Opus, a free, high-quality speech and audio codec

Jean-Marc Valin, Koen Vos,
Timothy B. Terriberry, Gregory Maxwell

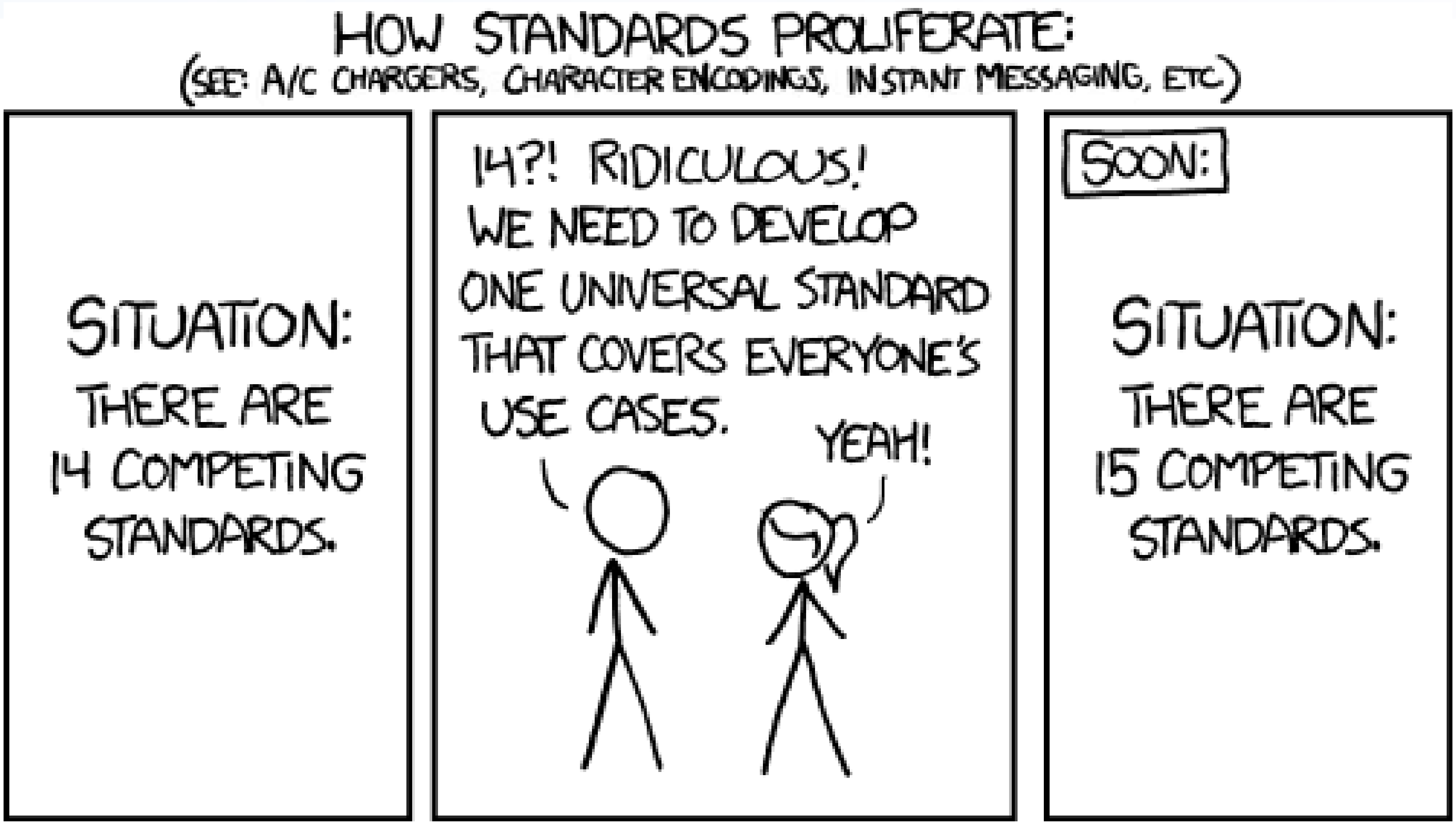
29 January 2014

Xiph.Org & Mozilla

What is Opus?

- New highly-flexible speech and audio codec
  - Works for most audio applications
- Completely free
  - Royalty-free licensing
  - Open-source implementation
- IETF RFC 6716 (Sep. 2012)

Why a New Audio Codec?



http://xkcd.com/927/
http://imgs.xkcd.com/comics/standards.png

Why Should You Care?

- Best-in-class performance within a wide range of bitrates and applications
- Adaptability to varying network conditions
- Will be deployed as part of WebRTC
- No licensing costs
- No incompatible flavours

History

- Jan. 2007: SILK project started at Skype
- Nov. 2007: CELT project started
- Mar. 2009: Skype asks IETF to create a WG
- Feb. 2010: WG created
- Jul. 2010: First prototype of SILK+CELT codec
- Dec 2011: Opus surpasses Vorbis and AAC
- Sep. 2012: Opus becomes RFC 6716
- Dec. 2013: Version 1.1 of libopus released

Applications and Standards (2010)

Application	Codec
VoIP with PSTN	AMR-NB
Wideband VoIP/videoconference	AMR-WB
High-quality videoconference	G.719
Low-bitrate music streaming	HE-AAC
High-quality music streaming	AAC-LC
Low-delay broadcast	AAC-ELD
Network music performance	

Applications and Standards (2013)

Application	Codec
VoIP with PSTN	Opus
Wideband VoIP/videoconference	Opus
High-quality videoconference	Opus
Low-bitrate music streaming	Opus
High-quality music streaming	Opus
Low-delay broadcast	Opus
Network music performance	Opus

