Synchronized real time audio streaming over ethernet in embedded systems

...

A thesis submitted for the fulfillment of the degree
Master of Science in Information Technology

Indumathi Duraipandian
926386

Agenda

Introduction
Thesis objectives
AVB protocol stack
Beagleboard platform
Prototype design
Development
Evaluations
Limitations
Conclusion

Introduction

Analog audio connections

Audio as electrical signals

+	No Latency	–	Prone to noise
+	Simple to use	–	Point to point connections
+	Synchronization can be achieved easily	–	Processing required at receiving end to cleanup signal

Digital general connections

USB, Firewire, Thunderbolt etc…

+	High bandwidth	–	High latency
+	Simple to use	–	No Synchronization

Past → Future

Digital audio connections

SPDIF, ADAT, AES, DVI, HDMI etc...

+	More bandwidth, i.e. single connection can carry several channels	–	Higher cost
+	Resilient to noise	–	Not portable and incompatible standards
+	Synchronization achieved transmitted clocks	–	Higher latency because of additional processing

Audio over Ethernet

COBRANET, Ravenna, AES67, Dante and AVB/TSN

+	High bandwidth	+	Low latency
+	Simple to use	+	Synchronization

Thesis Objectives

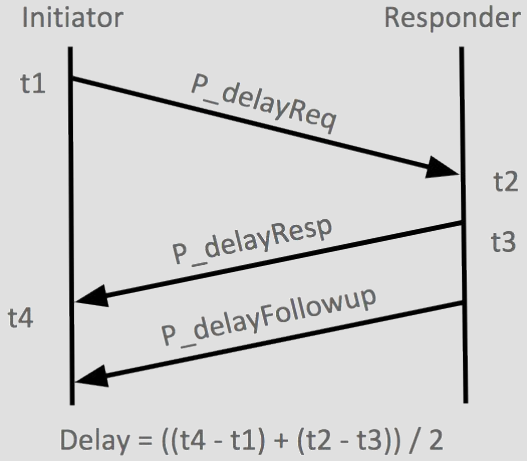
Main goal is to evaluate AVB as a possible candidate for "*Synchronized real time audio streaming over Ethernet in embedded systems*" which can be listed as the following objectives

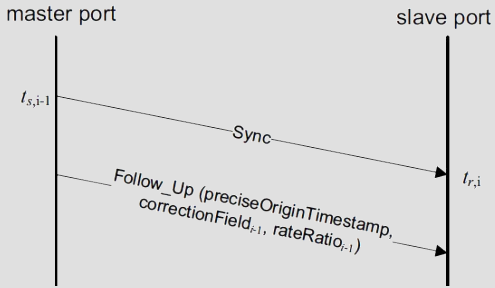
- ★ Study existing literature regarding current available solutions for AVB.
  - ○ Open Avnu
  - ○ Open AVB
- ★ Implement AVB protocol stack in an embedded system
  - ○ BeagleBoard devices
  - ○ Linux based debian OS
- ★ Evaluate the various operational parameters and conclude if the main goal is met.

AVB protocol stack - gPTP

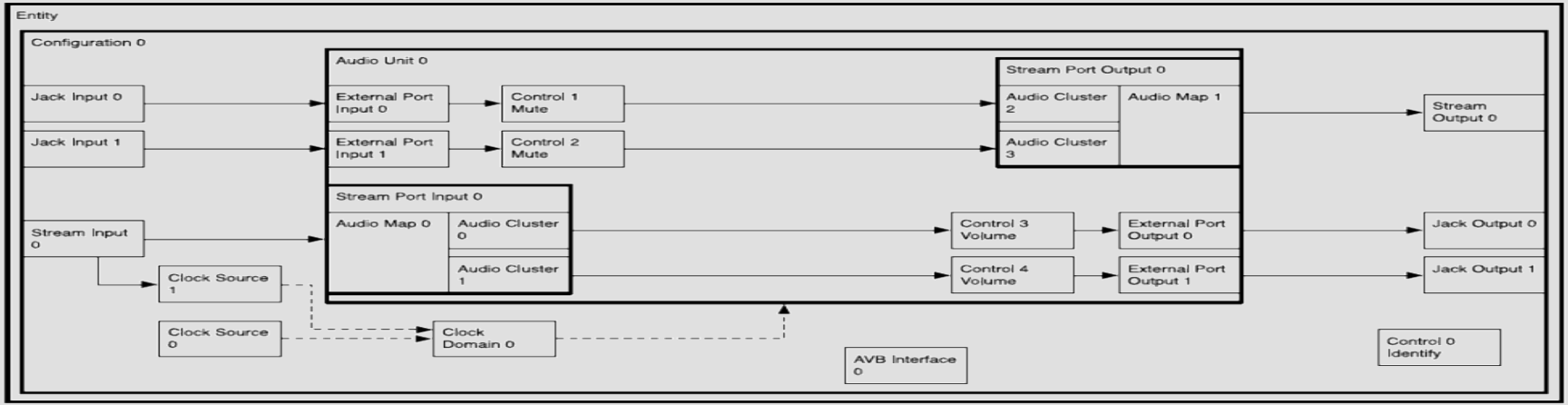
gPTP - Generalized Precision Time Protocol

