Solutions, Highlights and Challenges of Ethernet AVB Solutions

FTF-AUT-F0051

Stefan Singer | Director, EMEA FAE Automotive

APR. 2014
Agenda

• In-Car Ethernet Production Rollout
• Ethernet Audio Pre-Development
• Ethernet AVB
• Channel Utilization
• Stream Based Shaping
• Ethernet AVB Audio Solutions
• Media Clock Generation / Recovery
• Combined Audio / Video Network
In-Car Ethernet Production Rollout
NXP: Production Gateway with Ethernet Diagnostic – SOP 2008 (Freescale MPC5567)
First Production Ethernet Surround Camera System – SOP 2013 (Freescale MPC5604E Plus Ethernet Camera Software Application)
First Production Ethernet AVB Audio Amplifiers

MPC564xC

Vybrid
Ethernet Audio Pre-Development
Development Steps for Ethernet Infotainment

- 2006
- 2007/2008
- 2010
- 2011
- 2012
- 2013
- 2014
# Audio Over Ethernet Options

<table>
<thead>
<tr>
<th>OSI Layer</th>
<th>Attributes</th>
<th>Standard</th>
<th>Comment</th>
<th>Compatible with standard PC</th>
<th>Reuse of standard Software stack (e.g. on Linux)</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer 1</td>
<td>Modified Ethernet frame structure, therefore typically require non standard media access control (MAC)</td>
<td>e.g. A-Net, AES50, RockNet</td>
<td>Ruled out, since reuse of standard MAC layer was seen as a priority</td>
<td>No</td>
<td>No</td>
<td>Can be very low</td>
</tr>
<tr>
<td>Layer 2</td>
<td>Encapsulate audio data in standard Ethernet packets, use standard IEEE802.3 hardware</td>
<td>AVB</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Low single digit ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proprietary standards, e.g. CobraNet</td>
<td>Ruled out to avoid a proprietary standard</td>
<td>Depends on standard</td>
<td>No</td>
<td>Depending on standard, typically low</td>
</tr>
<tr>
<td>Layer 3</td>
<td>Encapsulate audio data in standard IP packets</td>
<td>UDP/IP</td>
<td>Used for early prototypes starting in 2006</td>
<td>Yes, easy use of standard tools, e.g. VLC player</td>
<td>Yes</td>
<td>Using standard stacks and equipment, latency is typically &gt;= 20 ms. With optimized stacks lower latency can be achieved, but that requires optimization of entire network and therefore loss of reuse benefit</td>
</tr>
</tbody>
</table>
Ethernet AVB
Ethernet AVB Standard

- **Audio Video Bridging** (AVB) is a common name for the set of technical standards developed by the IEEE Audio Video Bridging Task Group of the IEEE 802.1 standards committee. The charter of this organization is to "provide the specifications that will allow time-synchronized low latency streaming services through IEEE 802 networks".[1] These consist of:
  - **IEEE 802.1AS**: Timing and Synchronization for Time-Sensitive Applications (gPTP)
  - **IEEE 802.1Qat**: Stream Reservation Protocol (SRP)
  - **IEEE 802.1Qav**: Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
  - **IEEE 802.1BA**: Audio Video Bridging Systems

Ethernet AVB for Automotive Sub Standard

- **Specifications absolutely required:**
  - IEEE 1722 Layer 2 Transport Protocol
  - IEEE 802.1AS Timing and Synchronization

- **Further specifications that can either be optional or will not be used:**
  - **IEEE 802.1Qat:** Stream Reservation Protocol (SRP)
    - Used to dynamically reserve bandwidth for specific streams
    - Distinguishes between class A, class B, class C and best effort traffic
    - Actual reservation is really implemented in the switch, endpoints just need to understand the protocol to initiate the reservation (pure software on the endpoint)
    - For automotive applications dynamic service discovery is prohibitive because of startup time
      - **Option I:** Static switch configuration
      - **Option II:** SRP with preconfigured startup
  - **IEEE 802.1Qav:** Forwarding and Queuing for Time-Sensitive Streams (FQTSS)
    - See separate slide
  - **IEEE 802.1BA:** Audio Video Bridging Systems
    - Currently work on automotive profile is ongoing
  - **IEEE 1722.1:** Device Discovery, Enumeration, Connection Management and Control Protocol (DECC)
    - For automotive applications this protocol is unlikely to be used due to the startup time required
IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams – Generic

• A common **misconception** is that any AVB or AVA application requires dedicated traffic shaping hardware in the Ethernet controller.

