What's next for Bluetooth in PulseAudio?

Sanchayan Maity

asymptotic.io

What is PulseAudio?

- Network capable sound server program
- Initial release on 17 July 2004
- First appeared for users in Fedora Linux with version 8

Bluetooth Profiles

- What is a profile?
- Some of the widely used bluetooth profiles
  - Advanced Audio Distribution Profile (A2DP)
  - Audio/Video Remote Control Profile (AVRCP)
  - Hands-Free Profile (HFP)
  - Headset Profile (HSP)

Bluetooth Codecs

- Available codecs?
  - Low-complexity subband codec (SBC)
  - Audio Processing Technology (aptX, aptX-HD)
  - Advanced Audio Coding (AAC)
  - LDAC
- So which is better?

Current State

- Upstream only supports SBC :(
- HSP/HFP support via ofono
- Out of tree community effort with https://github.com/EHfive/pulseaudio-modules-bt
- In progress work by Pali Rohár

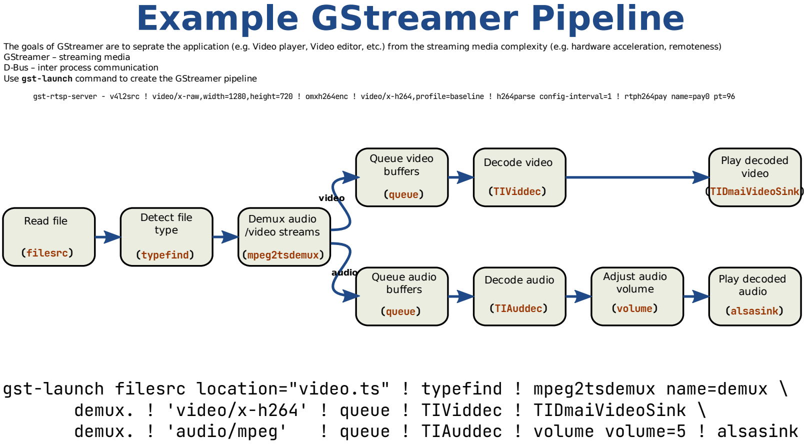
Challenges

- Profile and codec switching
- Patent encumbered licenses?
- How to support multiple encoders or decoders?

GStreamer

- What is GStreamer?
- Why?

Example Pipeline



Example GStreamer Pipeline

The goals of GStreamer are to seprate the application (e.g. Video player, Video editor, etc.) from the streaming media complexity (e.g. hardware acceleration, remoteness)
GStreamer – streaming media
D-Bus – inter process communication
Use **gst-launch** command to create the GStreamer pipeline

```
gst-rtsp-server - v4l2src ! video/x-raw,width=1280,height=720 ! omxh264enc ! video/x-h264,profile=baseline ! h264parse config-interval=1 ! rtph264pay name=pay0 pt=96
```

Read file (filesrc)
Detect file type (typefind)
Demux audio /video streams (mpeg2tsdemux)
video
Queue video buffers (queue)
Decode video (TIViddec)
Play decoded video (TIDmaiVideoSink)
audio
Queue audio buffers (queue)
Decode audio (TIAuddec)
Adjust audio volume (volume)
Play decoded audio (alsasink)

```
gst-launch filesrc location="video.ts" ! typefind ! mpeg2tsdemux name=demux \
      demux. ! 'video/x-h264' ! queue ! TIViddec ! TIDmaiVideoSink \
      demux. ! 'audio/mpeg'   ! queue ! TIAuddec ! volume volume=5 ! alsasink
```

Figure 1: GStreamer Pipeline

Progress so far

- LDAC support upstreamed in GStreamer
- Support for Low-Overhead MPEG-4 Audio Transport Multiplex (LATM) AAC in GStreamer
- GStreamer wrapper around Pali Rohár's *libopenaptx*
- Implementing profile and codec switching
- Proof of concept with GStreamer tested

Conclusion

What's next?