Linux Network Enabling Requirements for
Audio/Video Bridging (AVB)

Linux Plumbers 2012

Eric Mann

Software Architect - Intel

August 29 2012

Legal Disclaimer

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

All information provided related to future Intel products and plans is preliminary and subject to change at any time, without notice.

Intel, the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

Copyright © 2012

# Agenda

Audio Video Bridging (AVB) Overview

AVB on Linux

Questions

Why AVB?

People have been sending audio & video over networks for decades now ... so what is new about AVB?

- Buffering data to hide packet loss and bandwidth issues causes presentation lag (the echo cancellation in your ear tolerates up to 10 msec)
- Sending audio to multiple devices requires synchronization of the presentation-time (e.g. lip-sync of video to audio, warbling in 7.1 audio systems) ,
- Auto companies want to use AVB to reduce cost.
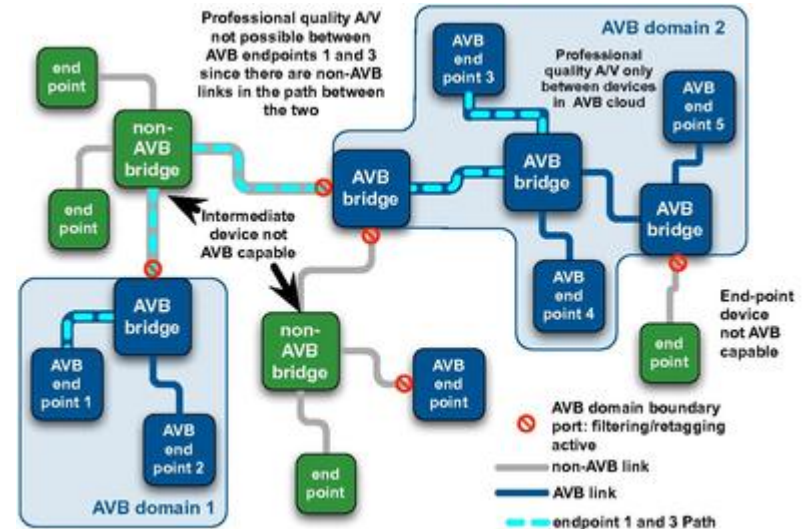
What is "AVB"?
for this discussion at least ...*

AVB := 802.1Qat + 802.1Qav + 802.1AS

- 802.1Qat := MMRP + MVRP + MSRP = "Stream Reservation"

- 802.1Qav := "Hardware-based traffic shapers"

- 802.1AS := "Time Synchronization"

* There are a number of additional user-mode protocols layered on top to handle AVB device discovery & control, as well as media framing:

- L2-based IEEE 1722, 1722.1

- L3-based 1733 & RTP

- I also don't talk about the receiving end of these streams – as they only really need to support 802.1AS and MRP – no need for traffic shaping.

(intel)

