Audio Video Bridging and Linux

Jeff Koftinoff <jeffk@meyersound.com>
LinuxCon Vancouver - Aug 19 2011

Jeff Koftinoff

- Linux Developer since 1994
- Real time DSP for pro-audio since 1992
- MIDI Show Control systems since 1994
- Current editor of AVB standards for IEEE P1722.1 and IEEE P1722A

Meyer Sound

- Develops and manufactures very high end pro audio speakers and audio signal processing embedded devices
- Based in Berkeley, CA
- Member of the Linux Foundation
- Committed to using, supporting and contributing to open source technologies and open standards

What is AVB

- “Audio Video Bridging” is the new standard ethernet
- AVB is a collection of IEEE standards, started over 5 Years ago
- AVB allows for audio, video, and other media transport over standard ethernet with features, quality and guarantees that are not possible with legacy ethernet
- AVB replaces all other existing ad-hoc and proprietary legacy based ethernet media transport systems, all of which have limitations in quality and guarantees.

Actual AVB uses

- Professional Audio systems (Speakers, Concerts, Stadiums, Orchestras, Theme Parks, Airports)
- Replacement for Firewire for media I/O box connectivity with computers
- Consumer Audio/Video systems
- Automotive Audio/Video systems for entertainment
- Security Cameras

Future AVB uses

- Replacing/augmenting CANbus/FlexNet/MOST for automotive control systems including brakes, accelerator pedals, driver assist and locks. This allowing automotive wiring harnesses to be extremely simplified and significantly reduce total mass of a automobile
- Industrial control systems allowing ultra low latency real time control

AVB Overview

AVB is defined by the IEEE amendments and standards for network layer 2:

- IEEE 802.1AS for timing
- IEEE 802.1Q-2011 (Qat, Qav) for packet flow
- IEEE 1722 for media transport
- IEEE P1722.1 for Discovery, Enumeration, Connection management and Control (AVDECC)
- IEEE 802.1BA for putting 802.1 standards together for AVB usages
- "Talkers" send multicast streams to "Listeners" and both are controlled by "Controllers"

Timing (802.1AS)

- "Generalized Precision Time Protocol" (gptp), based on PTPv2 (IEEE 1588-2008)
- Has more accuracy and less options for implementations so it "just works"
- Requires new hardware to do accurate time stamping of packets and which is not possible in pure software
- All network ports must support 802.1AS to be part of an "AVB Cloud"

Stream Reservation (Qat)

- 802.1Qat, Stream Reservation Protocol (SRP) allows end stations to reserve bandwidth through a network for “Time Sensitive Streams”
- Bandwidth is a measurement that has a numerator (bits) and denominator (time). With AVB Class “A” streams, the denominator is 125μs
- Up to 75% of link bandwidth can be reserved for Time Sensitive Streams
- All network ports must support SRP to be part of an “AVB Cloud”

Traffic Shaping (Qav)

- 802.1Qav defines how time sensitive packets are queued and sent
- All "Talkers" and bridges must support traffic shaping
- Stream shapers and class A/B shapers
- The shapers spread out packet timing so that the maximum worst case latency over many hops and interference packets can be calculated
- Shapers are very expensive to do in software so they are usually done in hardware

IEEE 1722

- Layer 2 Media Transport (ethertype 0x22F0)
- Can encapsulate Firewire style media packets (iec61883-4,-6,-8) such as audio (fixed and float), video (compressed and uncompressed), s/pdif, MIDI, etc
- Can support other types of media
- Packets contain presentation time in nanoseconds to tell the Listener exactly which gPTP time a media sample is to be presented to a D/A

Presentation Times

- Default presentation times for class A streams is 2.0 ms. This allows media packets to go over around 7 hops at 100baseT or around 14 hops at gigabit ethernet
- Listeners that are close to a Talker then automatically add more delay to the streams than a Listener that is more hops away from the Talker
- This allows the media that comes out of a D/A to be more than sample accurate across Listeners even when they are in different locations on your network

