



Freeswitch & Opus Audio Codec

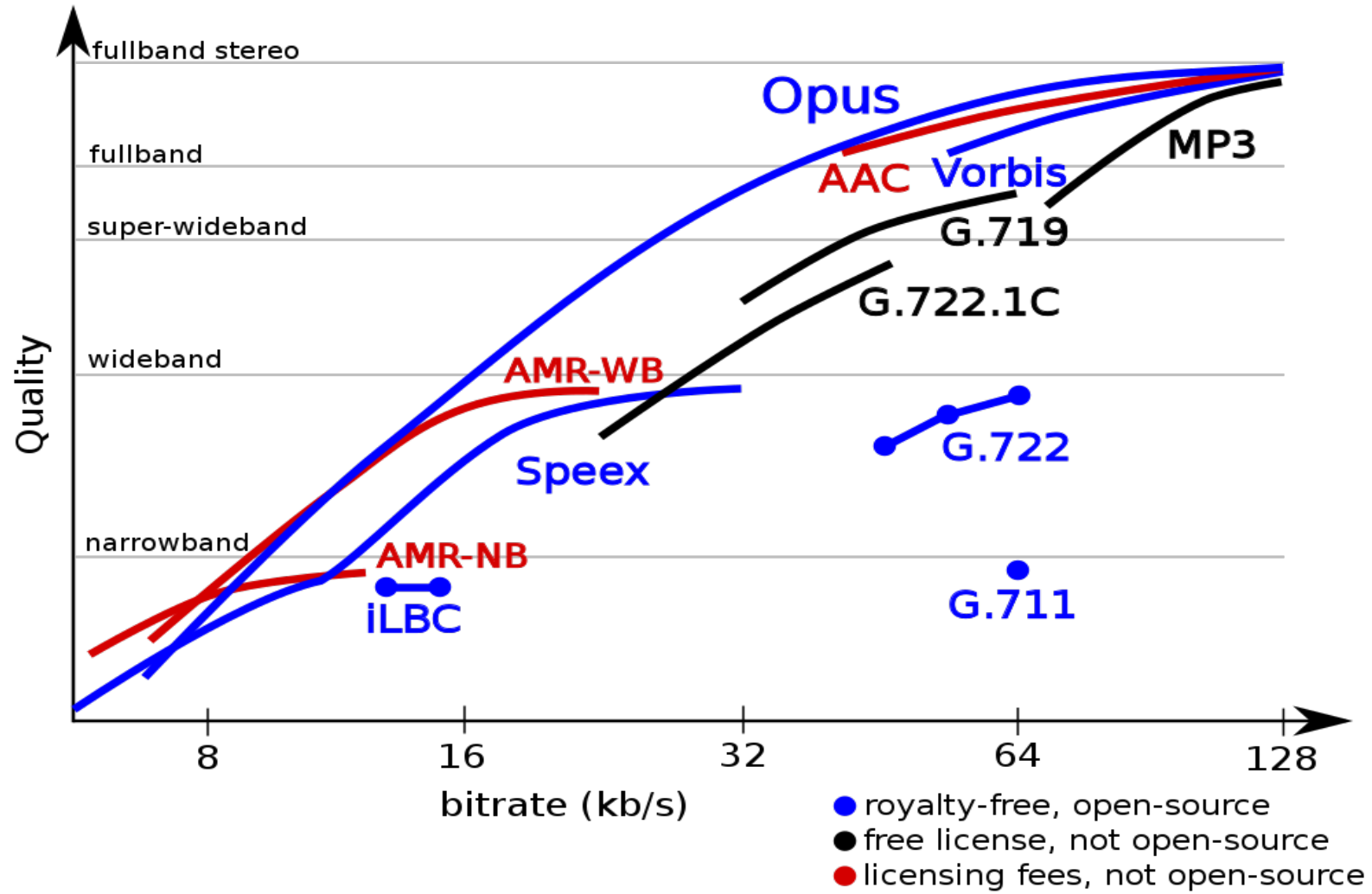
CLUECON 2017

- Intro (Opus)
- SDP Offer/Answer , fmp
- Forward Error Correction (FEC)
 - Encoder
 - Decoder
- Bitrate adjustment

“Opus is unmatched for interactive speech and music transmission over the Internet, but is also intended for storage and streaming applications. It is standardized by the Internet Engineering Task Force (IETF) as RFC 6716 which incorporated technology from Skype’s SILK codec and Xiph.Org’s CELT codec.”

