Audio over Ethernet: AVB/TSN

Florian “Florob” Zeitz

2020-11-18

→

Audio over Ethernet

- transmit digital audio over regular Ethernet
- replace (multicore) cables with a single Ethernet cable
- potential re-use of existing cabling
- flexible signal routing

AoE/AoIP

- many protocols exist
- some directly layered on Ethernet, some on IP
- AoIP
  - Dante: proprietary (Audinate), market leader
  - Q-LAN: proprietary (QSC)
  - AES67: open standard (AES), not a full solution
  - RAVENNA: open standard, solution based on AES67
- AoE
  - CobraNet: proprietary (Cirrus Logic)
  - EtherSound: proprietary (Digigram)
  - AVB: open standard (IEEE)

AVB/TSN

- **A**udio **V**ideo **B**ridging
- relatively old standard (2011)
- **T**ime **S**ensitive **N**etworking
- set of IEEE standards for AoE/industrial networks
- requires "AVB-compatible" bridges

Milan

- profile/extension of AVB for pro audio
- specified by the Avnu Alliance
- specifies a common AVB subset to support
- defines method for reliable reconnection
- adds protocol extensions for redundancy

Goals

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play
- low and constant latency
- synchronized (word) clock
- low jitter
- synchronized playout
- reliable delivery
- hard real-time constraints

Terminology

- each device is an AVB *Entity*
- three types of entities:
  - *Talker*
  - *Listener*
  - *Controller*
- not mutually exclusive

AVDECC

- **AV**B **D**iscovery, **E**numeration, **C**onnection Management and **C**ontrol
- IEEE 1722.1
- comprised of a variety of sub-protocols:

  ADP **A**VDECC **D**iscovery **P**rotocol
  ACMP **A**VDECC **C**onnection **M**anagement **P**rotocol
  AECP **A**VDECC **E**numeration and **C**ontrol **P**rotocol
  AEM **A**VDECC **E**ntity **M**odel
- AECP controls descriptors in the tree-like AEM

Synchronization

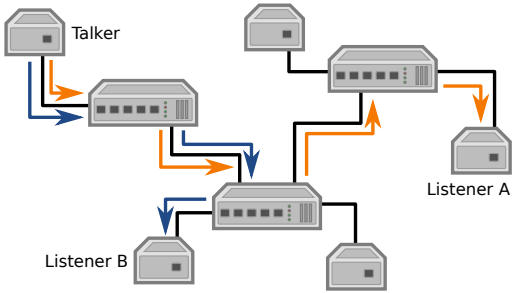
- establish a synchronized network clock
- network clock is clocked significantly higher than the word clock
- transmit samples along with network clock timestamps
- word clock / playout time is derived from the received timestamps
- AVB uses gPTP for network clock synchronization

Synchronization: gPTP

- **g**eneralized **P**recision **T**ime **P**rotocol
- Ethernet-only profile of PTPv2
- similar to NTP, but more precise
- good synchronization (<100 ns) requires hardware support
- each networked device has its own clock, including bridges
- synchronization is done in a master-slave architecture
- each segment selects a *master*
- the whole network selects a *grandmaster*
- bridges are preferred as (grand)master since they usually remain static

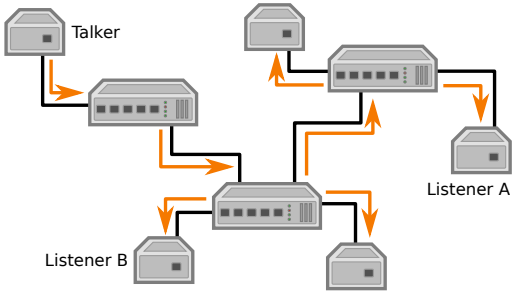
Stream Delivery: Unicast

- each frame is delivered to a device with a matching MAC address
- requires redundant frames
- wastes bandwidth and channels
- not used by AVB



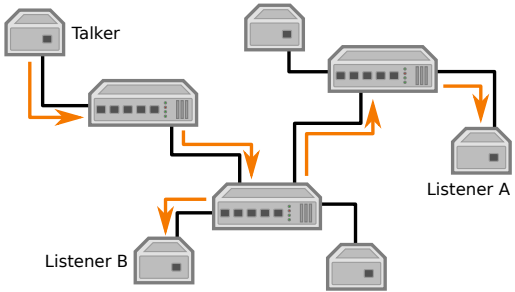
Stream Delivery: Broadcast

- each frame is delivered to all devices
- wastes bandwidth on nonparticipating links
- not used by AVB



