Review of AVB by Anton Prins 29-June-2010, Version 1.00

The review is based on information in IEEE documents, a basic implementation on XMOS devices, research on the protocol drafts public available and a lot of experience in the audio industry.

AVB looks like a rising star for media-networking. But what is the current status of AVB and what would be the impact on existing and future products/network?

A deeper look at the AVB protocols makes clear the media-network functionality is a combination of existing (draft) protocols:

P802.1BA Audio Video Bridging (AVB) Systems
P802.1AS Timing and Synchronization
P802.1Qat Stream Reservation
P802.1Qav Queuing and Forwarding

But also others protocols are important for correct functionality in now-a-days AVB:

IEEE1722 (or IEEE1733)/IEC 61883 Layer 2 transport protocol (Packets format)
802.1AB Link Layer Discovery Protocol
802.1Q VLAN Tagging
802.1D MAC Bridges standard

To build a total system solution it is required to make more agreements on protocols for:

- Management/Configuration (e.g. SNMP with AVB-MIB)
- Realtime Control Protocol (e.g. MambaNet)

Analyzing the 802.1AS/Qat/Qav/BA/AB and P1722 protocols makes clear AVB requires switches that have the 802.1 protocols implemented. So called AVB switches must be available to build AVB networks. Because of the multicast nature of AVB it is required these switches are strong in filtering multicast traffic. Also time stamping at the PHY is a requirement.

AVB uses Ethernet ‘layer 2’ and implement protocols for a QoS guarantee at ‘level 2’. QoS is already available for IP (which is a layer 3 protocol on Ethernet).
What is the main reason for AVB to use ‘layer 2’?

Will large manufacturers implement these protocols?
(How would this cooperate with their own QoS? and with the fact that an AVB listener can reserve 75% of the connection bandwidth?).

Conclusion: AVB cannot be used on current (non-AVB) Ethernet equipment: e.g. (managed) switches. This is very strange for an IEEE802.1 standard!

The AVB media package is defined in the IEEE1722. Further research on IEEE1722 made clear that inside the AVB-stream-packages the IEC 61883 is used. This protocol is mentioned for IEEE 1394 (FireWire). In other words the FireWire digital interface protocol is embedded in Ethernet/AVB.
It is a good solution to use existing protocols, but using the IEC 61883 package would mean that most of the header fields are fixed or unused by the AVB implementation. The timestamp of IEC 61883 could be valuable in applications 'FireWire-AVB-FireWire' but in "AVB only" systems it is redundant information because an AVB IEEE1722 packet already has timestamps.
Why is the IEC61883 standard chosen as base for the IEEE1722 package?

IEEE1722 in combination with IEC 61883 seems to use much more overhead then required.

In the AVB header there are fields to solve some issues (from higher level); for example:
tv = timestamp valid (selects to use AVB timestamp or other timestamp?).

The AVB timestamp is also called 'presentation time'. This suggests there are two different times…
Researching the documentation about the presentation time resulted in the following calculation for the presentation time in an AVB IEEE1722 package:

timestamp + variable delay (default 2ms) = presentation time

The AVB IEEE1722 package contains the presentation time and not the 'timestamp'.

When the original timestamp is used in the AVB stream package, and not presentation time, the listener can determine the 'stream delay'. At the listener you may:

1. Add the default 2ms 'transfer delay' to the timestamp to know its presentation time.
2. Present the stream with a different (fail-safe) latency.
3. Make timestamp to presentation time delay higher than 2ms if the listener is able to buffer this and gives no functional problems.

Why is the 'timestamp' adjusted by the talker before transmission?

Because the transfer delay is unpredictable for the talker a 2ms is used by default which is most of the time more than the real transfer delay.
Second, with the calculation of the presentation time (with a variable delay value) the relation to the source timestamp is lost.

Looking in the AVnu Whitepapers it says:
"The IEEE AVB task group is developing a series of enhancements that provide the means for the highly-reliable delivery of low latency, synchronized audio and video".