<https://opus-codec.org/>

Opus



- Low algorithmic delay
- Has the aim of maximizing the quality/bitrate tradeoff
- Resilient to packet loss
- Storage & playback format
- Strong API

- Opus support is mandatory for WebRTC implementations
- Some browsers default to Opus stereo, some to mono, some default to FULLBAND, some to WIDEBAND , some use DTX (different ways of using the WebRTC lib) . Behaviour may change from version to version

- Personal claim: mod_opus.c RFC 7587 compliant (RTP Payload Format for the Opus Speech and Audio Codec)
- Tied with the jitter buffer
- Opus @ 48 khz , Opus @ 8 khz , Opus @ 16 khz (new)
- Codec control
- 10 -120 ms ptime @ 8 khz , 10-60 ms ptime @ 16 khz , 10-40 ms ptime @ 48 khz

- Fmtp params (Local Encoder) :
sprop-maxcapture, sprop-stereo
- Fmtp params (Local Decoder):
ptime, maxptime, maxaveragebitrate, maxplaybackrate, stereo, cbr,
useinbandfec, usedtx

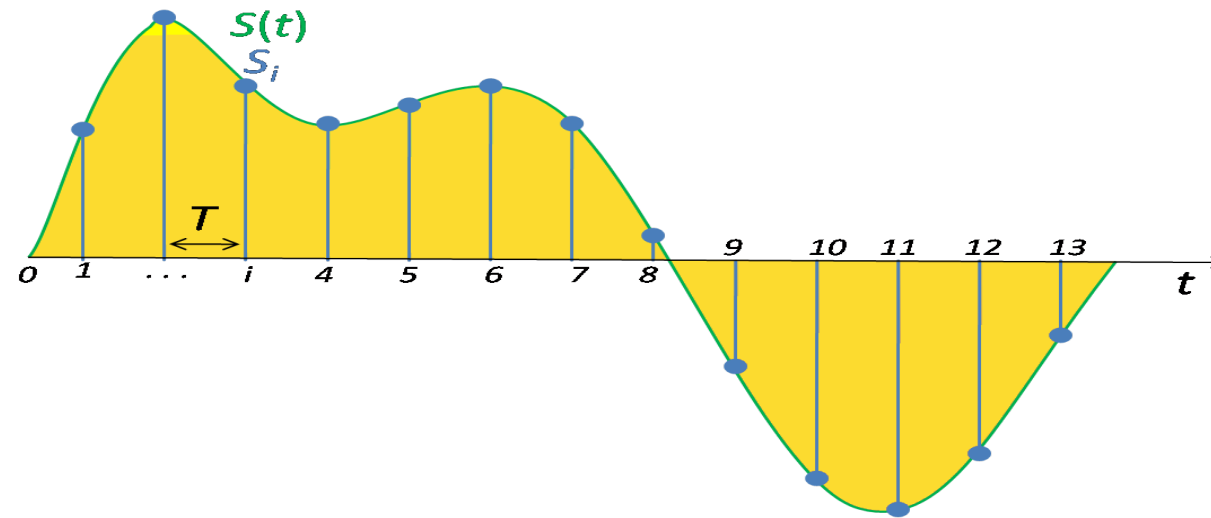
```
m=audio 54312 RTP/AVP 101  
a=rtpmap:101 opus/48000/2  
a=fmtp:101 useinbandfec=1; maxplaybackrate=8000; maxaveragebitrate=14000  
a=ptime:20
```


FMTP sent by FS is affected by these SIP profile settings:

```
<param name="inbound-codec-negotiation" value="greedy" />
```

```
<param name="inbound-late-negotiation" value="false" />
```

Resampling



Sampling is the reduction of a continuous signal to a discrete signal. The sampling frequency or **sampling rate**, f_s , is the average number of samples obtained in one second (samples per second)

$$f_s > 2B$$

- You don't want to do resampling for transcoding unless you have no choice
- Use `opus@8000h` for PCMA/PCMU transcoding into Opus
- `opus@8000h` Suitable for mobile apps



Freeswitch: FEC on Opus encoder/decoder

General Definition : Forward Error Correction (FEC) or channel coding is a technique used for controlling errors in data transmission over unreliable or noisy communication channels. The central idea is the sender encodes the message in a redundant way by using an error-correcting code (ECC)



FEC in audio codecs : The in-band FEC feature of Opus helps reduce the harm of packet loss by encoding some information about the prior packet.