Functional Overview
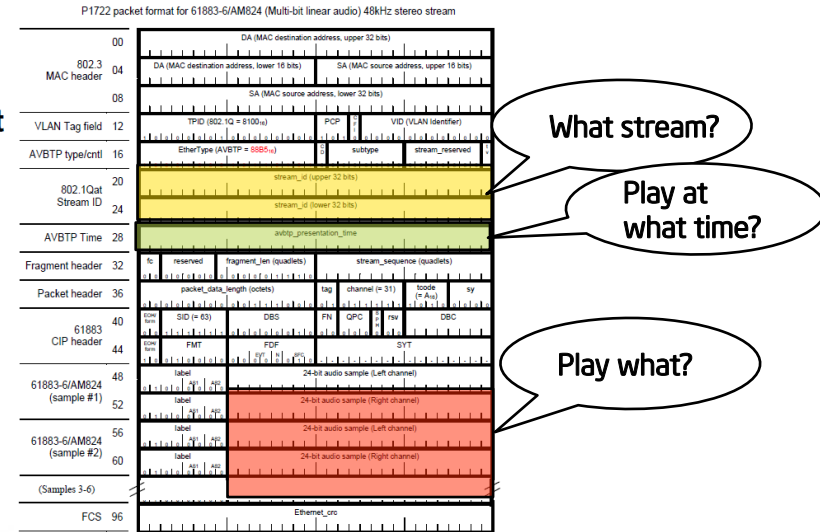## Streams to Classes to User Priorities to Ethernet
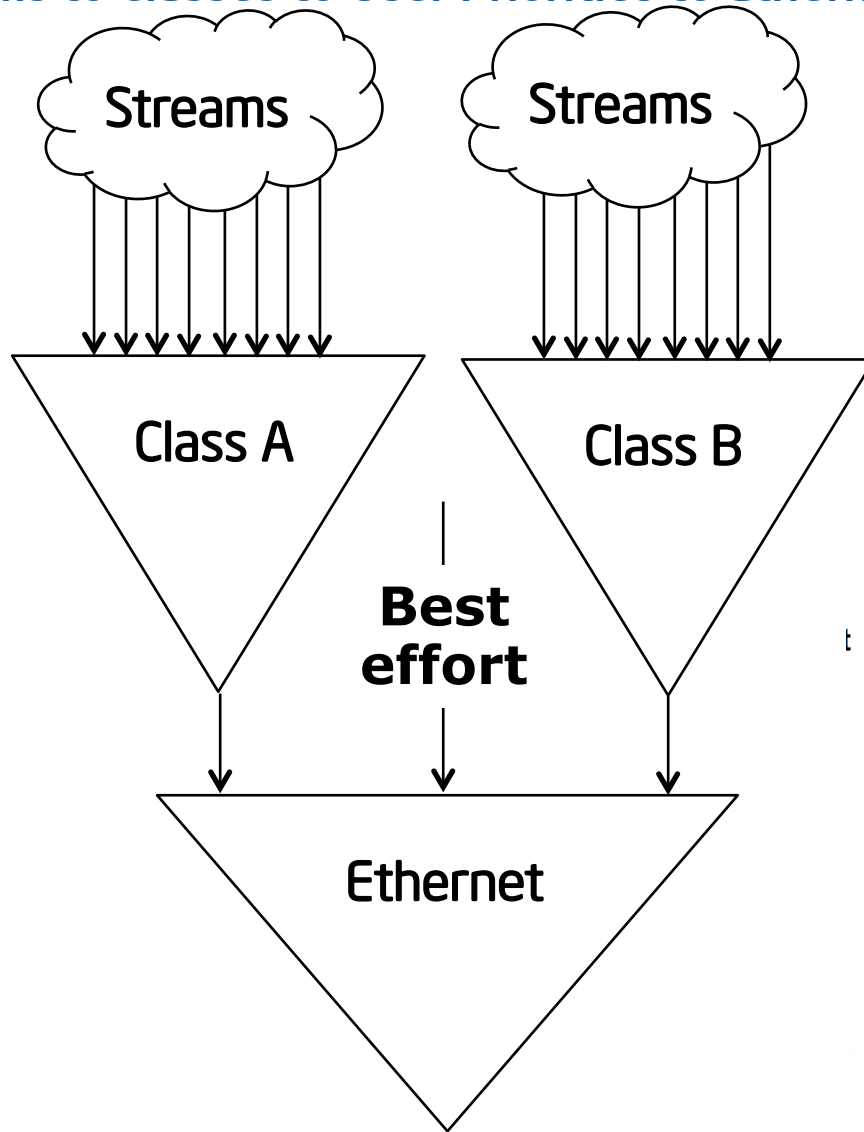
Notes

Each stream contains zero (0) or more media channels.

As streams are added/removed to a class, bandwidth of the <u>class</u> is adjusted.

Discovery of UserPrio and VLAN made during Qat (the "DOMAIN" msg).

Proof of Concept Linux Stack [DEMO]

e1000.soureforge.net /linux_igb_avb

Talker Endpoint App

Time services 802.1AS linuxptpd	Network Map And Reservation 802.1Qat	802.1Qav Bandwidth Control

igb library

User-space media queues

igb_avb.ko (kernel mode)