- ➔ **Responsible for synchronizing the clocks of all devices in the network.**
- ➔ **Propagation delay between every device and it's neighbor is measured.**
- ➔ **A best master clock is selected automatically.**
- ➔ **All devices in the network synchronize to the best master's clock by correcting the received time with the measured delay.**





AVB protocol stack - AVDECC



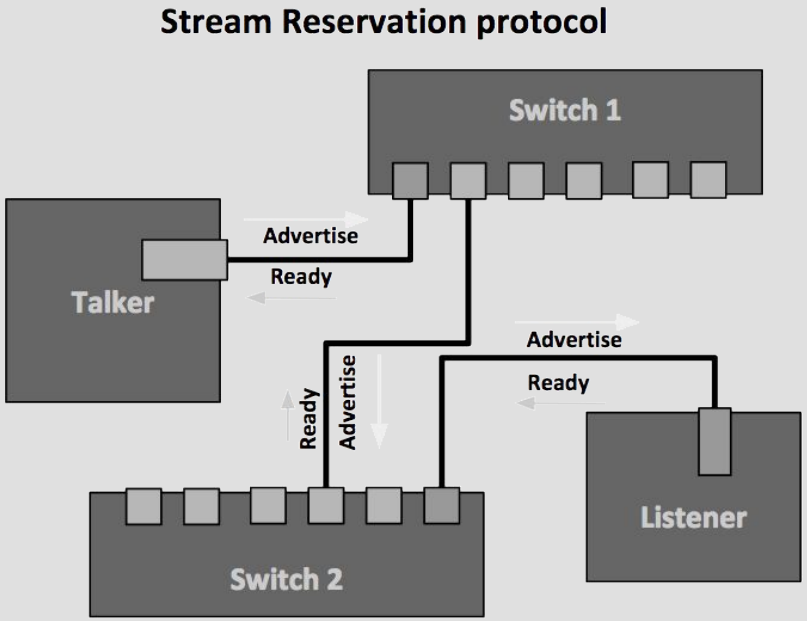
AVDECC - AVB Device Discovery, Enumeration, Connection and Control Protocol

- ➔ Discover other AVB devices in the network (AVDECC Discovery protocol -ADP)
- ➔ Enumerate the capabilities of other devices (AVDECC Enumeration and Control Protocol - AECP)
- ➔ Manage audio connections between the devices(AVDECC Connection Management Protocol -ACMP)
- ➔ Control the streaming and other features of the connected devices such as volume, mixing, equalizing etc… (if supported)(AVDECC Enumeration and Control Protocol - AECP)

AVB protocol stack - MSRP

MSRP - Multiple Stream Reservation Protocol

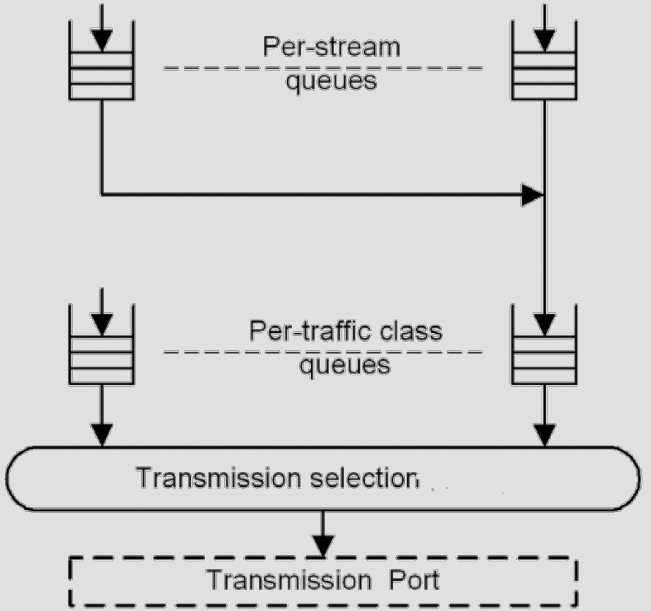
- ➔ Reserve network resources required for a stream such that the required quality of service is achieved.
- ➔ Resources are allocated in all devices in the path between talker and all the listeners.