Features

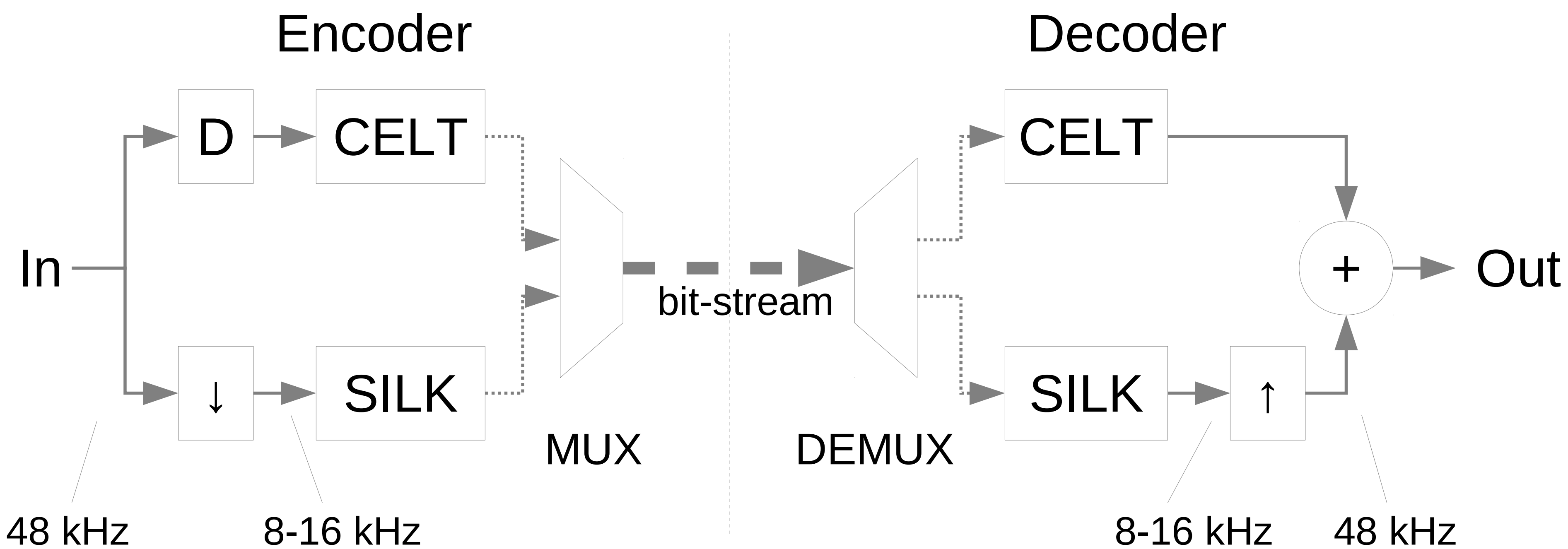
- Highly flexible
  - Bit-rates from 6 kb/s to 510 kb/s
  - Narrowband (8 kHz) to fullband (48 kHz)
  - Frame sizes from 2.5 ms to 60 ms
  - Speech and music support
  - Mono and stereo
  - Flexible rate control
  - Flexible complexity
- All changeable dynamically

Rate Control

- Opus supports true CBR
  - Every packet has the same number of bytes
  - No bit reservoir => no extra delay
  - Quality not as good as VBR
- Constrained VBR
  - Total variation within 1 frame of CBR (same as bit reservoir)
  - Bounded delay, better transients, etc.
- True VBR
  - Open loop: calibrated to a large corpus
  - Gets the most benefit from new encoder improvements
- Bitrate cap possible for both VBR modes

Opus Design

- SILK: Based on voice codec from Skype
- CELT: MDCT codec from Xiph.Org



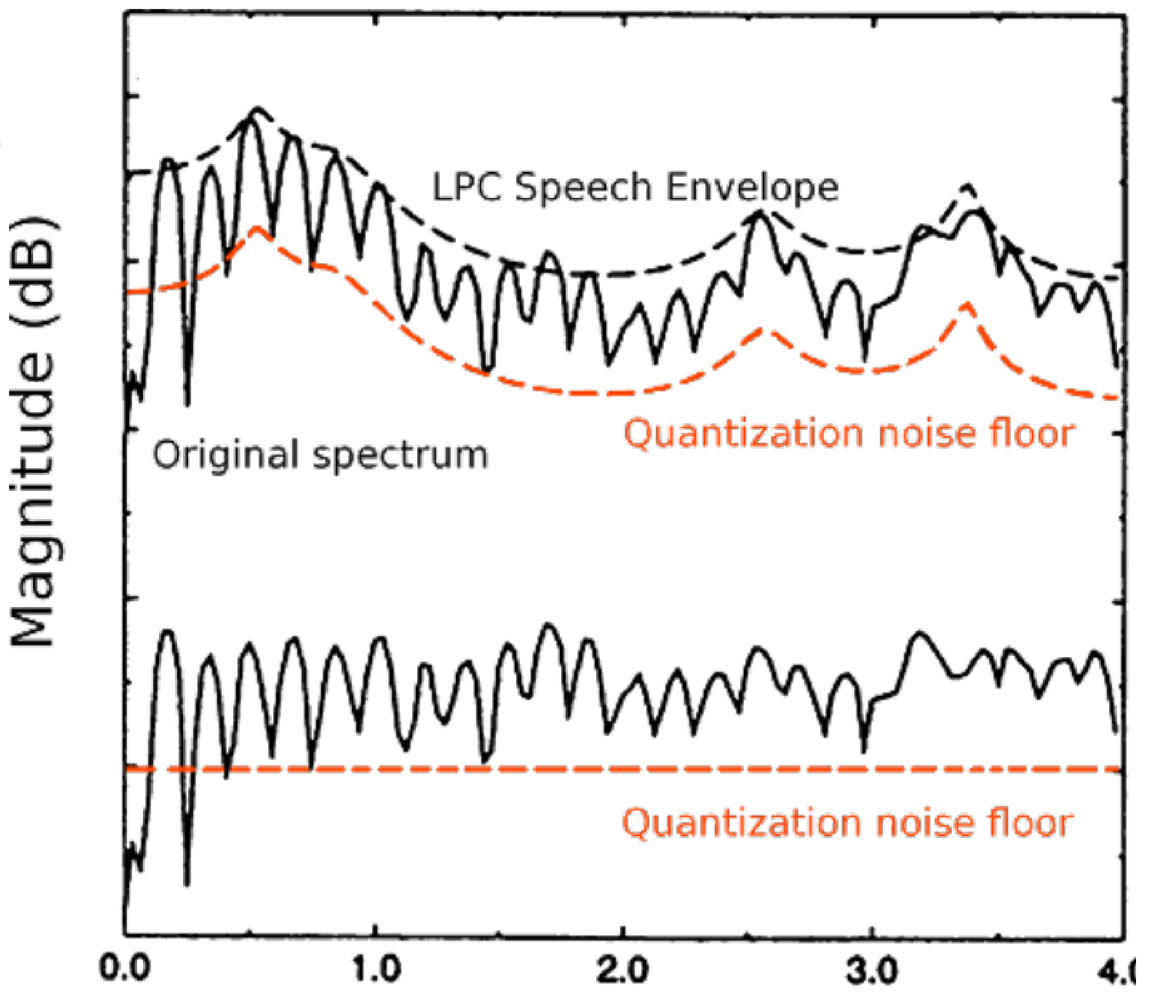
- Better than sum of its parts (Hybrid mode, seamless mode switching)

