



# The Opus Codec

---

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

CCBE  
27 September 2013



# 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?

HOW STANDARDS PROLIFERATE:  
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)



<http://xkcd.com/927/>

<http://imgs.xkcd.com/comics/standards.png>



# Why Should Broadcasters Care?

---

- Ultra-low delay
- Adaptability to varying network conditions
- Best-in-class performance within a wide range of bitrates
- No licensing costs
- No incompatible flavours



# 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, signaled within the bitstream



# Rate Control

---

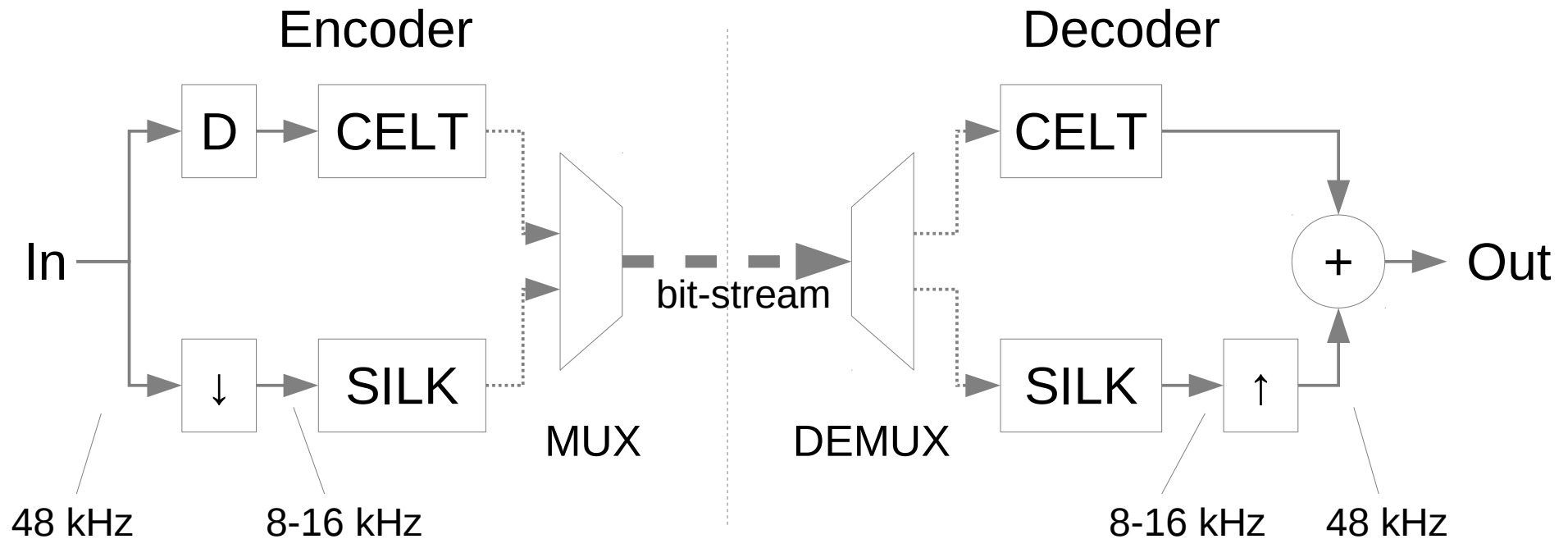
- Opus supports true CBR
    - Every packet has the same number of bytes
    - No bit reservoir => no extra delay
    - Quality not as good as VBR
  - Constrained VBR
    - Total variation within 1 frame of CBR (same as bit reservoir)
    - Bounded delay, better transients, etc.
  - True VBR
    - Open 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 Technology