• The system requirements need to be analyzed to discover how they can be most efficiently implemented.

• The question “Does your MCU / MPU support AVB?” is really misleading and should be replaced by one level of additional info – please see the following slides for guidance.
IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams – Streaming Sinks

- AVB does not envision any policing of the incoming traffic for a sink. Therefore a sink endpoint does not necessarily need to have different queues for the received traffic classes.
  - Requirement for priorities between receive queues is application specific. For example, a pure audio endpoint or Ethernet audio amplifier probably will not use multiple class queues.
  - However a multi-function infotainment box, such as a head unit, might need further hardware support for efficient processing of the receive queues.

- Freescale’s Ethernet Streaming Driver coupled with the Streaming IO layer provide an extremely efficient, high-performance mechanism to receive high bandwidth streaming traffic with minimal processing overhead.
IEEE 802.1Qav: Forwarding and Queuing for Time-Sensitive Streams – Streaming Sources

- **Multiple hardware queues in the AVB source is an application specific requirement**
  - An AVB source driven by a synchronous stream source (e.g. camera sensor in Ethernet camera, I²S / TDM audio for audio source) does not further queue the stream data
    - This would increase latency due to additional buffering and is therefore not desired
    - However, best effort traffic (e.g. TCP/IP diagnosis traffic) should be artificially throttled to ensure minimum latency for the streaming traffic
  - Freescale’s Ethernet Streaming Driver with its stream builder modules (e.g. 1722) provide an extremely efficient, high-performance mechanism to send high-bandwidth streaming traffic with minimal processing overhead
    - E.g. 70 Mbps video stream on 60 MHz e200z0 core on MPC5604E consumes only ~ 35 % CPU load
  - However, a source that sends compressed stream data (e.g. MP3 / AAC audio or H.264 video) would ideally have a hardware traffic shaper in the Ethernet controller. That would control the rate of the decoding process by back pressuring.
AVB Timing: Challenges

- AVB defines low latency Class A traffic with a maximum latency of 2 ms over 7 hops
- AVB defines higher latency Class B traffic with a maximum latency of 50 ms over 7 hops
- AVB 61883-6 Audio Format suggests “A Class A isochronous packet is generated every 8 kHz”
  - Represents 125 µs worth of audio
  - For an audio CD type (16 bit sample, stereo) stream @ 48 kHz, this means 6 samples → 24 bytes of payload
- If we need to buffer 2 ms worth of audio, sending audio data at 125/250 µs interval can pose a challenge
- Consider video playback scenario (extra slide)
# AVB Audio Stream Channel Utilization

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono 8 bit</td>
<td>1</td>
<td>8</td>
<td>48</td>
<td>0.384</td>
<td>4.992</td>
<td>8%</td>
</tr>
<tr>
<td>Mono 16 bit</td>
<td>1</td>
<td>16</td>
<td>48</td>
<td>0.768</td>
<td>5.376</td>
<td>14%</td>
</tr>
<tr>
<td>Stereo 16 bit</td>
<td>2</td>
<td>16</td>
<td>48</td>
<td>1.536</td>
<td>6.144</td>
<td>25%</td>
</tr>
<tr>
<td>Stereo 32 bit</td>
<td>2</td>
<td>32</td>
<td>48</td>
<td>3.072</td>
<td>7.68</td>
<td>40%</td>
</tr>
<tr>
<td>5.1 - 16 bit</td>
<td>6</td>
<td>16</td>
<td>48</td>
<td>4.608</td>
<td>9.216</td>
<td>50%</td>
</tr>
<tr>
<td>7.1 - 16 bit</td>
<td>8</td>
<td>16</td>
<td>48</td>
<td>6.144</td>
<td>10.752</td>
<td>57%</td>
</tr>
<tr>
<td>5.1 - 32 bit</td>
<td>6</td>
<td>32</td>
<td>48</td>
<td>9.216</td>
<td>13.824</td>
<td>67%</td>
</tr>
<tr>
<td>7.1 - 32 bit</td>
<td>8</td>
<td>32</td>
<td>48</td>
<td>12.288</td>
<td>16.896</td>
<td>73%</td>
</tr>
</tbody>
</table>