SILK Component

- Originally used in Skype
- Based on linear prediction (LPC)
- Very good at narrowband and wideband speech up to ~32 kb/s
- Not very good on music
- Heavily modified to integrate with Opus

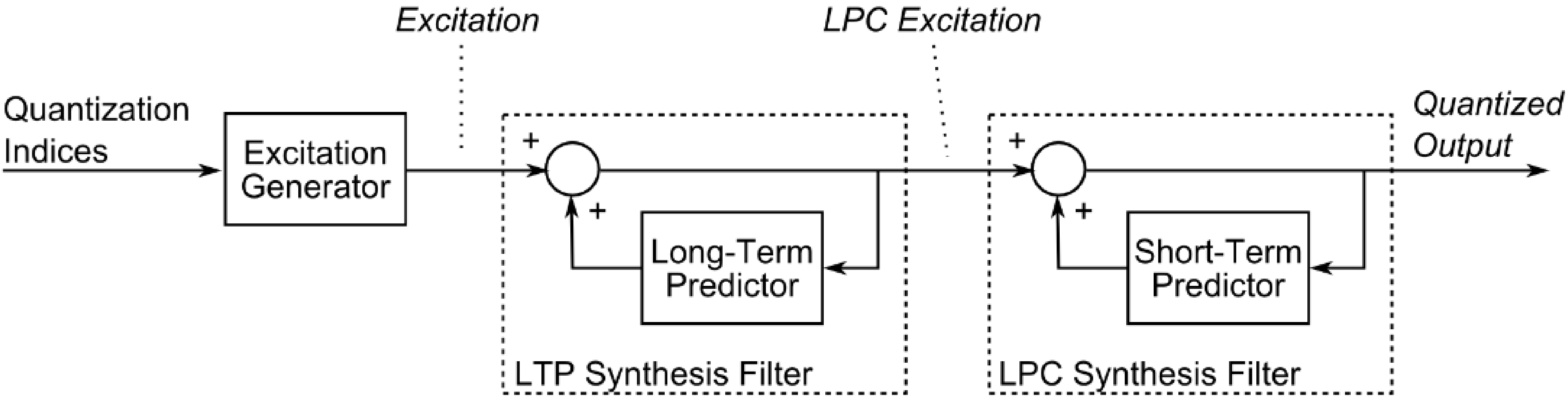
Linear Prediction Crash Course

- All-pole (IIR) filter
- Analysis "whitens" a signal
- Quantization (lossy compression) adds noise
- Synthesis "shapes" the noise the same as the spectrum