Intel NIC Hardware (802.1Qav Shaper)
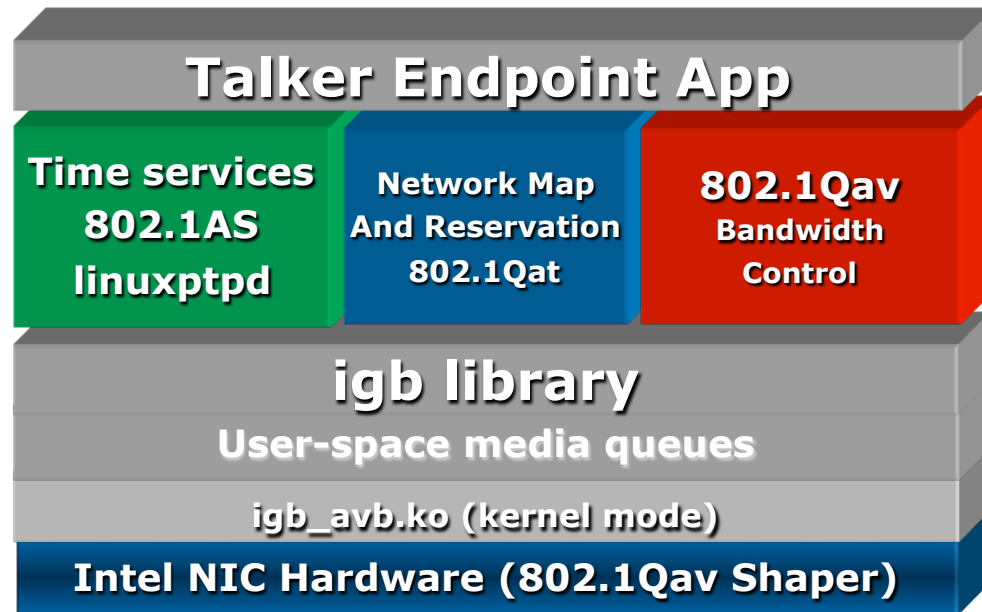
Notes

In the current design, one of the queues is kept by the in-kernel igb module for best-effort traffic. Media queues are mapped to user-space.

Remember DCB/FCoE? AVB drops right in …
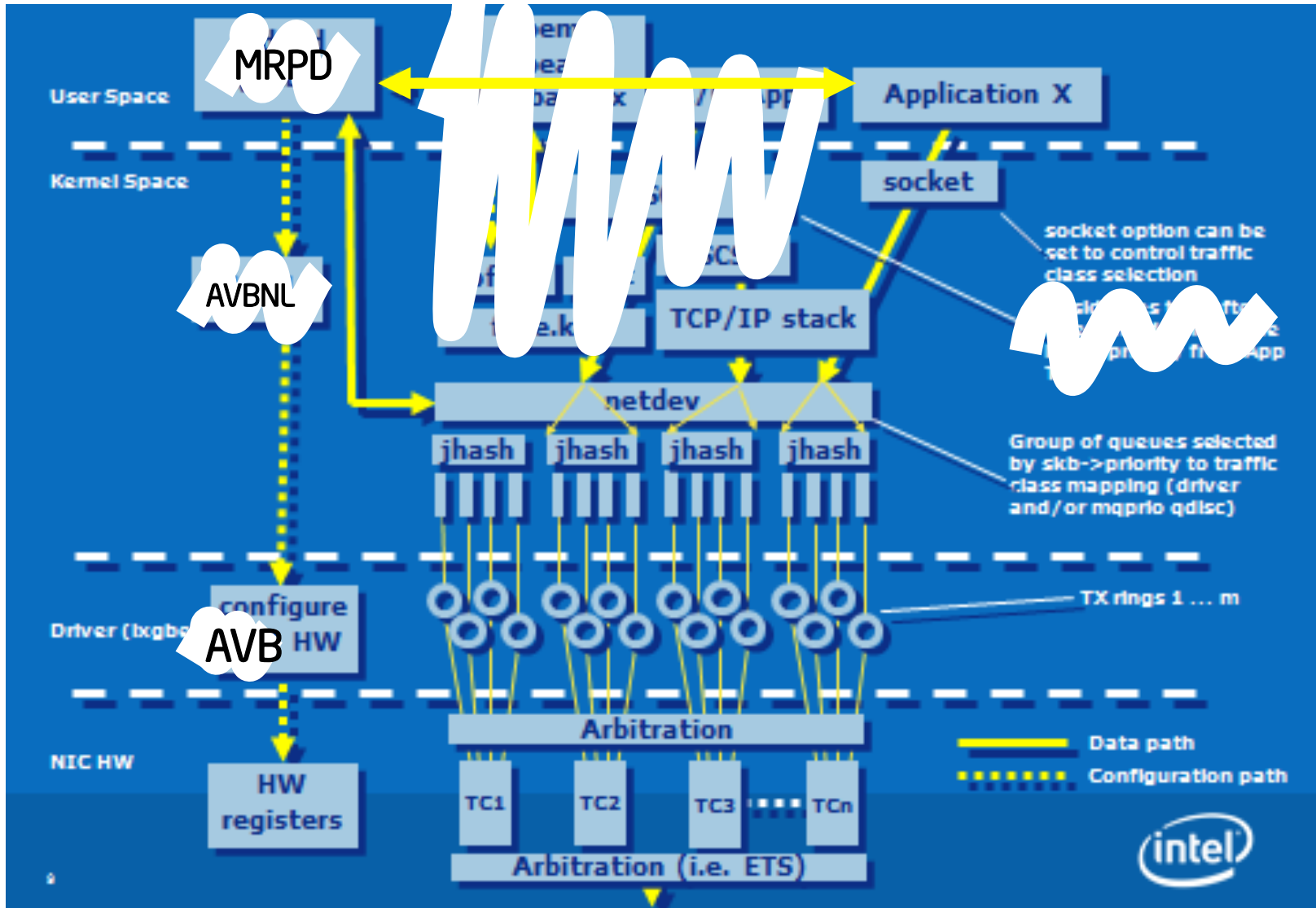
Gap 1 - Timed Transmission