Compared to existing technologies the 2ms default latency is not 'low latency' as people will expect (compared to MADI/HyperMAC/Ethersound etc.) Of course you may adjust this latency but that's engineering level.

Network virtual routes (streaming connections) are made to gain functionality. Important will be the possibility to make and delete streaming connections over the network. Currently the relation between the ‘Stream IDs’ and L2 multi-/unicast addresses is unknown. Possibly the ‘Stream ID’ parameter will be the one to determine if a connection must be made. But there is no information for ‘Stream discovery’ and management/configuration of AVB devices. To adjust the management and configuration parameters a good choice will be SNMP.

“The IEEE 802.1Qat Stream Reservation Protocol (SRP) provides mechanisms for reserving stream bandwidth that allows endpoint applications to configure the routes, eliminating the need for this type of infrastructure network engineering”.
This protocol is one of the reasons why AVB is not backwards compatible..
It will be difficult to explain to end-users they cannot use existing equipment to create an AVB network. The zero config principle is a nice idea, but this must not exclude backward compatibility (and of course everybody will normally understand older equipment will give fewer guaranties, but they don’t understand that ‘discovery/connection’ isn’t allowed/possible).

Beyond the technical aspects there is also marketing, and because marketing is doing a good job (thx to AVnu and other pioneers) technicians, engineers and manufactures have to research/develop fast. They also need to realize not to get in situations like seen before:

- mLAN (IEEE 1394 was adapted slowly; the speed/length forecasts took some time and finally never adapted large-scale)
- CobraNet (Is licensed, not free to implement parallel with other (custom) protocols)
- Proprietary solutions developed solely for own implementation

Currently products are released/presented with AVB compatibility. For customers AVB = AVB, but can we give them this guarantee?

A customer will have some (basic) assumptions:

- Can you buy AVB equipment? (at the same price level as current solutions)
- Can you use AVB equipment from many manufactures together? (Without configuration/adaption, zero config)
- Because AVB = 802.1, you must be able to use it in mixed environments… so with non-AVB 802.1 equipment (including switches, where you can discuss if blocking of AVB-data is allowed/required)

From a customer’s point of view you can say: Circuit switched solutions are good enough and easy to operate (e.g. physical cables/multi-cores). So important is to make clear the goal for networking and to understand how you can explain it to a customer/user (and what will his reaction be?).

We have to keep in mind that customers are used to current solutions. They could compare them to the arguments of the rising technologies:
So most of current analog connection implementation can be considered as:
- Zero config networks
- Extreme low latency
- Bandwidth guaranteed
- Routing by physical connection (physical label), no virtual IDs or addresses.
- Discovery for non-engineers possible.
- QoS is equal to cable quality.

Reading these last points can make you feel uncomfortable. Of course technology can bring solutions, but we have to focus on the 'pluses':
+ Standard for interconnecting streams over a network (routing/connection management)
+ Added functionality by cooperation between devices of multiple manufacturers over a network (Solve the video/word-clock sync problem(s), device management and control).

Nevertheless there are still some concerns about the current AVB (drafts):
- Need special network switches (AVB switch), not backward compatible with 802.1; only forward compatible.
- Standard latency is relatively high (2ms).
- Unnecessary data transfer because of using 'FireWire-over-Ethernet'.
- Routing, Management and real-time control unclear path for current AVB (while market(ing) is ready/waiting).

The zero config for media-clock synchronization looks like a great solution certainly if it allows multiple media-clock-demains. AVB has one great advantage, it may be a worldwide open standard and if a solution for stream routing/management and clocking is implemented AVB, devices of multiple manufacturers could work together.

My final conclusion regarding AVB is that AVB is promoted to work over standard Ethernet, the reality is: 'standard Ethernet can go over AVB switches'.

Before finishing the AVB project it would be important to rethink the chosen path and second to have good (standard) solutions for the currently missing 'pluses'.

Thanks for reading and I appreciate any reaction towards this point of view.

Anton Prins
a.prins@d-r.nl

Special thanks to Wouter van Bakel and Jan Betten for their assistance and comments.