SILK Decoder

- Standard defines only the decoder
  - Leaves more flexibility to the encoder

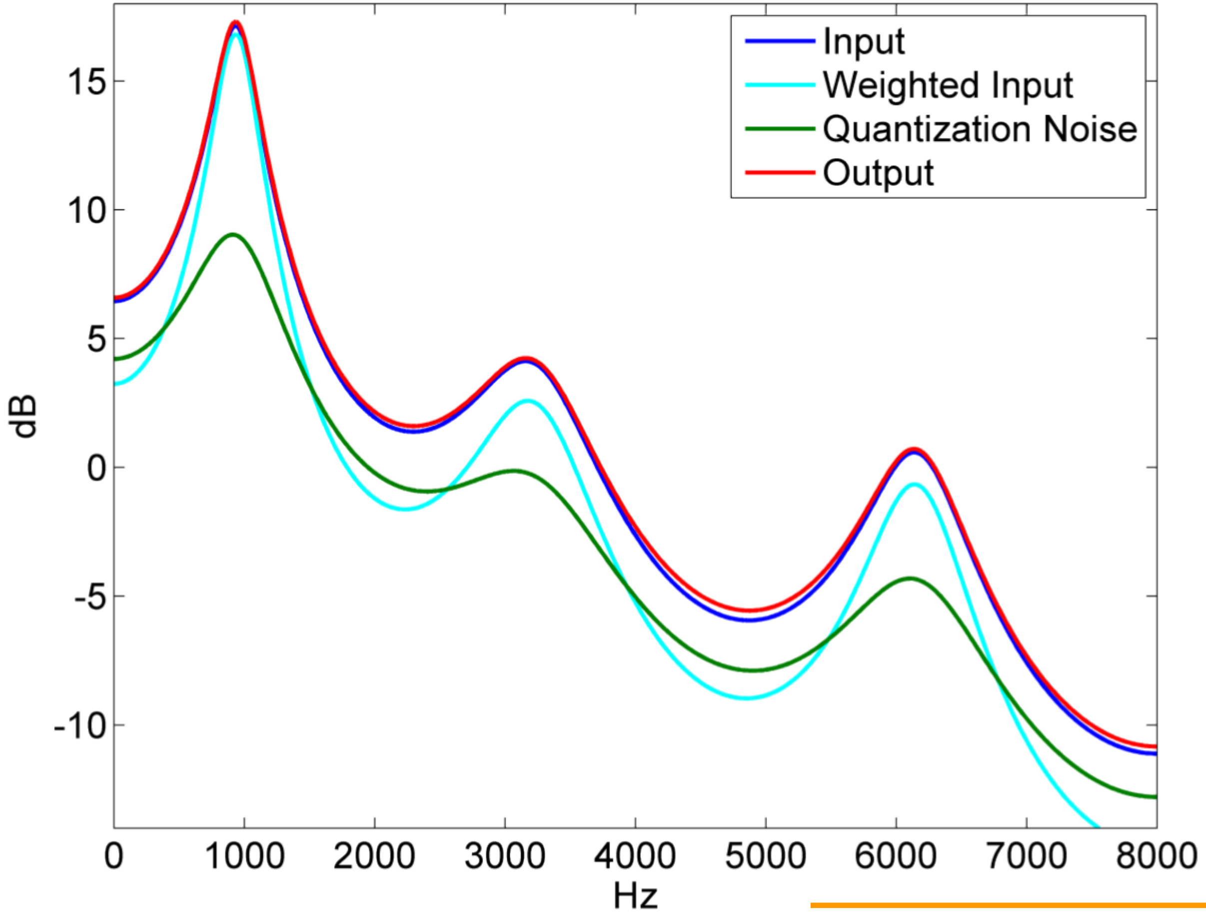


SILK Technology

- Very different from typical CELP codecs
  - Based on Noise Feedback Coding rather than Analysis-by-Synthesis
  - Makes heavy use of entropy coding
    - Decisions are rate-distortion optimized (RDO)
  - Postfilter replaced by a prefilter
  - Smart encoder, very simple decoder

SILK Noise Shaping

- Analysis/synthesis mismatch to de-emphasize spectral valleys



Robustness Features

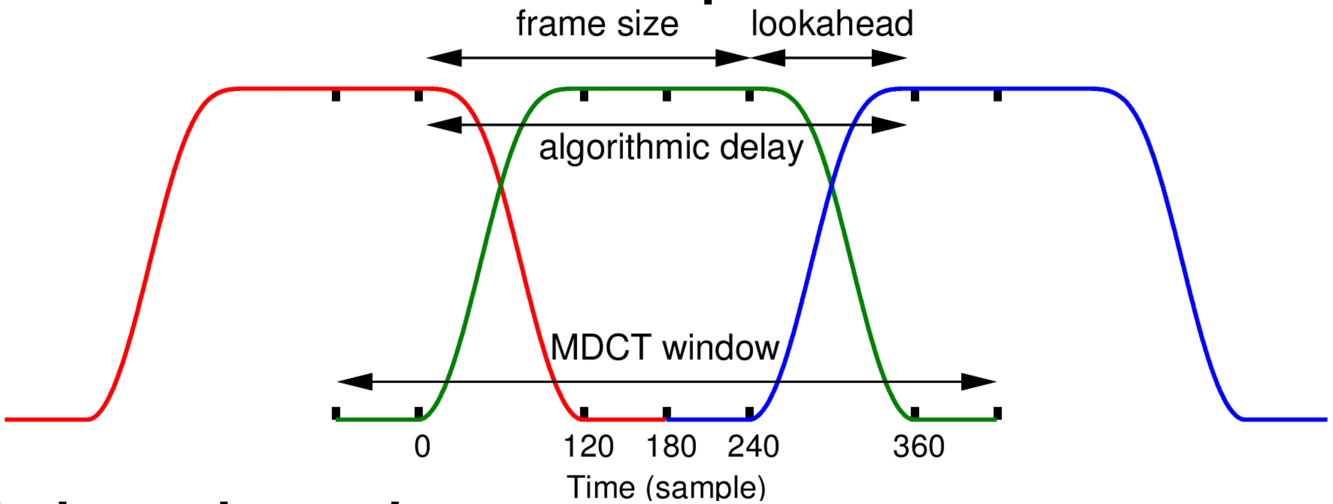
- Flexible prediction
  - Reduces inter-frame dependency at high loss rate
- Packet loss concealment
  - Makes up a plausible packet in case of loss
- Forward error correction (FEC)
  - Optionally includes a low-quality version of the previous packet in case of loss

CELT Component

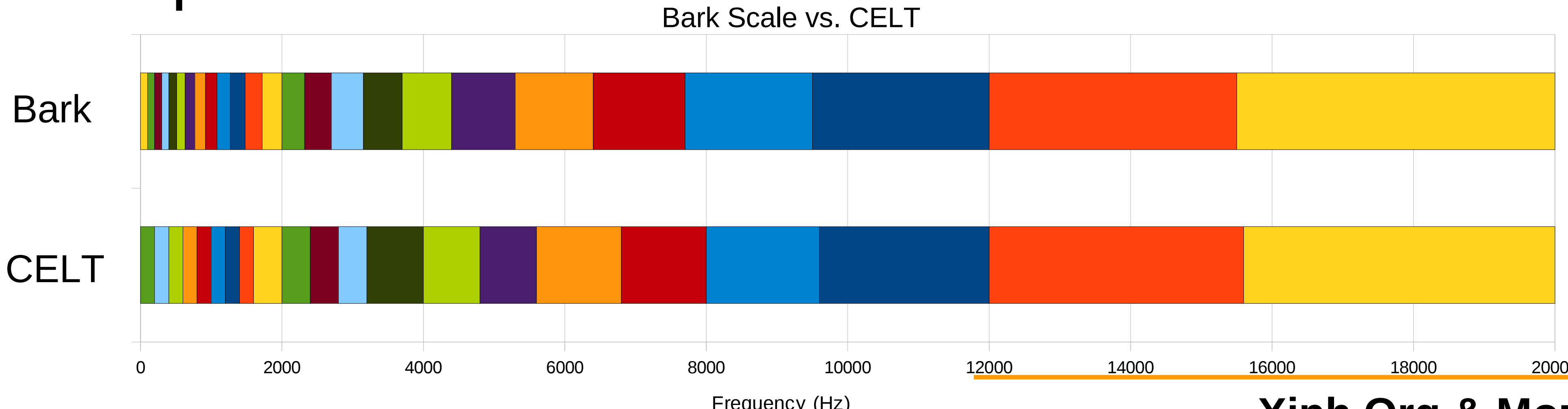
- “Constrained-Energy Lapped Transform”
- Works on speech and music
- Most efficient on fullband audio (48 kHz)
- Scales to ultra-low delay
- Less efficient on low bitrate speech