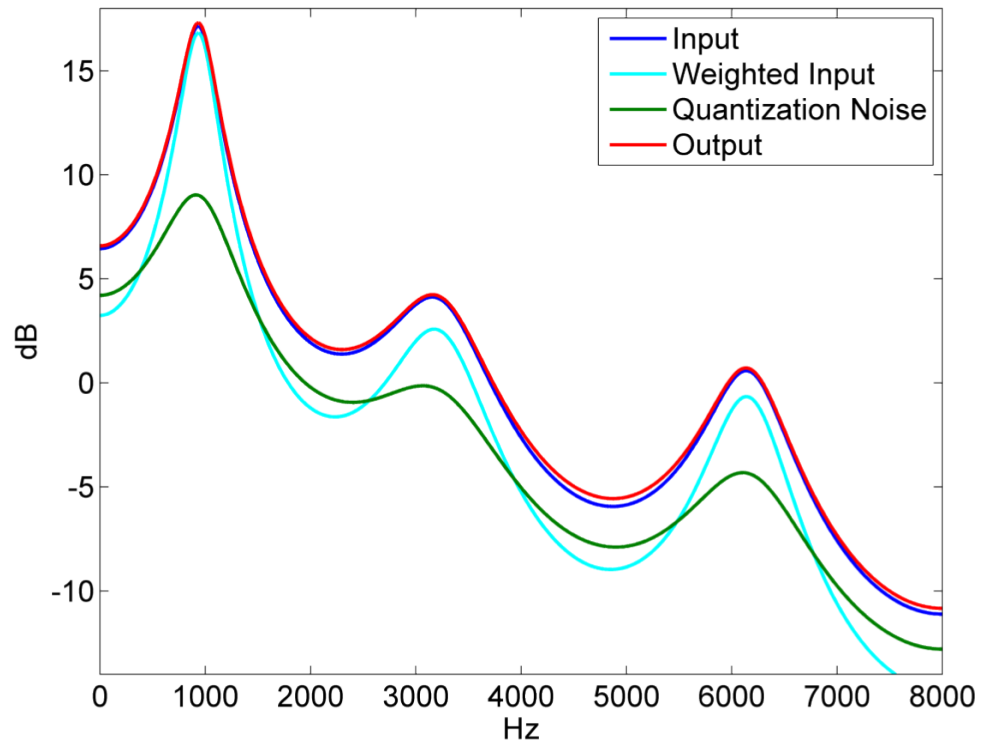
---

- 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



# SILK Technology

- Based on Noise Feedback Coding rather than Analysis-by-Synthesis
- Analysis/synthesis mismatch to de-emphasize spectral valleys
  - Replaces post-filters
- Variable-rate coding





# CELT Technology

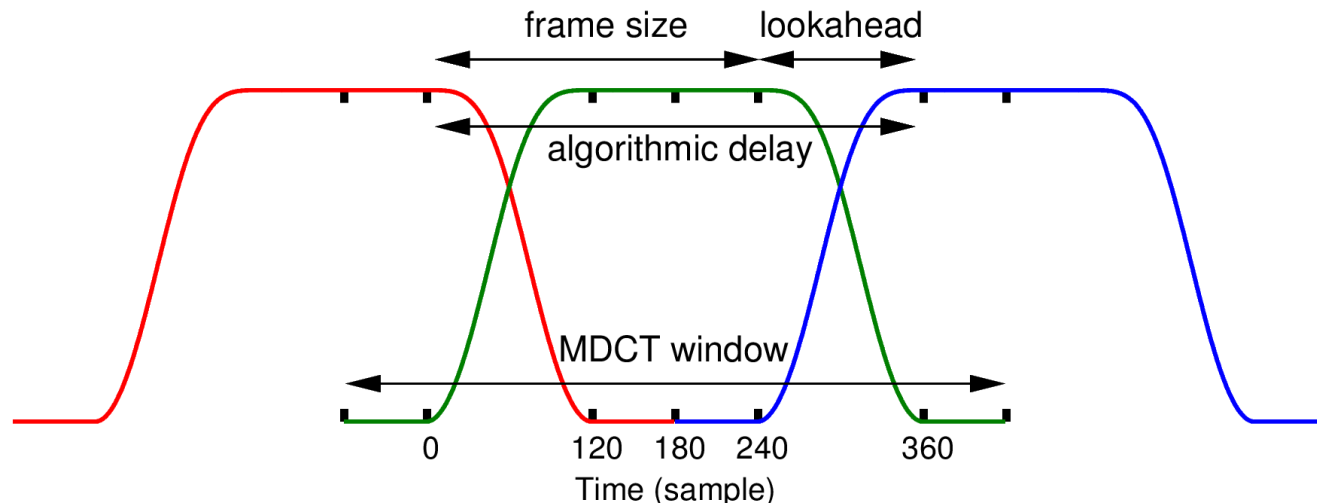
---

- “Constrained-Energy Lapped Transform”
  - Psychoacoustics built into the format
  - Harder to write a bad encoder
- Works on speech and music
- Most efficient on fullband audio (48 kHz)
- Less efficient on low bitrate speech