AVB protocol stack - FQTSS

FQTSS - Forwarding and Queueing for Time sensitive streams

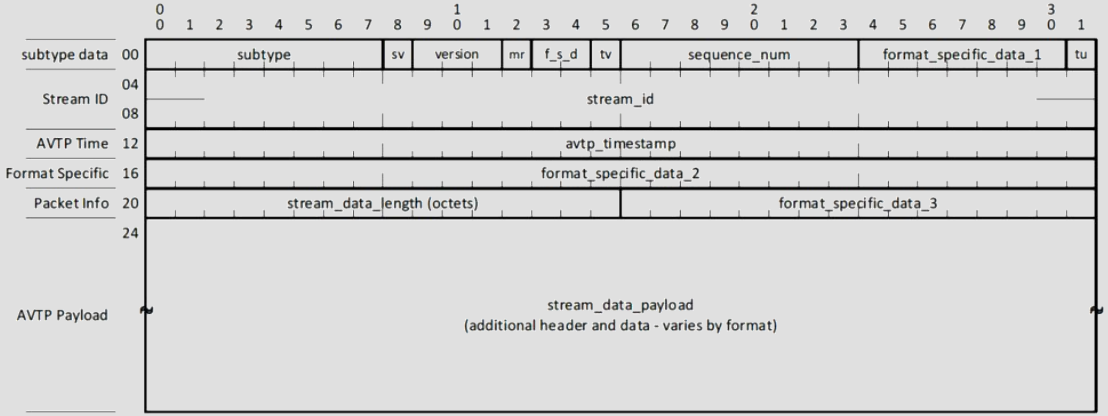
- ➔ **Specifies various priorities for different audio classes.**
- ➔ **Specifies a credit based traffic shaper which prioritises time sensitive data.**