Stream Delivery: Multicast

- deliver a single frame to multiple devices
- frames are addressed to a multicast MAC address
- listeners request multicast traffic
- requires support from the bridges



Stream Delivery: MAAP

- **M**AC **A**ddress **A**cquisition **P**rotocol
- allocate a multicast MAC address
- addresses come from a pool reserved for AVTP (AVB streams)
- simple probe/defend protocol

Reliable Delivery

- stream delivery becomes unreliable when interacting with other traffic
- frames get dropped when buffers are full
- frames may get delayed by other frames
- Solution: reserve bandwidth, prioritize and shape stream traffic

Reliable Delivery: Prioritization

- bridges sort received frames into multiple queues
- one per stream priority
- one for best-effort traffic
- prioritization is done based on VLAN priority tags

Reliable Delivery: VLANs

- virtual LANs
- regular frames are prefixed with a VLAN tag
- priority is part of the VLAN tag
- VLAN frames only forwarded to configured bridge ports
- we need dynamic configuration for plug and play

Reliable Delivery: MVRP

- **M**ultiple **V**LAN **R**egistration **P**rotocol
- allows requesting specific VLANs
- MVRP-aware bridges forward VLANs to requesting ports

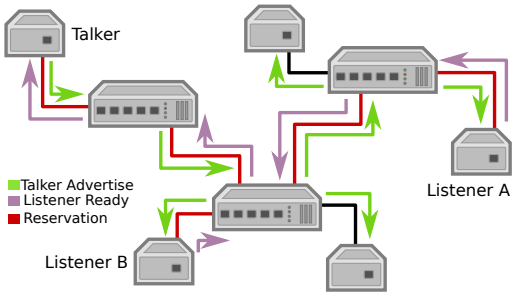
Reliable Delivery: Traffic Shaping

- streams at higher priority could starve streams at lower priorities
- streams should only use as much bandwidth as necessary for timely delivery
- AVB uses the **C**redit **B**ased **S**haper (CBS)
- queues build credit over time at a bandwidth specific rate
- queues may send a frame once enough credit has been build

$\Rightarrow$ for each queue we need to know the required bandwidth

Reliable Delivery: MSRP

- **M**ultiple **S**tream **R**egistration **P**rotocol
- talker advertise streams
- bridges broadcast talker advertisements
- listeners declare streams they want to receive
- bridges forward listener declarations towards the talker
- bandwidth is registered along the path



Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

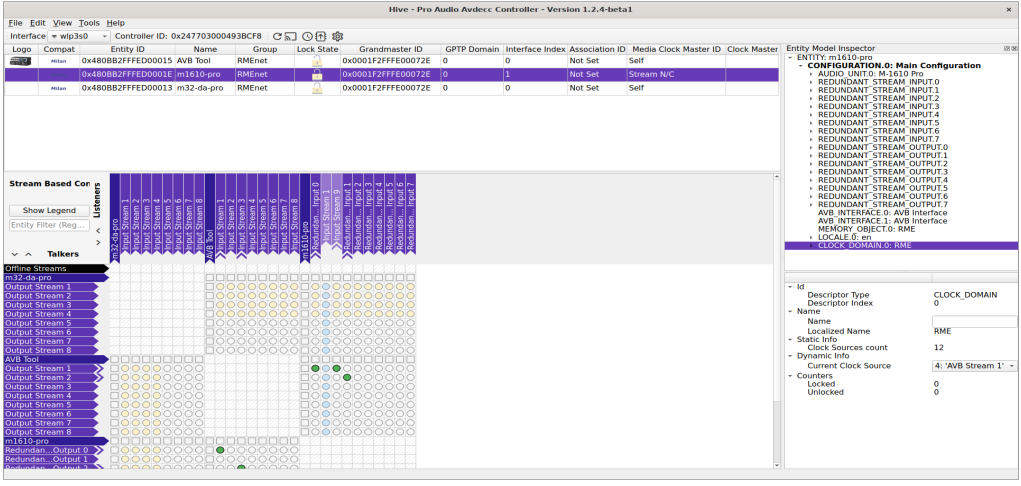
Goals?