# CELT Technology

- MDCT with low-overlap window



- Code band energy separately from spectrum “details”
  - Preserves the energy in each critical band
- Implicit masking curve defined by the format
  - No need to code scalefactors



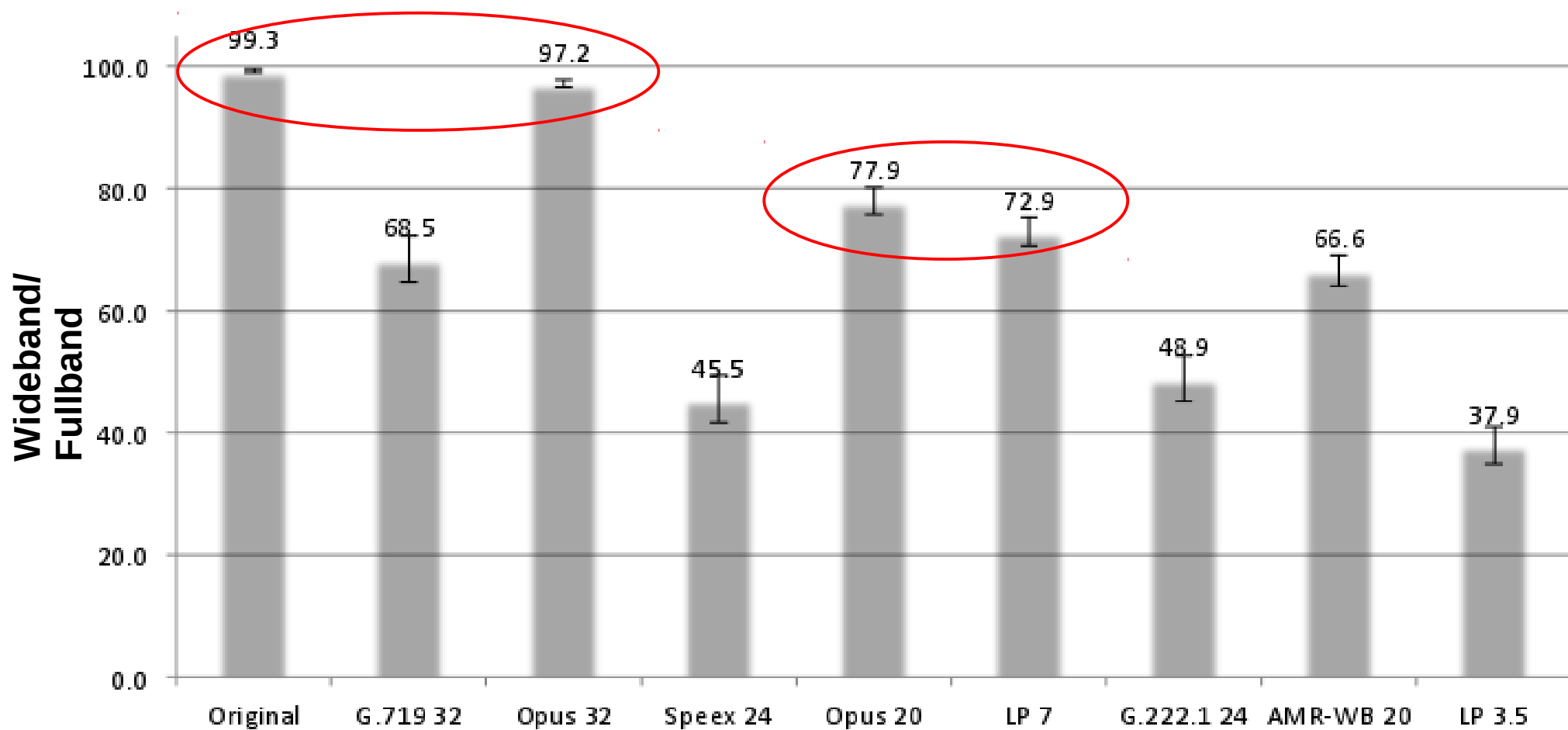
# CELT Stereo Coupling

---

- 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

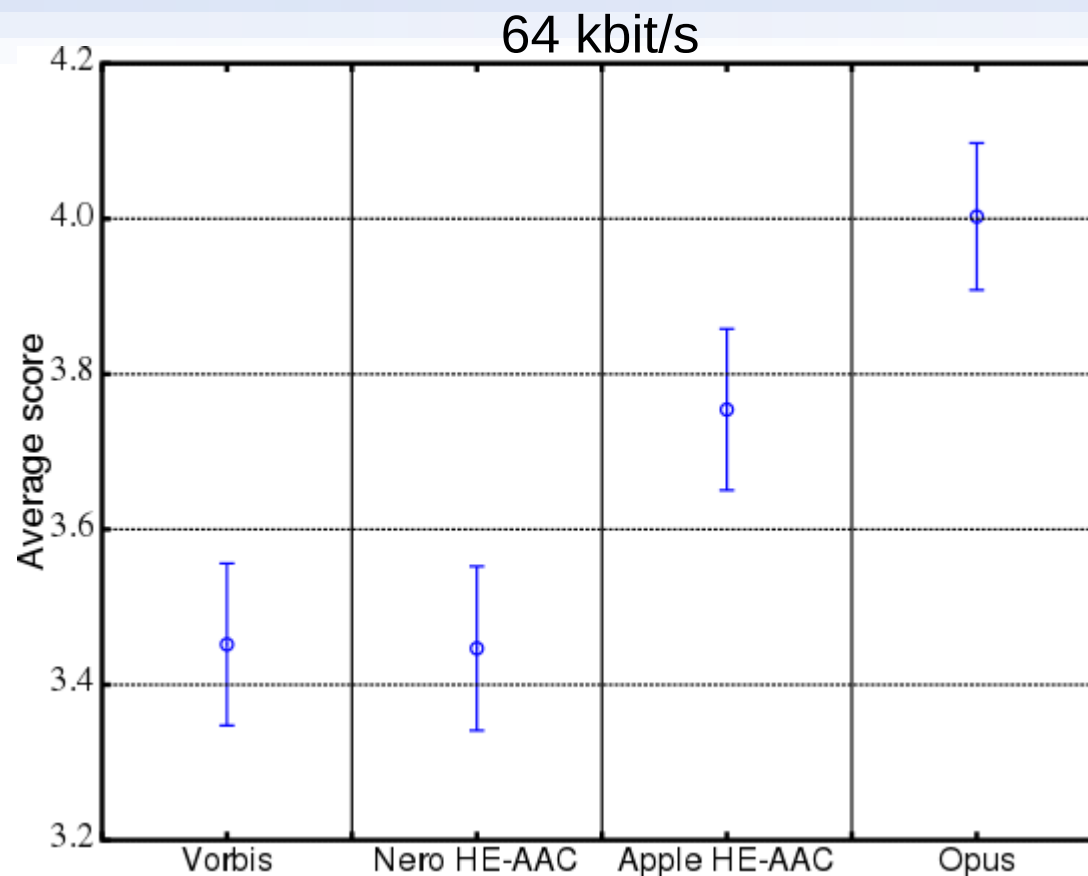