CELT Transform

- MDCT with low-overlap window



- Split into bands



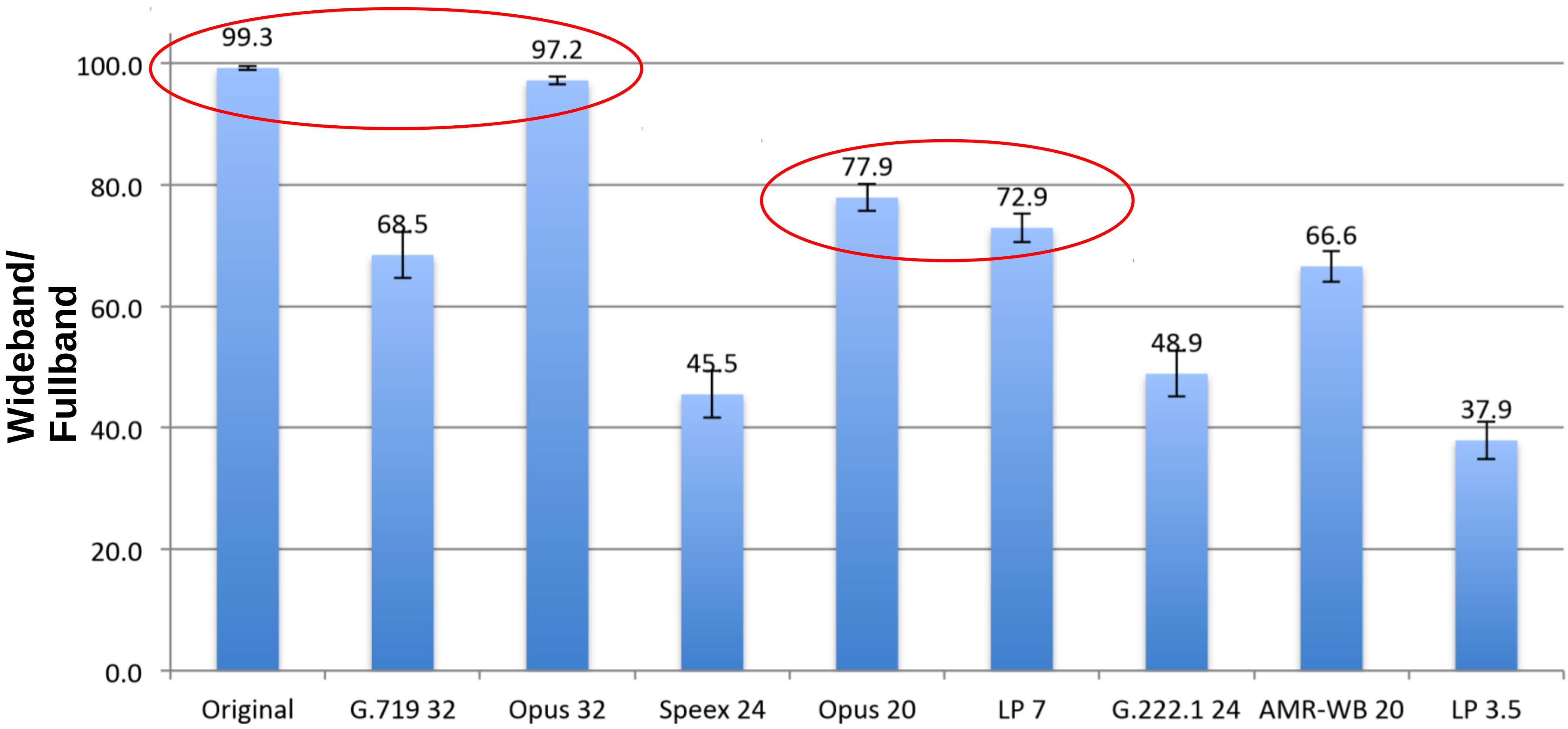
CELT Technology

- Explicitly code/constrain energy of each band
  - Spectral envelope preserved no matter what
- Code remaining details using algebraic VQ
  - Gain-shape quantization
- Implicit psychoacoustics and bit allocation
  - Masking curve built into the format
  - No need to code scalefactors
  - Hard to write a bad encoder
- Several psychoacoustic “tricks”