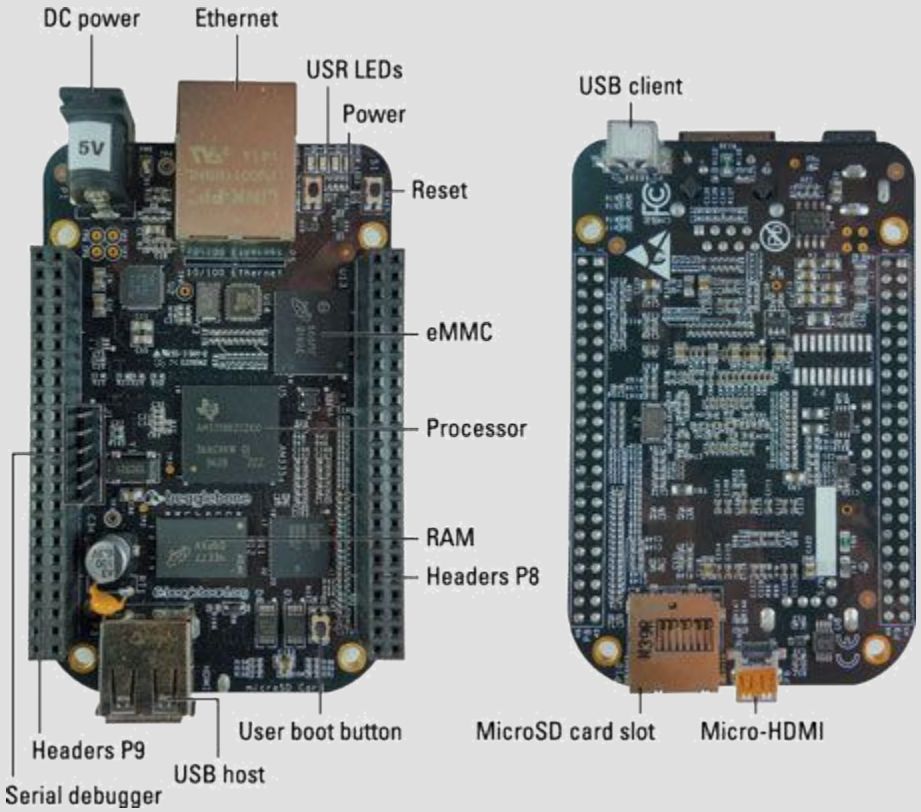
AVB protocol stack - AVTP

AVTP - Audio Video Transfer Protocol

- ➔ **Transfer audio video data as chunks in various formats.**
- ➔ **Transfer the presentation time for each media packet indicating at which time the media has to be presented to the user application.**



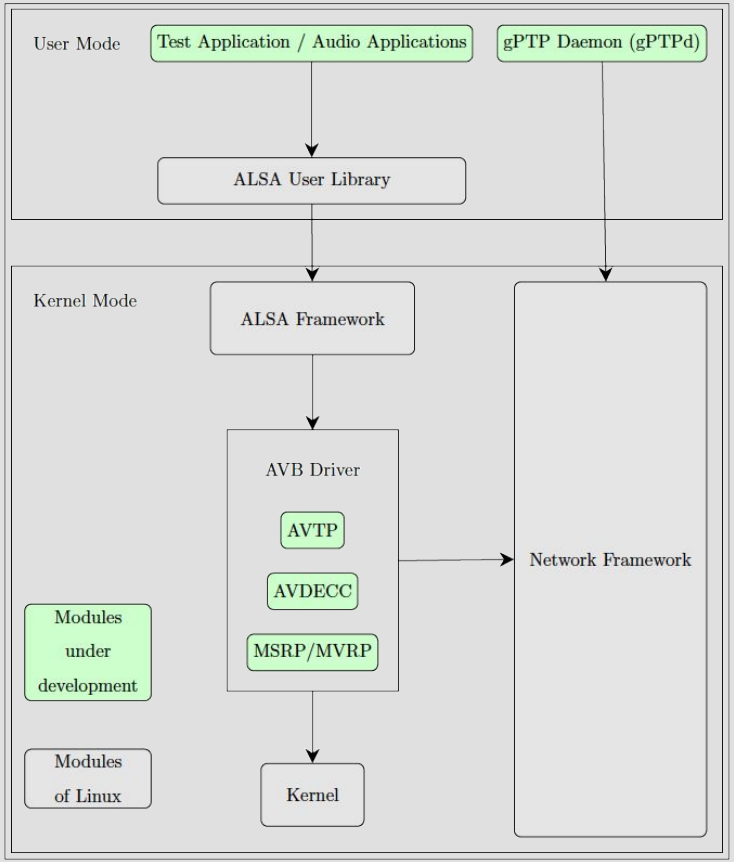
Beagleboard platform



BeagleBone Black is a low cost high performance development platform with the following features

- AM335x Siatra Processor. 2000 MIPS @ 1 GHz
- 500 MB DDR3L @ 800 MHz
- 4 GB embedded MMC onboard
- 2x USB, 1x UART, 1x micro SD card port, 1x HDMI and 1x 10/100 RJ45 Ethernet connector
- 2x 46 pin expansion headers through which up to 4 expansion devices can be connected
- Supports several extension capes through which several new hardware can be attached.