# Google Listening Tests





# HydrogenAudio Results



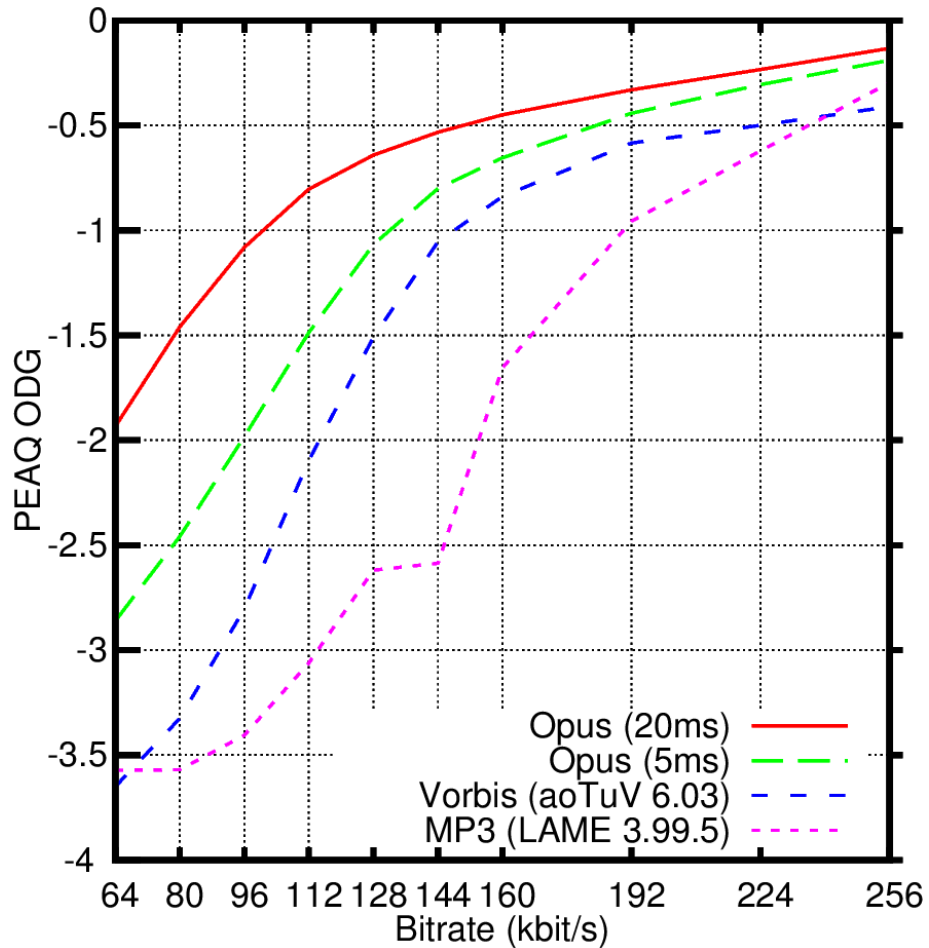
	Sample 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Opus	Red	Red	Green	Green	Green	Green	Green	Green	Grey	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Yellow	Green	Green	Green	Green	Grey	Red	Grey	Yellow	Grey	Green
Apple HE-AAC	Green	Green	Yellow	Green	Yellow	Red	Red	Red	Grey	Red	Grey	Red	Red	Grey	Red	Yellow	Green	Yellow	Green	Green	Red	Green	Red	Grey	Green	Green	Grey	Green	Green	Green	Green
Nero HE-AAC	Green	Green	Red	Red	Red	Green	Red	Red	Grey	Yellow	Red	Red	Red	Grey	Red	Red	Red	Red	Yellow	Yellow	Red	Red	Red	Red	Red	Green	Grey	Yellow	Grey	Red	
Vorbis	Red	Yellow	Red	Red	Yellow	Red	Green	Grey	Grey	Green	Grey	Red	Red	Red	Red	Red	Yellow	Red	Red	Red	Red	Grey	Grey	Red	Red	Red	Grey	Red	Red	Green	