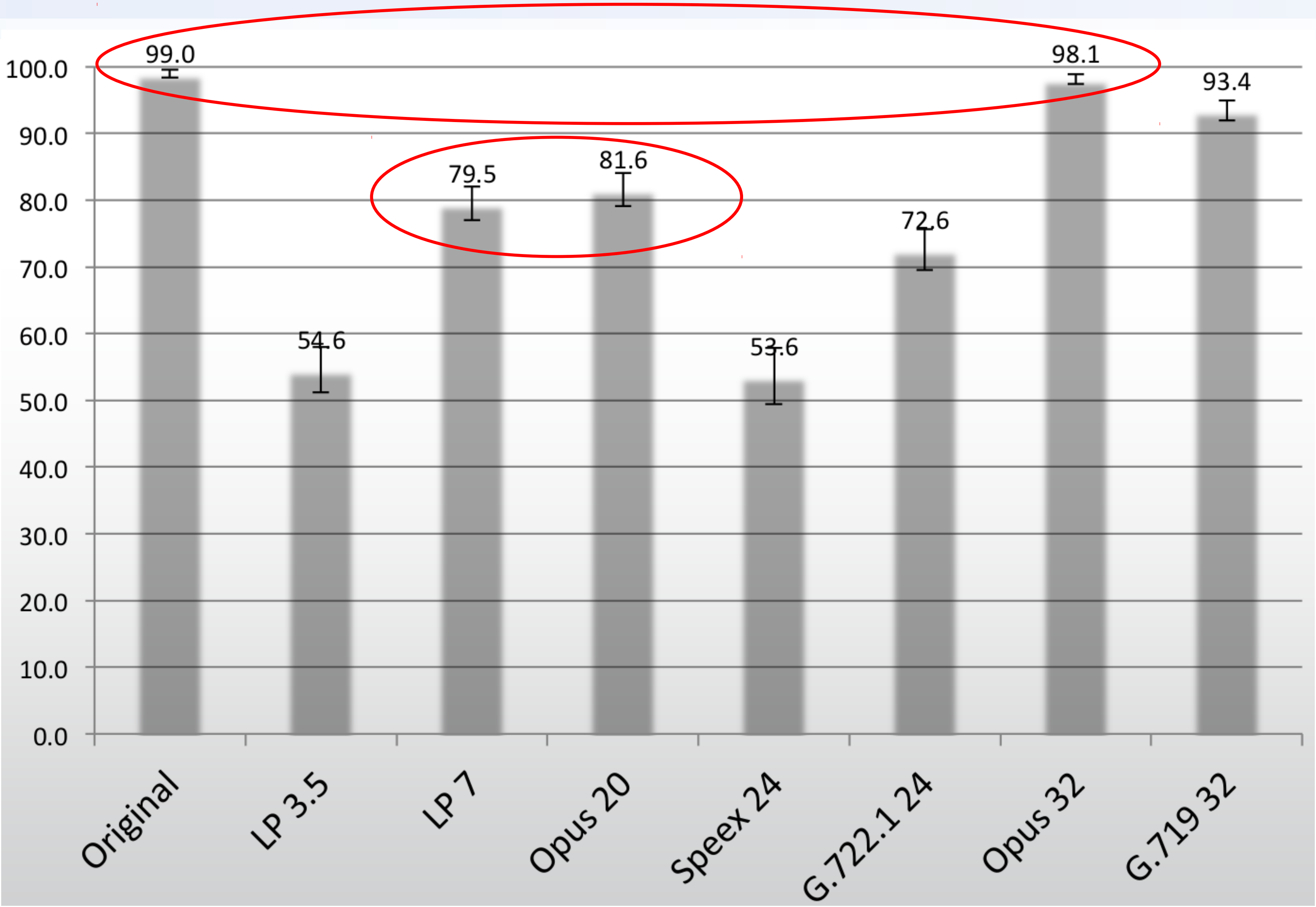
CELT Stereo Coupling

- Code separate energy for each channel
  - Prevents cross-talk
- Converts to mid-side after normalization
  - Mid and side coded separately with their relative energy conserved
  - Prevents stereo unmasking
- Intensity stereo
  - Discards side past a certain frequency

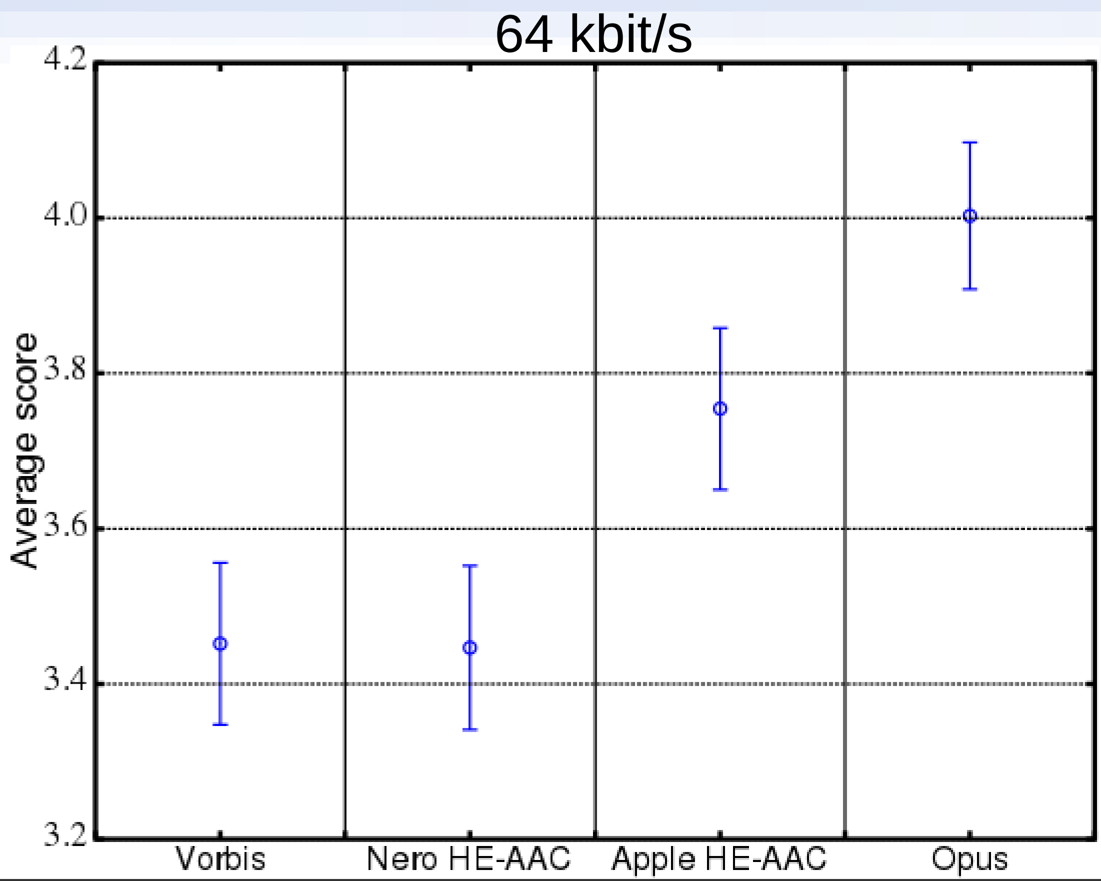
Google Listening Tests (English)