Class B Streams

- Wireless can not provide Class A guarantees
- Default presentation times for class B streams is 50 ms.
- This allows media packets to go over around 4 hops at 100baseT and 2 hops over 802.11

IEEE P1722.1

- P1722.1 is a work in progress, currently at Draft 15A
- Discovery, Enumeration, Connection mgmt, and Control
- Layer 2 is mandatory (Ethertype 0x22F0)
- Layer 3 is optional
- Current draft is already used for a subset of use cases

Solutions

1. Standalone custom FPGA with media I/O
2. Custom silicon chip with switch fabric, embedded linux, media I/O
3. Custom FPGA as PCIe device for mainstream CPU
4. AVB enabled NIC with mainstream CPU running real time O/S

FPGA Solution

- FPGA based closed source gateware for handling time stamping, traffic shaping and media packetizing/depacketizing to/from DSP/I2C hardware
- Most devices shipping now or soon use FPGA's for assisting AVB support
- Typically run microblaze soft cpu µClinux based on:

  https://github.com/jdkoftinoff/mb-linux-msli

- GCC4 and µClinux changes are public and are being pushed upstream

FPGA Solution

Benefits:

- Very flexible for firmware, software and media I/O
- Good for medium quantity embedded devices

Drawbacks:

- FPGA hardware is typically expensive, limited and hard to develop
- FPGA solutions are closed

Silicon Chip Solution

- Conversion of FPGA solution to an ASIC to handle media packets and processing
- Typically includes CPU running Linux to handle management

Silicon Chip Solution

Benefits:

- Low parts cost
- Good for huge quantity embedded devices

Drawbacks:

- Signal flow and processing is typically limited and not as flexible
- Long lead time, massive cost when mistakes are found
- ASIC solutions are closed

FPGA + CPU solution

- FPGA handles AVB enabled ethernet, does all AVB layer 2 protocols with embedded CPU
- FPGA passes media to mainstream CPU via PCIe
- Mainstream CPU O/S typically needs to have real time extensions to allow for AVB latency requirements to be achievable

FPGA + CPU solution

Benefits:

- Flexible

Drawbacks:

- Expensive
- Hard to design
- FPGA solutions are closed

AVB NIC and mainstream CPU

- NIC provides hardware support for time stamping of incoming and outgoing messages
- NIC provides hardware support for separate priority queues that are capable of scheduling packets to be sent at a specific time in the future
- Mainstream CPU O/S needs very low latency but not necessarily real time

AVB NIC and CPU

Benefits:

- Flexible
- Low cost
- Available now (BCM57765)

Drawbacks:

- Needs Linux Kernel support of 802.1AS, 802.1Qat, 802.1Qav, 1722, iec61883-6, 1722.1, software PLL's for resynthesizing media clocks

Getting AVB into Linux

- The foundations for 802.1AS support framework is already in Linux mainline kernel since 2008, contributed by Patrick Ohly and Kevin Stanton at Intel.
- Network infrastructure needs to allow for NIC's with time sensitive packet queues with 802.1Qav traffic shaping
- Network infrastructure needs to be able to handle 802.1Qat for stream reservation

Getting AVB into Linux

- Network infrastructure needs to be able to pass ethertype 0x22F0 packets (IEEE 1722) to/from the linux audio/video system
- The linux audio/video system needs to be able to packetize/depacketize IEEE 1722 packets and present the media to the user mode applications
- The linux audio/video system needs to be able to handle most of IEEE P1722.1 packets for advertising and connection management of streams and media sources and sinks

Conclusions

- There is a lot of work needed in the Linux kernel to make it directly support new standard ethernet
- This work is both in the network infrastructure and audio/video infrastructure
- Network and audio/video subsystems need to play nice together
- There needs to be APIs developed to control, manage, monitor, and extend these functionalities
- AVB will be used for real time control, not just for media transport.