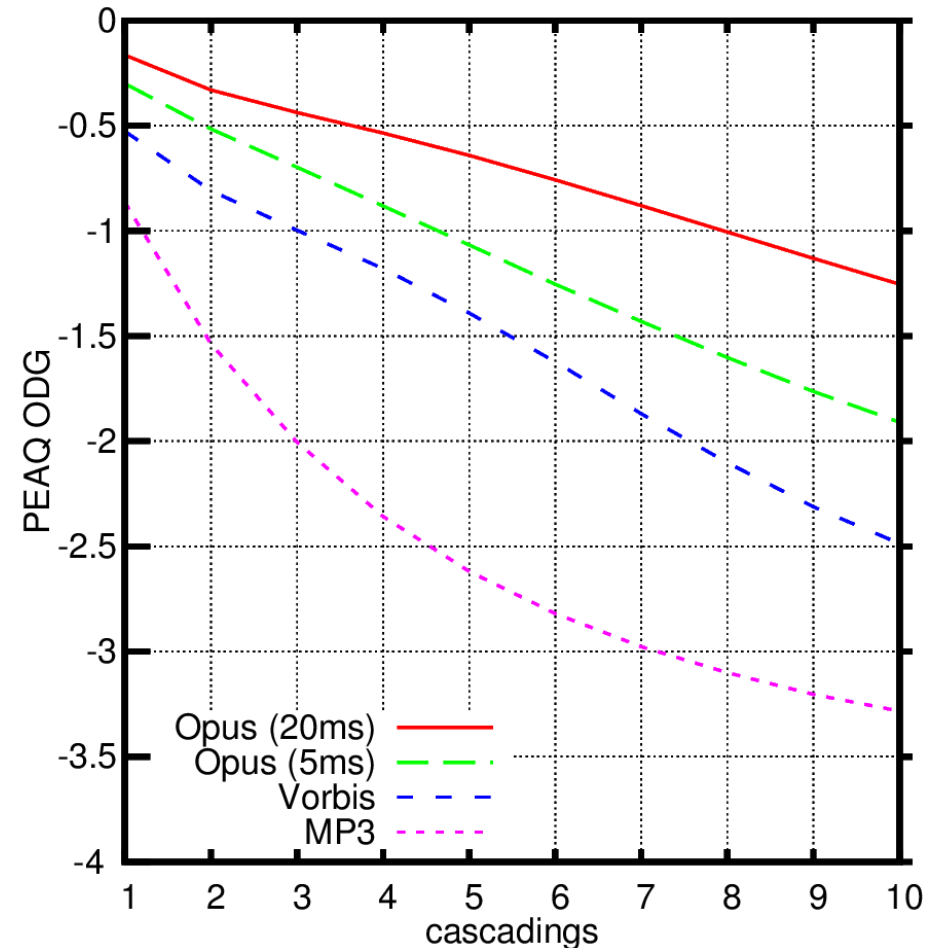


# Cascading Tests (AES 135)

5 cascading



Bitrate = 128 kbit/s





# Adoption

---

- Broadcast
  - Tieline, Mayah, Harris Broadcast
  - CBS, ABC, NBC, NPR, Fox, Cumulus, ...
- Distribution
  - Magnatune music store
  - StreamGuys CDN
- VoIP and videoconference
  - Jitsi, Meetecho, CounterPath, Mumble, Teamspeak, ...
  - Mandatory-to-implement for WebRTC



# 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://dir.xiph.org/by_format/Opus)
  - <http://www.absoluteradio.co.uk/listen/labs.html>



# Roadmap



# libopus 1.1

---

- Beta released in July, full release “soon”
    - <https://people.xiph.org/~xiphmont/demo/opus/demo3.shtml>
  - First release with True VBR
    - Tonality estimation
    - Better dynamic allocation
      - Improves on the built-in psychoacoustics
    - Temporal VBR (discovered by accident!)
  - Automatic speech/music detection
    - Optional delayed decision (better high-latency performance)
-



# libopus 1.1 (cotd.)

---

- Better surround encoding
  - Better API (knows which channel is which)
  - Better LFE encoding
  - Inter-channel masking
- Major ARM performance gains:
  - 40% decoder CPU reduction
  - 27% encoder CPU reduction (33% with Neon)



# Standards

---

- RTP (draft-ietf-payload-opus)
    - Hopefully WGLC soon
  - Ogg (draft-ietf-codec-oggopus)
    - Maybe WGLC soon?
  - 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](mailto: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](mailto:opus@xiph.org)
- IRC: #opus on [irc.freenode.net](http://irc.freenode.net)
- Git repository: [git://git.opus-codec.org/opus.git](https://git.opus-codec.org/opus.git)

## Questions?