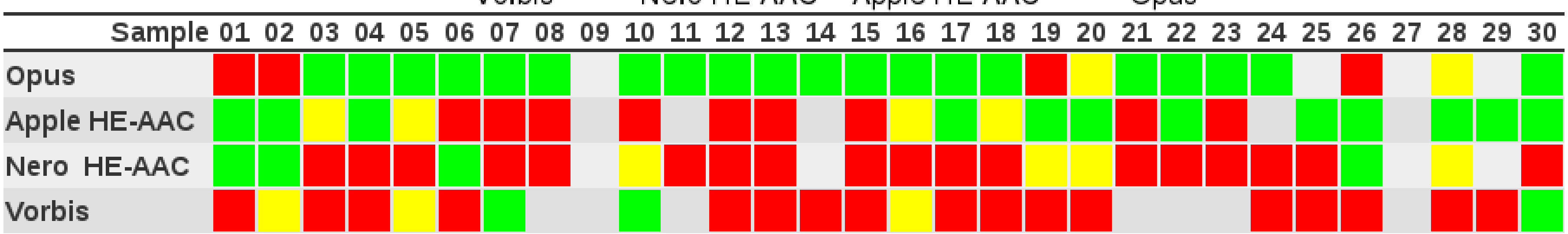


Google Listening Test (Mandarin)