Prototype design



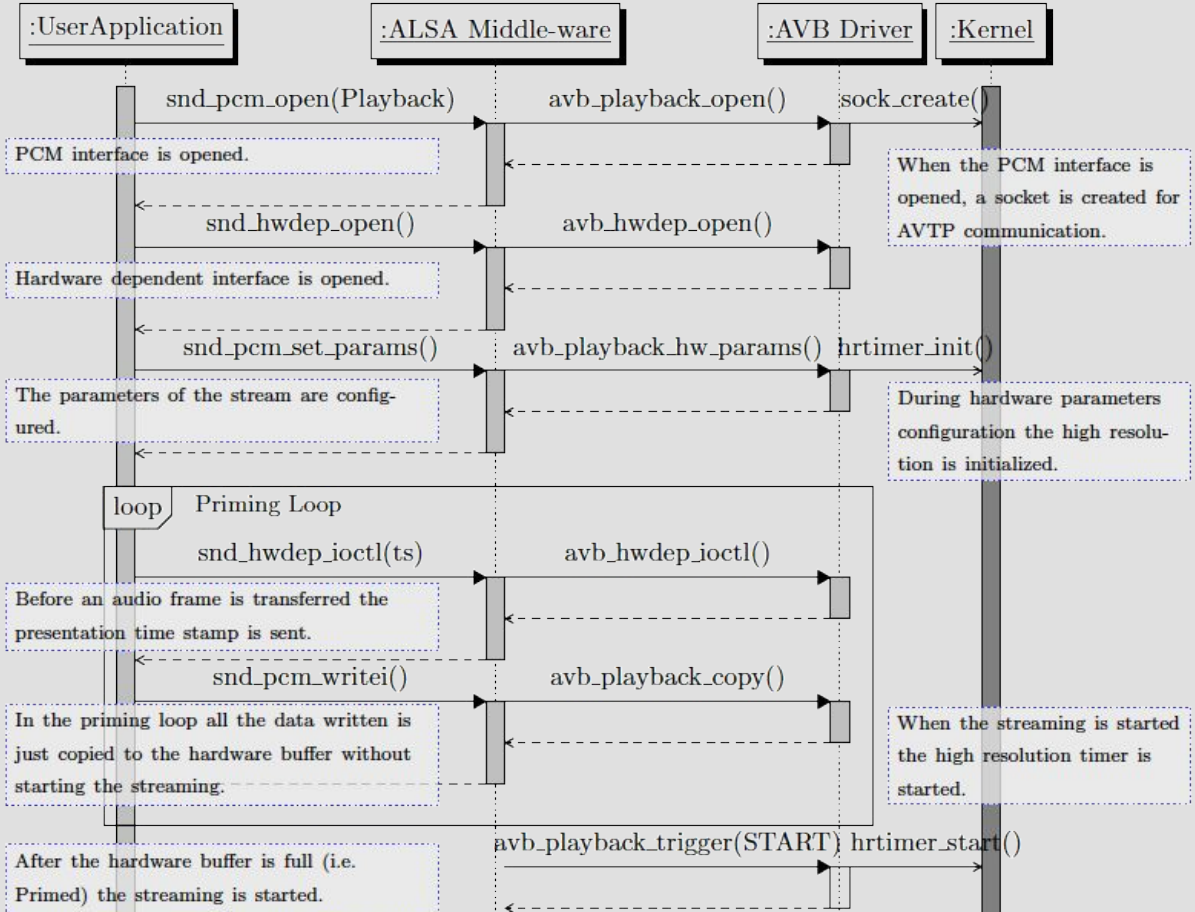
The following modules are developed,

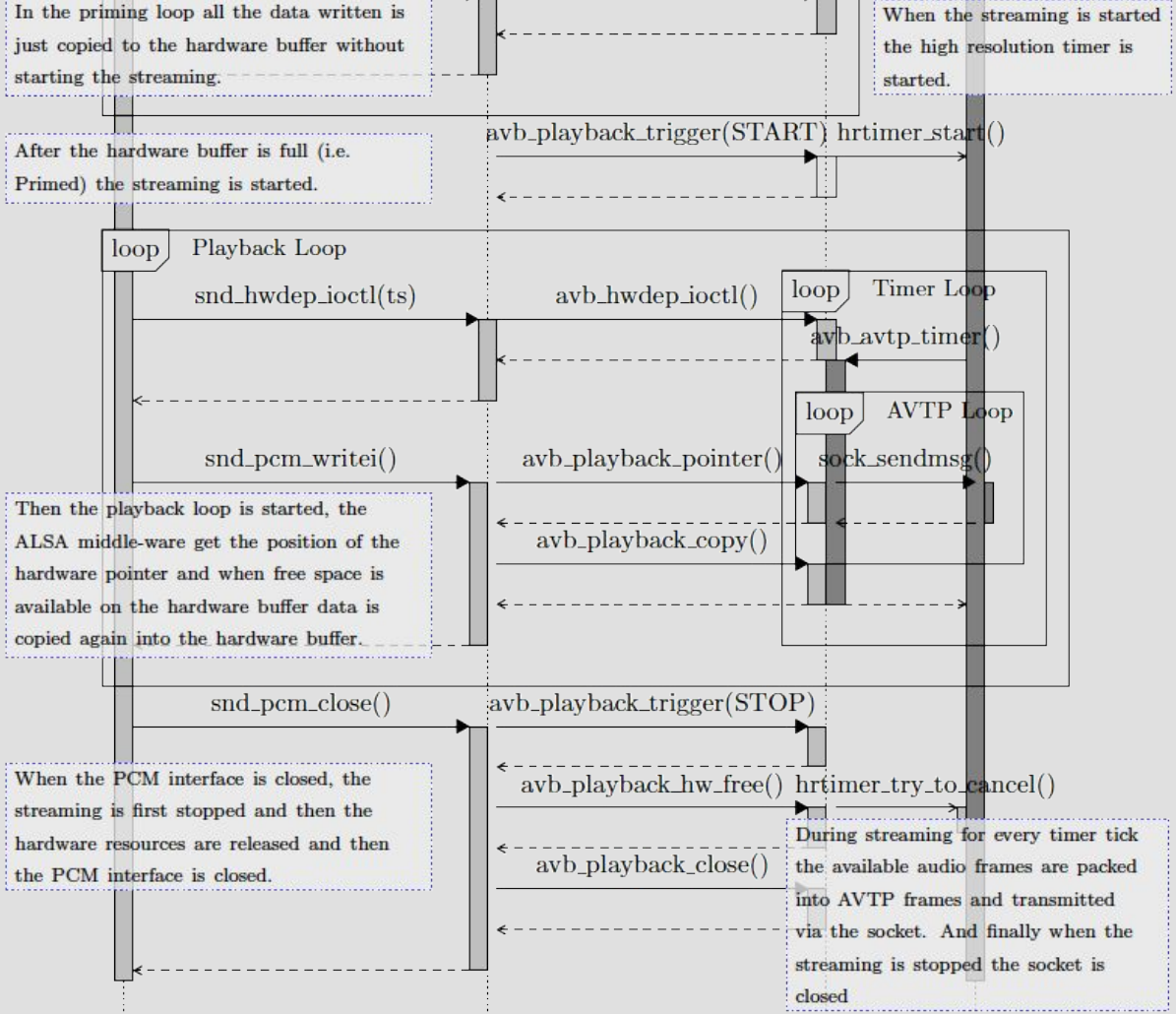
- gPTP Daemon
  - gPTP Implementation
- AVB ALSA Driver
  - AVTP Implementation
  - AVDECC Implementation
  - MSRP Implementation
- Test Application
  - To test AVB protocol suite

Development

Build Host	:	**PC running Ubuntu 15.04**
Cross Compiler	:	**GCC Linaro 6.3.1 ARM Linux gnueabihf**
Beaglebone OS	:	**Debian for beaglebone**
Version Control	:	**Git (Hosted at github.com)**
Debugging	:	**Remote terminal through serial header**
gPTP Daemon	:	**Linux user space daemon application**
ALSA AVB Driver	:	**Linux Kernel space virtual audio device driver**
Test Application	:	**Linux user space application**
Audio Cape	:	**CTAG face 2|4 for beaglebone black**

Development

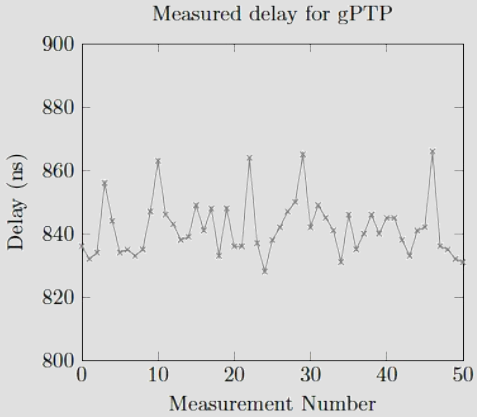



In the priming loop all the data written is just copied to the hardware buffer without starting the streaming.
When the streaming is started the high resolution timer is started.
avb_playback_trigger(START)
hrtimer_start()
After the hardware buffer is full (i.e. Primed) the streaming is started.
loop
Playback Loop
snd_hwdep_ioctl(ts)
avb_hwdep_ioctl()
loop
Timer Loop
avb_avtp_timer()
loop
AVTP Loop
snd_pcm_writei()
avb_playback_pointer()
sock_sendmsg()
Then the playback loop is started, the ALSA middle-ware get the position of the hardware pointer and when free space is available on the hardware buffer data is copied again into the hardware buffer.
avb_playback_copy()
snd_pcm_close()
avb_playback_trigger(STOP)
When the PCM interface is closed, the streaming is first stopped and then the hardware resources are released and then the PCM interface is closed.
avb_playback_hw_free()
hrtimer_try_to_cancel()
avb_playback_close()
During streaming for every timer tick the available audio frames are packed into AVTP frames and transmitted via the socket. And finally when the streaming is stopped the socket is closed

Evaluations

gPTP Delay Measurement Stability

Variations in the successive measured delay values from the gPTP daemon in beaglebone black.



