rate of its internal clock.
  - Push model is where the source device sends the data at a rate based on "its" clock. That clock can be an internal crystal or an external word clock. The destination has no control of the data rate, in fact the destination will likely clock off the incoming data.
- Michael wrote:
  - I'd really rather not use the terms "push" and "pull" in this discussion,since those terms have been co-opted in the application community to mean:
  - "pull" is done by HTTP "get" commands, where an application tries to keep a buffer at approximately the right level by periodically asking for more data from the source. This means that data delivery is paced by the listener in a most direct way, and all kinds of indirect implications (such as using TCP as the transport layer). This is most definitely NOT our model (even with L3+ AVB-TP).
  - "push" is done using RTP ... Where any regulation on delivery and synchronization is done explicitly via dedicated protocols such as RTCP. I think this is what we mean ... So, what the AES42 spec calls "mode 2" is not "pull", but "externally synchronized", and "mode 1" is not "push", but "asynchronous" (or, perhaps, "not externally synchronized". Perhaps we can use those terms, or just call them "mode 1" or "mode 2" like in the AES42 spec.

Problems with 61883-x sync

- Presentation time was not well defined
  - Squishy - kind of optional
    - Consumer device could ignore it
    - Pro-devices could use a smaller value
      - No way to negotiate new lower value
  - Too large for low latency
    - Had to deal with bus resets and lost cycle catch up
  - No reporting of hardware latencies
  - No support for individual plug latencies
  - For large scale network sync, vendor specific commands were needed to align streams

MA Layering


RTP
ZeroConf
UDP/TCP/IP
61883-6
61883-4
Custom
MA
P1722
Packet
MAC
PHY
AVB Device

References

- AM824 document
- 61883-1 -4 -6 document
- AVB 802.1as assumptions document
- Dave Olson Presentation time document

Time stamp resolution

- 802.1AS timestamp is in nanoseconds.
  - Current assumption is that the raw timestamping can be based on the 25Mhz clock of a 100baseTX PHY. This means that a PLL is needed for filtering for any but the most basic requirements ... Fortunately, this is pretty basic technology. For GigE links, the timestamping should be based on the 125MHz clock from the PHY ... No matter what, we need the PLLs since Ethernet is based on frame-based store and forward bridges, so there is significant jitter accumulation from bridge to bridge. Unlike 1394, however, we *knew* this was going to be a requirement, so we spent the time doing the analysis of the kind of filtering we are going to need ... See, for example, one of Geoff Garner's studies at http://www.ieee802.org/1/files/public/docs2006/as-gmg-further-simul-resul-p2p-tc-transport-060712.pdf