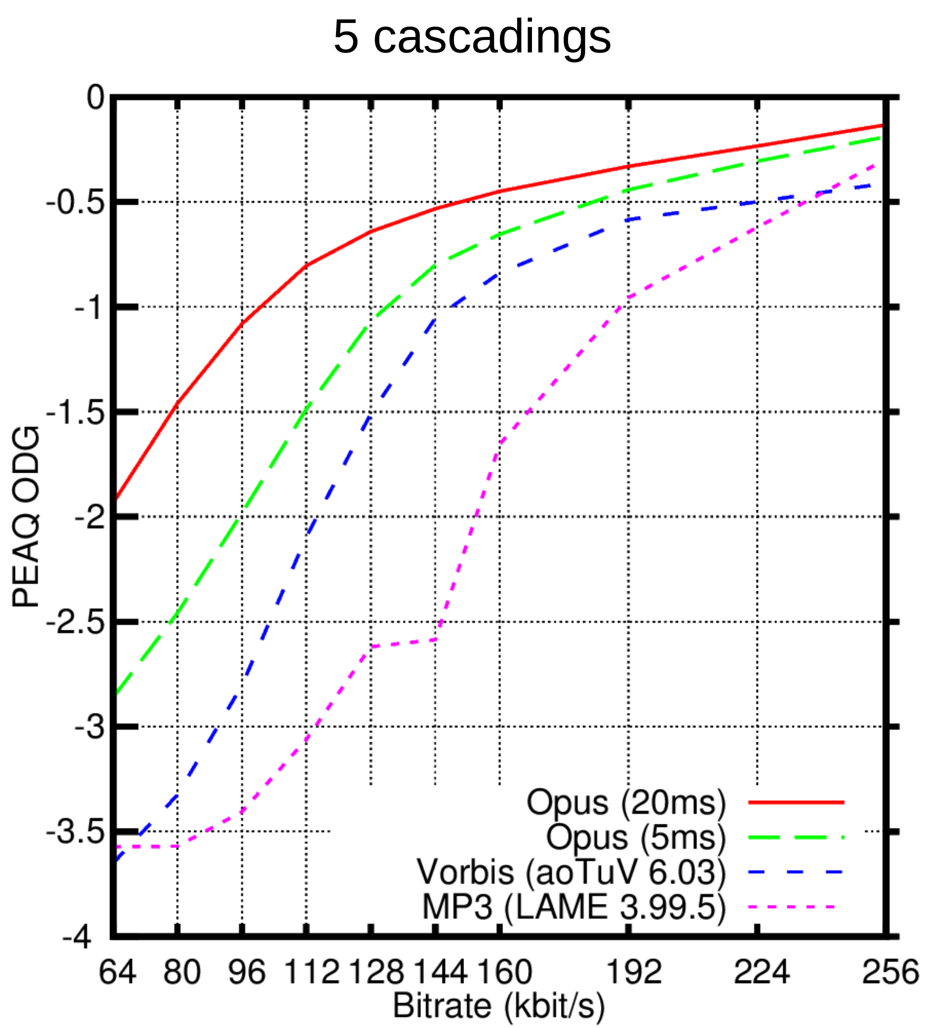


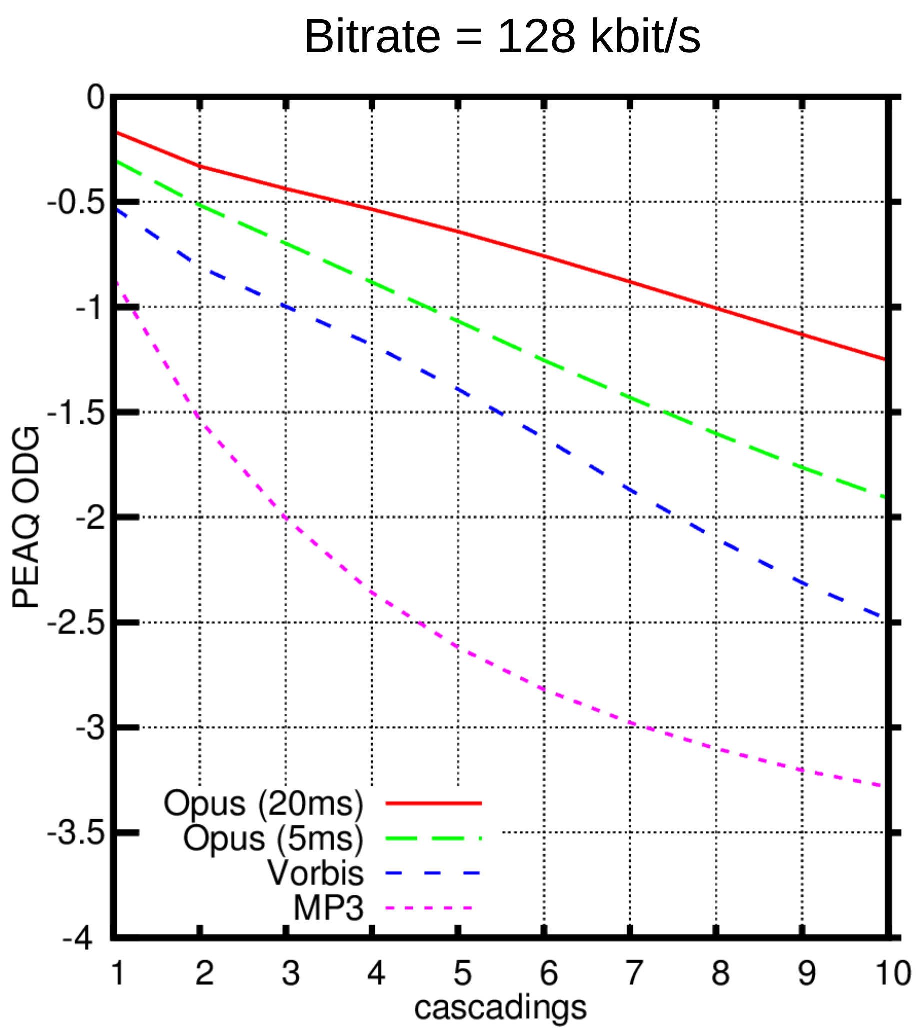
HydrogenAudio Results





Cascading Tests (AES 135)





Adoption

- VoIP and videoconference
  - Jitsi, Meetecho, CounterPath, Mumble, Teamspeak, ...
  - Mandatory-to-implement for WebRTC
    - Already supported in Firefox and Chrome
- Broadcast
  - Tieline, Mayah, Harris Broadcast
- Distribution
  - Magnatune music store
  - StreamGuys CDN

Adoption

- HTTP streaming
  - Firefox 18+ (incl. FFOS), Chrome, Opera
  - Lots of other players:
    - FFMpeg, GStreamer, VLC, Foobar2k, Winamp (with a plugin), Amarok, xmms2, etc.
  - Icecast 2.4-beta1 added Opus support
- Examples:
  - http://dir.xiph.org/by_format/Opus
  - http://www.absoluteradio.co.uk/listen/labs.html

Implementation (libopus)

- Good quality reference implementation
- Opus 1.1 released last December
  - https://people.xiph.org/~xiphmont/demo/opus/demo3.shtml
  - First release with True VBR
  - Automatic speech/music detection
  - Better surround encoding (down to ~64 kb/s)
  - ARM/Neon optimizations

Implementation Flexibility

- Many knobs
  - Application (OPUS_APPLICATION_{VOIP,AUDIO})
  - Complexity (OPUS_SET_COMPLEXITY)
  - Robustness (OPUS_SET_PACKET_LOSS_PERC)
  - Speech/music (OPUS_SET_SIGNAL)
  - Bandwidth (OPUS_SET_BANDWIDTH)
  - Rate control (OPUS_SET_VBR*)
- Defaults are sane, so use only when needed

Standards

- RTP (draft-ietf-payload-opus)
- Ogg (draft-ietf-codec-oggopus)
- WebM (Matroska)
  - Opus paired with VP9 for next RF video format
    - Used by YouTube
  - Spec'd at https://wiki.xiph.org/MatroskaOpus
    - Implementations underway
- Minor RFC 6716 revisions (draft-valin-codec-opus-update)
  - 3 minor bug-fixes to the reference implementation
  - Feedback at codec@ietf.org welcomed!

Opus in RTP

- Very simple: 1 RTP payload == 1 Opus packet
  - From 2.5 ms to 120 ms audio
- Packets decodable with *no OOB signaling*
  - No negotiation failure, always `opus/48000/2`
  - All SDP parameters are informative
  - Mono/stereo, bitrate, audio bandwidth, frame size, mode, etc., signaled in band
  - Receiver decodes all of these transparently
    - Encoder and decoder can run at different rates

Opus in Ogg

- Includes surround support, up to 255 channels
- Similar to RTP mapping
  - Header is informative (except surround)

Resources

- Website: http://opus-codec.org
- Mailing list: opus@xiph.org
- IRC: #opus on irc.freenode.net
- Git repository: git://git.opus-codec.org/opus.git

Next Step: Daala Video Codec

- Creating a free state-of-the-art video codec
- New technology so far:
  - Multisymbol arithmetic coding
  - Lapped transforms
  - Frequency-domain intra prediction
  - Gain-shape quantization (similar to CELT)
  - Overlapping-block motion compensation
- Website: http://xiph.org/daala/

Questions?