To get the ease of the analog XLR connector transformed into full media and data interoperability!

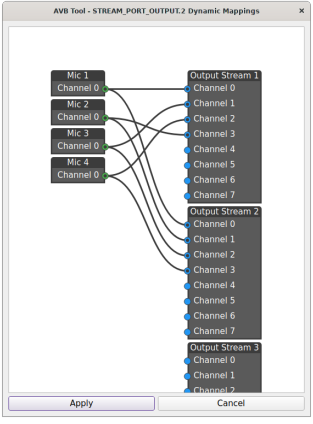
(*Henning Kaltheuner, d&B audiotechnik GmbH*)

- plug and play bridges configured by SRP
- low and constant latency 2ms (10 hops 100 Mbit/s)
- low jitter PTP typically achieves < 10 ns
- synchronized (word) clock PTP + transmitted timestamps
- synchronized playout PTP + transmitted timestamps
- reliable delivery prioritization and shaping
- hard real-time constraints

Hive



Hive


AVB Tool - STREAM_PORT_OUTPUT.2 Dynamic Mappings
Mic 1
Channel 0
Mic 2
Channel 0
Mic 3
Channel 0
Mic 4
Channel 0
Output Stream 1
Channel 0
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Output Stream 2
Channel 0
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Output Stream 3
Channel 0
Channel 1
Apply
Cancel


AVB Tool - STREAM_PORT_INPUT.2 Dynamic Mappings
Input Stream 1
Channel 0
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Input Stream 2
Channel 0
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Input Stream 3
Channel 0
Channel 1
Headphones 1
Channel 0
Headphones 2
Channel 0
Apply
Cancel

Hive

- AVDECC Controller
- GNU LGPL 3.0
- developed by L-Acoustics
- entity discovery
- allows connecting streams
- alpha-quality channel based routing
- clock management

OpenAvnu

- formerly Open-AVB
- components for building AVB/TSN systems
- GPLv2 and BSD license (mixed, not either)
- daemons for: msrp, mvrp, maap, gPTP, shaping
- example talker/listener for: ALSA, JACK, gstreamer
- mostly tested on Intel I210
- hard to get compiling/working

linuxptp

- de-facto standard PTP daemon for Linux
- supports gPTP
- includes `phc2sys` to synchronize system time to PTP

Linux VLAN support

- VLAN support has been in the kernel for ages
- also includes MVRP support

```
# ip link add link eth0 \
> name eth0.2 \
> type vlan \
> id 2 \
> egress-qos-map 2:2 3:3 \
> mvrp on
```

Linux Traffic Control

- traffic control is available as a Linux subsystem
- includes Queuing Disciplines (qdiscs)
- qdiscs implements CBS
- needs a lot of manual configuration particularly with multiple queues
- refer to `https://tsn.readthedocs.io/qdiscs.html`

ALSA AVTP plugin

- alsa-plugins includes a plugin to send/receive AAF streams
- no interaction with VLANs or qdisc
- no ACMP, MAAP, MSRP support
- applications need `CAP_NET_RAW`
- requires synchronization of system time and gPTP time

```
pcm.aaf0 {
  type aaf
  ifname eth0.2
  addr 01:AA:AA:AA:AA:AA
  prio 2
  streamid AA:BB:CC:DD:EE:FF:000B
  mtt 50000
  time_uncertainty 1000
  frames_per_pdu 12
  ptime_tolerance 100
}
```

GStreamer AVB elements

- Gstreamer Bad has an AVTP plugin
- supports AAF (audio), CRF (clock) and CVF (video)
- same shortcomings as ALSA plugin
- GStreamers PTP element could support gPTP directly, eliminating the need for system time changes

What's missing?

- components work with each other, but not necessarily with compliant AVB equipment
- full talker/listener with ACMP, MAAP, MSRP support
- integration and convenience between components
  - get required VLAN interface from MSRP
  - set up queues according to local talkers
  - make MSRP declarations
  - get Multicast MAC via MAAP
  - ...
- OpenAvnu provides most of this in theory, but doesn't hold up

Thank you for your attention.
Any questions?