- Prior patches enabled passing timestamp information to applications on receive, and returning tx timestamps on the ERRMSG queue …

- Need to extend timestamps to control actual packet transmit time …

  - As the class shaper is adjusted, the class shaper may momentarily send more or less data on existing streams with traffic outstanding on a class. This can cause over-runs or under-runs on receivers.

  - Helps with sequencing of streams onto the same Class shaper (e.g. multiple apps write onto streams and expects the class shaper to pace out the traffic fairly …) ← can workaround with driver-based knowledge of individual streams (not desirable).

  - This works with L2 or message-based protocols (like UDP) – not stream based like TCP however.

CMSG PROPOSAL for Timed Tx

New CMSG type to indicating timed transmission (as well as corresponding transmission time).

- Transmission time is relative to the physical interface.

- User-mode application will need to perform translation if required from the global "AS" time to the physical interface relative time.

```
cmsg->cmsg_level = SOL_RAW; /* UDP too? */
cmsg->cmsg_type = TIMED_TX;
cmsg->cmsg_len = CMSG_LEN(8); /* seconds.nsec format */
/* Initialize the payload: */
tx_time = (u_int64_t*) CMSG_DATA(cmsg);
*tx_time = tx_seconds << 32 | tx_nsec;
```

Gap (2) – ethtool support

Would be desirable to dynamically configure "AVB" functionality (enable/disable) via ethtool.

- Based on our experience, requires device reset.

- New option to display as well as change "AV" state of adapter interface.