AVB framing overhead per observation interval

<table>
<thead>
<tr>
<th>Field</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter Frame Gap</td>
<td>12</td>
</tr>
<tr>
<td>Preamble and SFD</td>
<td>8</td>
</tr>
<tr>
<td>Ethernet Header</td>
<td>14</td>
</tr>
<tr>
<td>IEEE 802.1Q VLAN</td>
<td>2</td>
</tr>
<tr>
<td>AVBTP</td>
<td>24</td>
</tr>
<tr>
<td>IEC61883</td>
<td>8</td>
</tr>
<tr>
<td>Frame Check Sequence</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
</tr>
</tbody>
</table>
Stream Based Shaping
Standard Approach for IEEE 802.1Qav Shaping for Multiple Streams

Streaming Source #A → Software Scheduler → Typical AVB Ethernet IP with Class-based Shaper

Class A Credit-based Shaper

Class B Credit-based Shaper

Best Effort Traffic

Static Priority

Ethernet MAC

Streaming Source #B

Streaming Source #N
Deficiencies of Software Scheduler Approach

• Somebody reading these slides could think that the scheduling function could be implemented in software without special hardware support.
  – This software would basically run at an interval, which is a multiple of the class A/B schedule, and preload a number of streams for the credit-based shaper.

• However, if a streaming source does not use its entire bandwidth reserved in the bandwidth reservation process, the shaper will still have credits available after it has sent all the prescheduled frames. This will lead to the credit-based shaper actually trying to send frames that were intended for the next transmit schedule.
  – This is a violation for the AVB traffic contract and must be avoided!
Three streaming sources have been configured with bandwidth reservation for their nominal bandwidth. In the first slot they completely use up their reserved bandwidth.

In the second slot, one or more of the streaming sources do not completely consume their reserved bandwidth. The shaper has extra credits left at the end of the shaper interval. Without additional mechanisms it would start to pull in frames from the next schedule interval.

This frame will be pulled in and sent before its desired transmit schedule.
AVB Class A/B Implementation Challenge

- Class A/B traffic needs to be shaped / scheduled with a 125/250 μs interval rate
- Class queuing requires that only max interval frames of each stream may be scheduled within one scheduling cycle (125 μs for class A, 250 μs for class B)
- This means that either a software scheduling task needs to run at the frequency of the scheduling cycle (125 / 250 μs), or the shaper needs to be aware of the individual stream IDs
  - On a preemptive multitasking operating system (e.g. Linux, QNX, Windows) a scheduling cycle of 125 / 250 μs is basically impossible. If implemented as an interrupt service routine, the additional load of this overhead will basically kill any system performance
  - Extending the shaper to be aware of the individual stream IDs breaks the OSI layer approach
    - Shaper needs to be aware of the used protocols (e.g. understand 1722 frame format)
    - Or have separate FIFOs for each stream ID (since the scheduling interval is small and Ethernet minimum frame size is 64 bytes, this would actually be a limited number of entries)
  - AVB standard only describes “class shapers”, so this challenge is not addressed in the standard
Problem Solving

• With existing standard IP, the only option is to run the software scheduler at the class observation interval of 125 or 250 μs
  – For a preemptive operating system, e.g. Linux, this is almost impossible. Instead, an RTOS should be used to run that realtime load
• Use “stream-based” Ethernet AVB hardware IP, e.g. on i.MX6SLX or MPC5748G, to avoid that hard realtime requirement
Ethernet AVB Audio Solutions

e.g. Audio Amplifiers
Ethernet AVB Audio Node Solution Range

- **MPC56xx/MPC57xx MCUs**
  - Highest integration: true single-chip solution, memories and power supply included, lowest overall system cost
  - MCU is intended as a “network interface” to a standard audio processing solution (e.g. DSP or smart codec) **with little or no audio sample processing** (e.g. amplifier / mixer) from the MCU
    - e200z0 or e200z2 core-based MCUs – NO audio sample processing
    - e200z4 or e200z6 core-based MCUs – little audio sample processing, e.g. few channels of MP3 decode

- **Raleigh**
  - Specific Solution for active Tuner

- **Vybrid VFxxxR**
  - Mid-level integration (external NVM needed, but up to 1.5 Mbyte of internal SRAM will remove the need for DRAM in many cases)
  - Up to 400 MHz ARM® Cortex™-A5 core with NEON SIMD engine and integrated ASRC (Asynchronous Sample Rate Converter) HW allow for **medium level audio sample processing** throughput