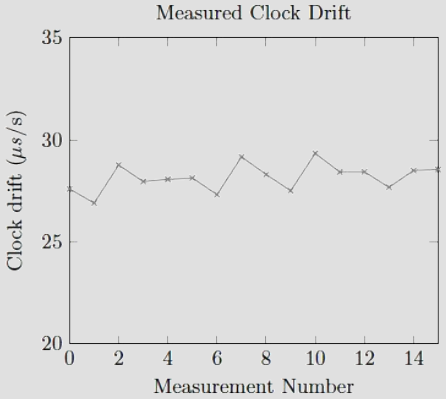
Mean Delay: 841.88 ns

Standard deviation: 8.89 ns

Variance: 79.05 ns

gPTP Clock Drift

Variations in the difference between the local clock to the master clock for every periodic sync command.

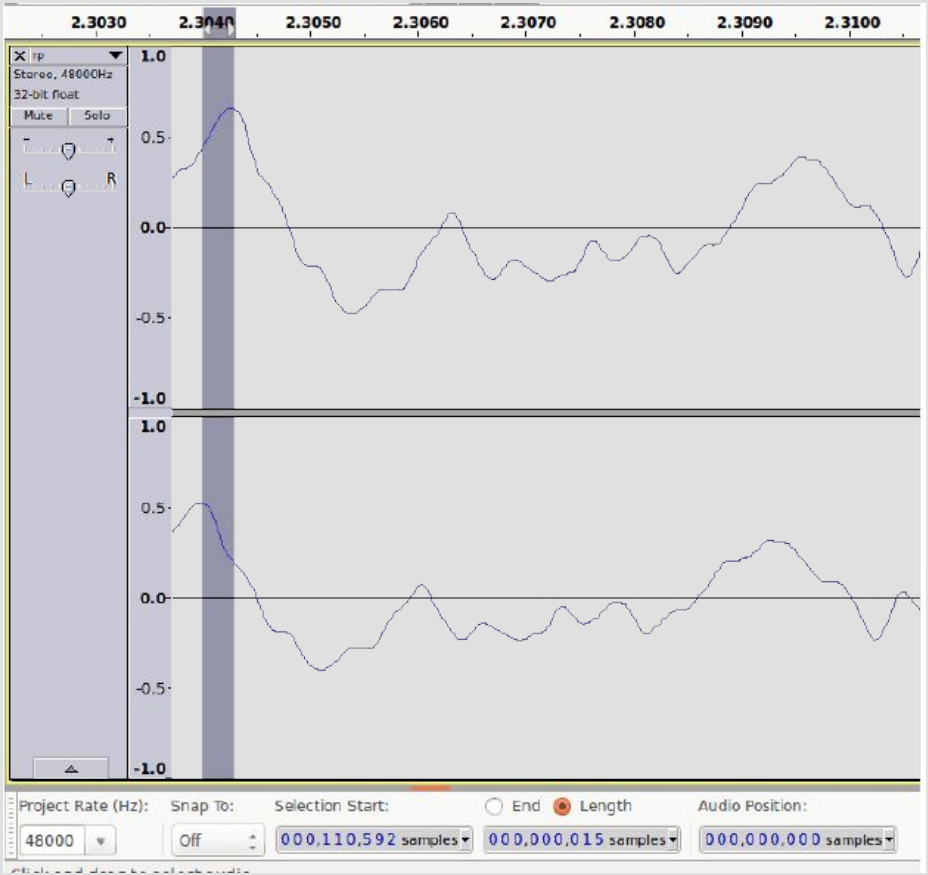


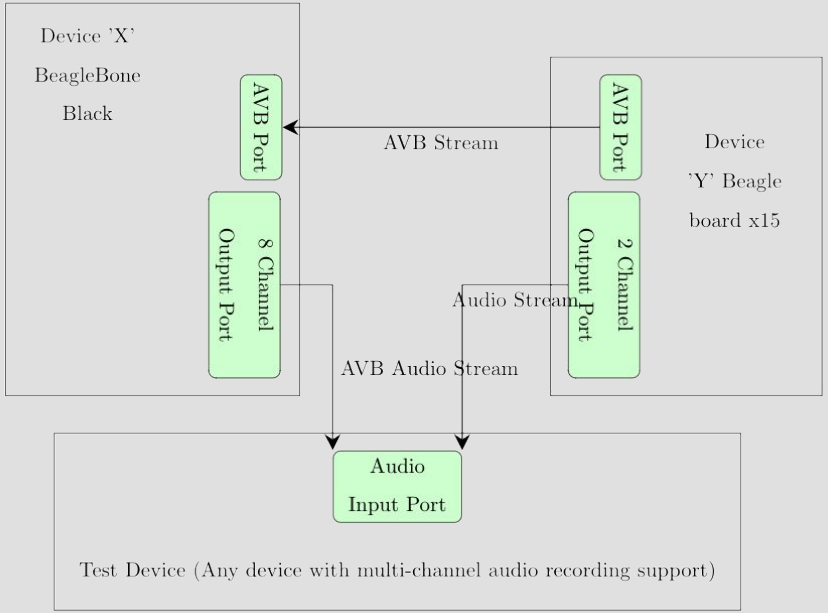
Mean Drift: 28.17 $\mu s/s$

Standard deviation: 0.637 $\mu s/s$

Variance: 0.406 $\mu s/s$

Evaluations



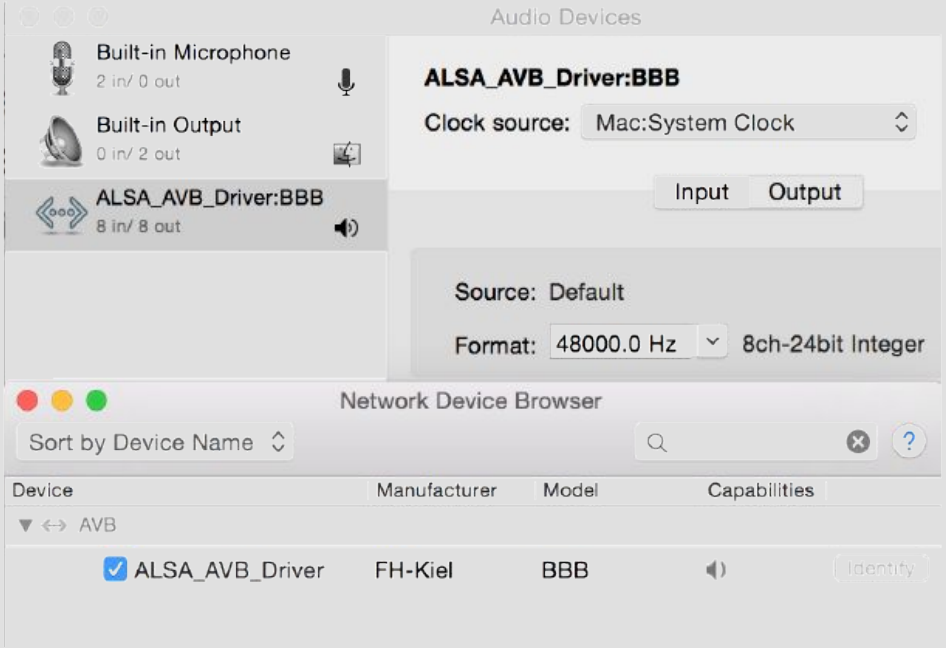


- The measured synchronization error between both devices is 312.5 us

Evaluations

- Latency was measured using the test application.
- The latency can be modified by varying the size of the audio buffer inside the ALSA AVB driver.
- The minimum latency measurable was 18.46 ms

The AVB stack developed is able to be detected by a macbook pro computer as a AVB device.



Limitations

- Maximum of 8 channels and maximum sampling rate of up-to 192kHz is supported.
- Only one playback and capture stream is possible in parallel.
- The Forwarding and Queuing for Time-Sensitive Streams (FQTSS) improvement for the networking queues are not implemented.
- Only the AVDECC responder role is implemented, but does not support the AVDECC controller role.

Conclusion

- Synchronized, real time audio streaming is possible in embedded systems using the proposed system.
- Sufficient for consumer applications.
- Might not be suitable for professional applications.

DEMO