RTP Payload Format for the CELT Codec

draft-valin-celt-rtp-profile-00.txt

IETF 74 — March 2009

Greg Maxwell, Jean-Marc Valin

gmaxwell@juniper.net, jean-marc.valin@octasic.com

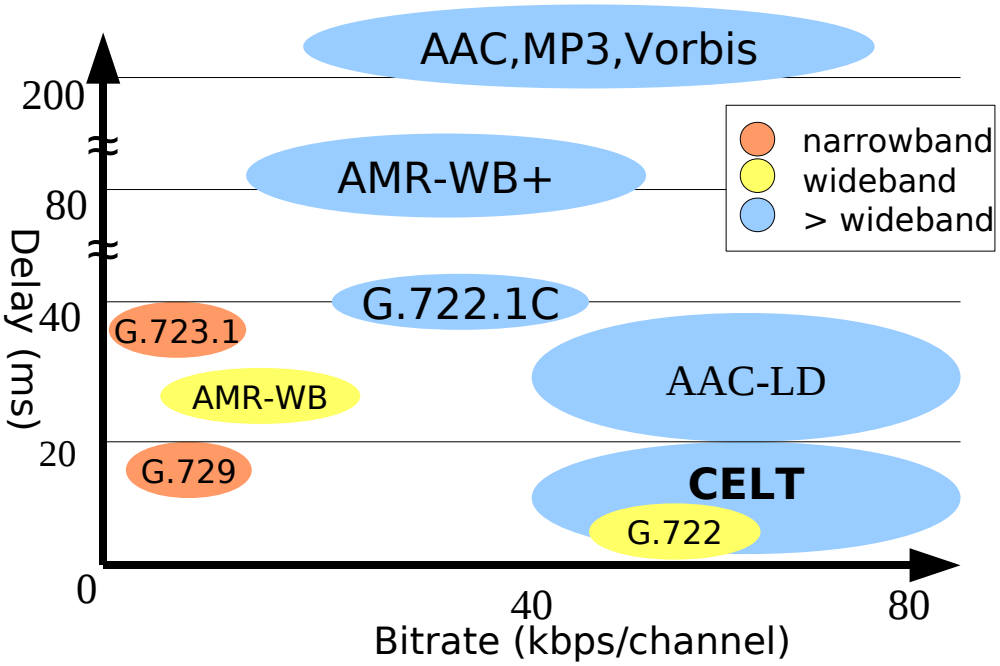
About CELT…

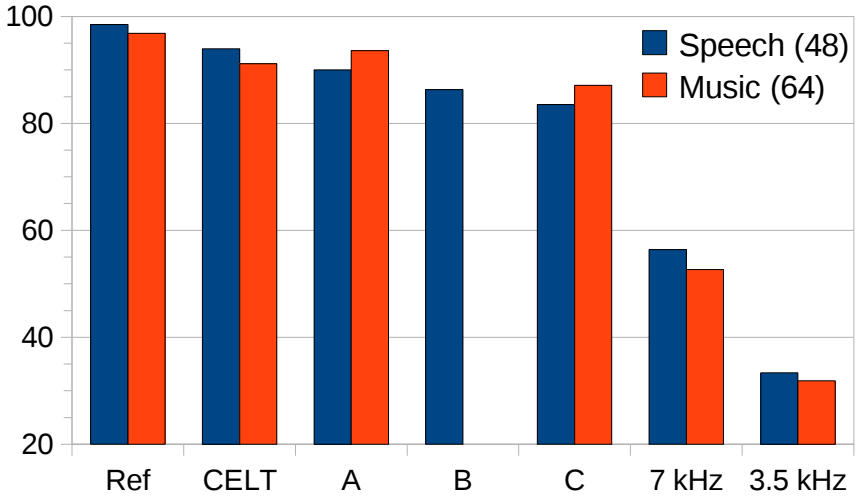
- There are two main classes of audio codecs
  - Speech codecs with low to medium quality and low delay
  - Music codecs with high quality and high delay
- CELT aims for both high quality and ***very*** low delay
  - Prevents collisions during conversations (higher sense of presence)
  - Reduces or remove the need for acoustic echo cancellation
  - Allows synchronization for live music performance
- Perceptual transform (MDCT) codec
- Developed within the Xiph.Org Foundation
- Reference implementation is open source (BSD-licensed)
- No royalties, avoids known patents in the field

CELT characteristics

- Sampling rates from 32 kHz to 96 kHz
- Total algorithmic delay from 2 ms to 24 ms (8 ms typical)
- Frame sizes from 64 samples to 512 samples
- One or two channels of audio encoded into a single frame
- Error and Loss robustness
  - Monotonically decreasing 'bit importance'
- Signaling-free on-the-fly rate adjustment
- Bit-stream ''not frozen yet''

Audio codec landscape





Codec behavior impacting the draft

- The decoder **MUST** know
  - The sample rate the sender is using
  - The codec frame size the sender is using
  - The length of each compressed codec frame
  - If the encoded frame codes for one or two channels
- Of these only the compressed length should reasonably change frame to frame
- Sample rate, frame size changes require somewhat computationally expensive setup

Frame size

- Power of two sizes give the best performance
  - Embedded implementations may only support some sizes
  - Single frame size concurrently
- External factors often drive frame size preferences
- Current draft negotiates using fmtp and requires the answerer will respond with a single supported size and presumes it will send with that rate
  - This has early media issues

Channel mapping

- Indicates the grouping of audio channels into CELT frames and how the channels are used
- Not all receivers will support multi-channel reception
- Common use cases would have asymmetric configurations
  - Stereo down to conference bridge clients, mono up
- Current draft is simply broken in this regard
  - SDP signals a 'mapping' parameter
  - If its used like a 'sprop' there is no way to indicate receiver capability
  - Change to having separate capability and sender mode attributes
    - Early media problems

Compressed length

- CELT can output any requested number of bytes
- Support for multiple CELT frames per packet requires signaling the distribution of bytes to frames
- Signaled in-band
- CELT compressed lengths at the start of each RTP
  - Most common case is short lengths
  - Lengths under 255 bytes use a single byte
  - Longer lengths encode a 0xFF for each 255 bytes of payload then another byte with the remaining length.
- No issues with this approach?
- Is the low overhead in the draft mode worthwhile?

Common SDP attributes

- ptime
  - Profile treats this as a receiver requested minimum packetization interval only
- b=AS:
  - Profile treats this as a receiver requested maximum bitrate
- These are the simple, conventional uses, no codec interaction
- No issues here?
- Some implementers appear to have incorrect beliefs about ptime

Open issues

- Early media issues with current negotiation approach
  - The offerer could use distinct payload types with single configurations
  - How acceptable is it to burn payload types for this?
  - Also send in-band?
    - Could be done without continual overhead
- Re-invite not addressed
  - Obvious solution is to recommend different payload types be used when sender parameters change

Future work

- CELT would be more flexible with configuration data (~100 bytes)
  - Expected all receivers would support all configuration data
  - Configuration packet transmitted at regular interval?
  - Incrementally transmitted in-band?
  - Base64-encoded in SDP parameters?
- Freezing the CELT bit-stream