The In and outs of Wireless Audio

Noel McKenna
Senior Director of aptX
Background

- Joined APT in 1994 as development engineer.
- Led the MBO in 2005 and the introduction of IP licensing within APT,
- Split Hardware and Licensing into 2 separate business units in 2009.
- Sold Hardware business to Audemat in 2009.
- Sold APT Licensing to CSR in 2010.
- Responsible for CSR aptX audio compression rollout.
CSR Background

- Founded 1999
- Bluetooth, Wifi, GPS, FM radio technologies.
- Kalimba DSP platform.
- Fabless semiconductor design company.
- Acquired SIRF in 2009 for GPS
- Acquired aptX in Belfast in Aug 2010
- Around 3000 people with Zoran merger
- $800M revenue (2010)
Introduction

Why is audio important for wireless devices?

Bluetooth – a typical wireless use case.

The implications on codec selection.

Latency and wireless audio.
Why is audio important for wireless devices?

- Audio is one of the most common forms of data transferred between portable devices
  - Speech is an obvious use case
  - Music is becoming more popular
- Wireless ecosystems are becoming more complex
  - Multiple simultaneous links between devices.
  - Games console, phone, headset, controllers.
  - A connectivity centre for multiple use cases.
Different forms of audio streaming using a wireless link...

- Broadcast (e.g. Digital Radio)
- On demand listening (e.g. Spotify)
- Cellular conversations (e.g. Mobile phone)
- Multi-room music distribution (e.g. Sonos, Airplay)
- Ultra low latency streaming (e.g. Microphones)
- Personal music network (e.g. Bluetooth)
Two tier distribution

Wide

Local
Local ecosystem
Wireless audio definition...

- Local streaming
- Multiple connections
- Overlap with wider networks
- Distinct requirements
- Voice and music
Music streaming...
Streaming or downloading?

Do we want a continuous stream with low latency or a fast and efficient method of transferring the data?
Audio quality is defined “as good as a wire”.

- Just use PCM?

Ideal solution:

- Zero power consumption
- Perfect audio quality
- Zero latency
- Very low cost… (cable is cheap)
- Has value add – passive to active
General data transfer

Streaming

Time

© CSR plc 2008. All rights reserved.
Why do we need codecs?

Efficient transfer of audio data is essential
  ~ Bandwidth limitations

This affects:
  ~ Complexity
  ~ Battery life
  ~ Perceived audio quality
  ~ Latency
  ~ Transcoding effects
  ~ Cost…
Coding for wireless transfer

The codecs that are part of the Bluetooth A2DP specification are based on existing audio compression codecs.

These are targeted for storage, such as MP3, AAC, WMA etc.

Primary tradeoff is data rate/compression/complexity against audio quality.

Encoders can be substantially more complex than decoders.

Wireless transfer requires a different set of parameters to be considered…
A wireless audio stream has inherent latency.
The combination of the audio codec and the wireless stream protocol must maintain low latency.
Data rate, compression, complexity and audio quality are still important.
Encoder and decoder should be matched in terms of complexity.
A wireless audio stream is susceptible to radio errors, the codec should be resilient against this.
Coding for Bluetooth

There are two options for source devices:

- **Native streaming**
  - ✓ No additional processing
  - ✓ Low delay
  - ❌ No local integration of sounds
  - ❌ Receiving device needs to support all the codecs of the source
  - ❌ Interface logic is complex, different codec negotiation per device/use case

- **Transcoding**
  - ✓ Device performs “as normal” for decoding
  - ✓ Interface logic is simple, wired or Bluetooth
  - ❌ Additional processing and delay
The Bluetooth use case – native transfer

SOURCE

Encoded Audio

Frame 1  Frame 2  Frame 3

Packet 1  Packet 2

SINK

Packet 1  Packet 2

Frame 1  Frame 2

Decoder

PCM Audio

© CSR plc 2008. All rights reserved.
The Bluetooth use case – common transfer

**SOURCE**

Encoded Audio

Frame 1  Frame 2  Frame 3

Encoder

Decoder

PCM Audio

New Encoded Audio

Frame 1  Frame 2  Frame 3

Packet 1  Packet 2

**SINK**

Packet 1  Packet 2

Frame 1  Frame 2

Decoder

PCM Audio

© CSR plc 2008. All rights reserved.
Bluetooth latency

System latency is accumulated at the following points:

- Transcoding processing at source
- Frame size of codec
- Frame to packet misalignment
- Bluetooth transmission
- Robustness buffering at receiver
  - Frame to packet misalignment
  - Jitter in packet arrival time
  - Retransmissions
  - Rate matching
- Decoding processing
Low Latency (Fast Stream)

- Lower latency for games and “lip sync” applications
- Video Lip Sync requires a latency below 40ms
- As aptX is a sample based codec, low latency can be achieved through efficient population of packets while retaining transmission robustness
- 32 ms latency from the Audio Adapter
- Requires CSR devices at both ends of the link
- A2DP Vendor specific codec implementation
Frame and Packet Size

1 SBC packet = 72 bytes = 84 audio samples @ 300kbps
1 aptX word = 2 bytes = 4 samples

Option 1:
- SBC Frame 1
- SBC Frame 2
- SBC Frame 3
- SBC Frame 4

Option 2:
- SBC Frame 1
- SBC 2
- SBC Frame 3
- SBC Frame 4
- SBC 5
- SBC Frame 6

Inefficient use of Packet Stuffing

SBC only starts to decode when complete Frame has arrived

Time / MIPS on splitting and re-assembling Frame

Time / MIPS on splitting and re-assembling Frame

Note: Detail above is for illustrative purposes and not to scale.
Frame and Packet Size Continued....

Very efficient use of Packet Stuffing

aptX aptX aptX aptX
aptX aptX aptX aptX
aptX aptX aptX aptX
aptX aptX aptX aptX

Etc, etc, etc...

Start to decode when aptX word arrives
Scalable coding

- Bandwidth over-the-air
  - Reduce/increase the data rate
- Audio quality
  - Expend resources achieving excellent quality when it is applicable.
- Complexity
  - Reduce MIPS when possible and necessary
- Error handling
  - When interference detected, adapt the coding scheme to compensate
- Latency
  - Algorithmic delays vary when required
Scalable coding – “sweet spot”

Latency

Quality

Scalable Coding

MPEG-4

MPEG-2

Layer 3

aptX

SBC

© CSR plc 2008. All rights reserved.
Scalable coding - capabilities

- Scalable latency
- Scalable complexity
- Scalable bit rate
- Scalable error resilience
- Configurable stream structure
- Dynamic and compile-time reconfiguration
- Cognitive adaptation capability
- Backwards compatibility with SBC and aptX
Scalable coding – algorithm adaptation

Audio coding algorithm:

Suite of coding tools:

© CSR plc 2008. All rights reserved.
Scalable coding – power consumption
Scalable coding – system constraints
Scalable coding – synchronization
Scalable coding use case – music stream

© CSR plc 2008. All rights reserved.
Scalable coding use case – multi-stream
Scalable coding use case – Dualstream
Scalable coding use case – 2.1
Scalable coding use case – 5.1
Post processing

Audio stream must be “as good as a wire”
- No audible artefacts of the transcoding for wireless transfer.
- Some codecs introduce spectral holes.

Must handle errors due to interference or lost packets
- Temporal holes must be masked/filled.

Must allow post processing of audio
- Codec compensation
- Equaliser
- Dynamic range compression
Audio for wireless microphones

Professional:
- Proprietary transmission required to compete with wired microphones
- Low latency and high quality are paramount
- Restricted bandwidths require data compression
- Enhanced level of error tolerance is necessary
Audio for wireless microphones

Consumer:

~ Bluetooth can offer quality and latency suitable for A/V and interactive multimedia
~ Extend Bluetooth functionality required by many devices for live microphone performance
~ CSR knowledge of entire Bluetooth audio streaming chain provides unique ultra low latency solutions
Summary

- Wireless audio streaming is not just moving data from device A to device B.
- Must consider the use case, audio content, environment (radio and acoustic), RF ecosystem...
- Use cases overlap, so will RF technologies.
- Coexistence is a significant challenge.
QUESTIONS?