For More Information & Planned Future Steps
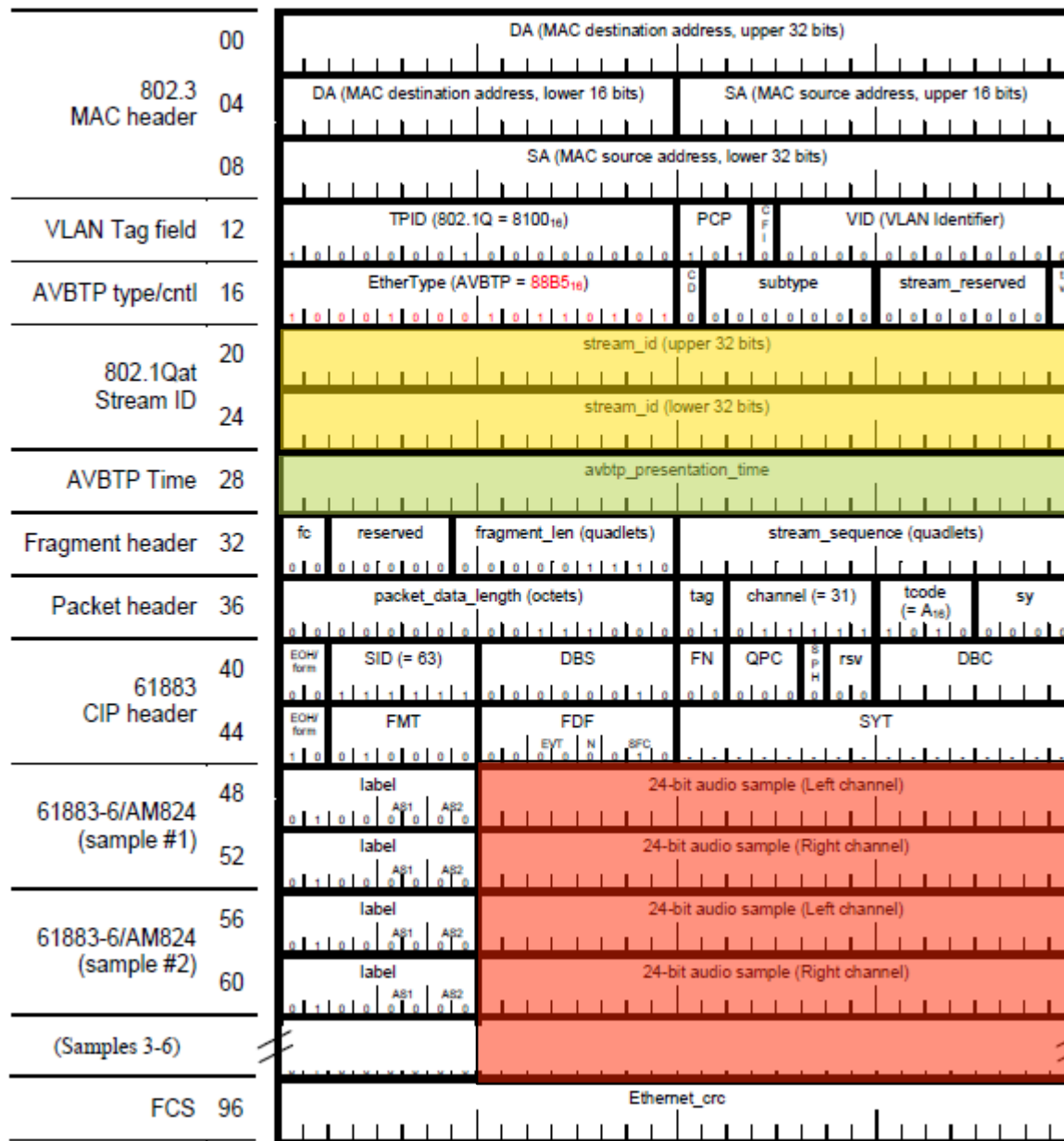
- "linux_igb_avb" tarballs of this code are posted to Intel's existing e1000.sourceforge.net project.

- In process of establishing a github repos* for open source collaboration and contribution *project name pending Intel Legal approval

- Enhancements to example applications.

(intel)

Q & A?

P1722 packet format for 61883-6/AM824 (Multi-bit linear audio) 48kHz stereo stream

P1722 packet format for 61883-8, Source Packet Type=0 (video data)

Subtype data	00	CD 0 / subtype 0 0 0 0 0 0 0 0 / stream_reserved 0 0 0 0 0 0 0 0 / t v / subtype_data2 0 0 0 0 0 0 0 0 / gm_info
802.1Qat Stream ID	04	stream_id (upper 32 bits)
	08	stream_id (lower 32 bits)
AVBTP Time	12	avbtp_timestamp
Fragment header	16	fc 0 0 / pkt_hdr_len 0 0 0 0 1 1 0 / fragment_len (quadlets) 0 0 0 0 0 0 1 1 1 1 0 / stream_sequence (quadlets)
Packet header	20	packet_data_length (octets) 0 0 0 0 0 0 0 1 0 1 1 0 1 1 0 0 / tag 0 1 / channel (= 31) 0 1 1 1 1 1 1 1 / tcode (= A_{16}) 1 0 1 0 / sy 0 0 0 0
61883 CIP header	24	EOH/ form 0 0 / SID (= 63) 1 1 1 1 1 1 / DBS / FN 0 0 / QPC 0 0 0 / SPH 0 / rsv 0 0 / DBC
	28	EOH/ form 1 0 / FMT 0 0 0 0 0 0 1 / FDF reserved / SYT
61883-8 Source Packet Data	32	Compression Mode Specific Information / VDSPC / Line number / r / Ver = 0 0 0 / Type = 0 0 0 0
	36	byte 1 video data / byte 2 video data / byte 3 video data / byte 4 video data
	...	... / ... / ... / ...
	752	byte 717 video data / byte 718 video data / byte 719 video data / byte 720 video data