- **i.MX6 family of applications Processors**
  - Lowest integration (external NVM and DRAM needed)
  - Up to 4 x ARM® Cortex™-A9 core @ 1 GHz with NEON SIMD engine and integrated ASRC (Asynchronous Sample Rate Converter) hardware allows **very high audio sample processing** throughput
Freescale Ethernet Audio Framework

- **Audio Clock**
  - audio_clock
  - media_clock
  - stream_time
  - stream_io
  - i2s_if
    - Interface to I2S Audio
  - most_if
    - Interface to MOST Audio
  - ethersync
    - Ethernet (AVB) to Synchronous Audio Bridge
  - helix_if
    - Interface to Helix MP3 Decoder
  - wave_if
    - Interface to WAVE playback
  - decode_sync
    - Software Decoder to Synchronous Audio Bridge

- **PTP v2 stack**
  - Normal IP-Stack
    - FreeBSD / LwIP / EB / 3SOFT
  - AVA_LL_PTP_if (wrapper)
  - AVA_LL_EB_if (wrapper + limiter)

- **Ethernet MAC**
  - Ethernet Low Level Driver
    - Freescale

- **Audio Applications (e.g. PDC, GONG, …)**
  - sync_bridge
  - i2s_common
  - dsplll
  - sai
  - mlb_if
  - MLBSYNC
  - MediaLB

- **AUTOsAR OS**
  - or
  - eCOS

- **MPC5xxx Audio Framework**
  - demo code from Freescale
  - – not included in commercial offer

- **Ethernet Streaming Software**
  - licensable from Freescale

- **Off the shelf software**
  - licensable from 3rd party (Elektrobit, IXXAT)

- **Hardware**
MPC5604E (Salsa) Ethernet Audio Amplifier Reference Design

- Automotive type connector
- Integrated BroadR-Reach Ethernet PHY
- Integrated audio codecs
  - 2 x stereo codec
  - 1 x surround codec (8 Channels)
- Integrated AVB Mediaclock PLL
AVB Audio Design with Vybrid Tower Stack

- Vybrid TWR-VF65GS10
- Tower multichannel codec and AVB mediaclock multiplier card
- TWR-SER
  - Ethernet PHY
  - Serial connector
MPC5748G (Calypso) Ethernet AVB Audio design

- MPC5748G (Calypso) EVB features an audio connector
- This can be used to build an audio design using an external daughtercard
Media Clock Recovery
Media Clock Recovery with ENET Module

- ENET module has dedicated hardware for media clock generation and recovery
- This hardware generates a reference clock (e.g. 8 kHz for AVB class A stream)
- This reference clock needs to be multiplied up to the desired media clock (e.g. audio MCLK of 12.288 MHz) by a hardware clock multiplier
- Only a very thin software layer is involved for the entire media clock generation or recovery
Combined Audio / Video Network
**Scenario:** Head unit streams movie to the rear seat entertainment unit

- Video data is received by the RSE, audio to the Ethernet amplifier
- Consumer video (e.g. H.264 main profile) decompression has some very significant latency (typically 50-100 ms)
- Audio and video need to be “lip synced” with ~ 10 ms accuracy

Some significant amount of audio data needs to be stored temporarily

- **Option A:** In the Ethernet amp → significant amount of RAM required, approach not compliant to AVB concept
- **Option B:** Streaming source (head unit) needs to buffer audio sufficiently and send audio delayed compared to video
One central unit in the infotainment architecture (typically the head unit, or potentially the RSE) needs to perform transport stream demuxing. This extracts the time stamps of the transport stream. Those time stamps can then be used to calculate the IEEE1722 presentation time for the separate audio and video streams that are generated in this demuxing process and that can be forwarded as IEEE1722 frames to other sinks. Video will be forwarded compressed to other sinks (e.g., the RSE), audio will typically be uncompressed to PCM audio data, but since we now have time stamps, it could also be forwarded in compressed form.
Summary

- Ethernet is in automotive production today
- Ethernet AVB is the most promising solution for next-generation infotainment networks and first production projects have been awarded
- Implementation details must be analyzed
  - For example, stream-based shaping
- Ethernet AVB audio solutions are ready for production
- Combined audio / video network must be analyzed with its constraints