Freeswitch: FEC on Opus encoder/decoder

- keep-fec-enabled (encoder)

FEC f(packet_loss, bitrate)

```
<param name="keep-fec-enabled" value="1" /> (opus.conf.xml)
<param name="packet-loss-percent" value="15" />
```

- Jitter Buffer (decoder)

```
<param name="advertise-useinbandfec" value="1" />
<param name="use-jb-lookahead" value="1" />
<action application="set" data="jitterbuffer_msec=2p:25p:" />
<action application="set" data="rtp_jitter_buffer_plc=true" />
<action application="set" data="rtp_jitter_buffer_during_bridge=true" />
<action application="set" data="suppress_cng=true" />
```

Feedback loop: reading RTCP packet loss, calculating an average, calling a function (`codec_control`) to tell the codec that there is a certain percentage of packet loss for the outgoing media stream.

Even when FEC is not used, telling the encoder about the **expected level of loss** will help it make more intelligent decisions. By default the implementation assumes there is no loss.

- Implemented logic for codec_control in mod_opus.c , added a function to keep FEC enabled by slightly increasing the bitrate
- #define SWITCH_OPUS_MIN_FEC_BITRATE 12400 (no FEC info below this bitrate)
- f(packet_loss, bitrate)

```
<param name="keep-fec-enabled" value="1"/> (opus.conf.xml)
```

Initial packet loss:

```
<param name="packet-loss-percent" value="15"/>
```


- We will recalculate a new bitrate based on the packet loss percentage taken from RTCP arriving to us from the user app if the packet loss is higher than 10%
- Trick: by forcing FEC on the local encoder the remote decoder will decode more FEC packets if there is packet loss, hence increase call quality

• “In order to make use of in-band FEC the decoder must delay its output by at least one frame so that it can call the decoder with the `decode_fec` argument on the next frame in order to reconstruct the missed frame. This works best if it's integrated with a jitter buffer.”

- The jitter buffer intentionally delays the arriving packets
- FEC will be played if available (voiced frames)

Opus PLC will be played instead of FS PLC (`libspandsp`)

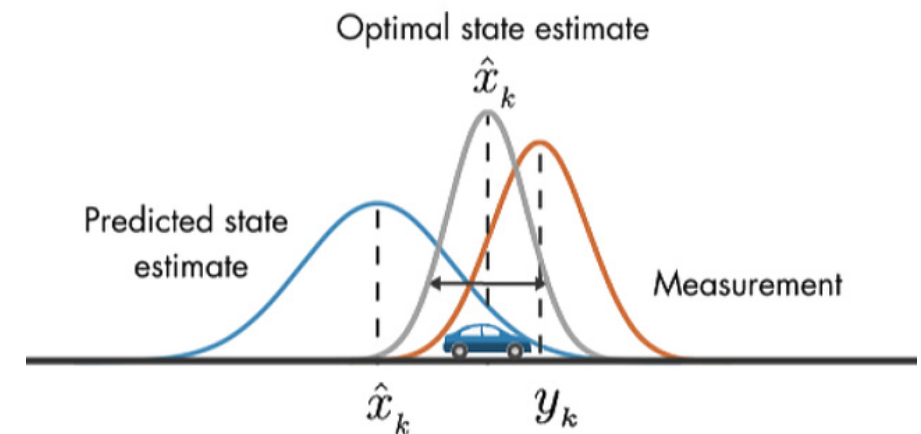
- ✓ Disable default Freeswitch PLC (done with libspandsp)
- ✓ Enable Jitter Buffer so it would pass SFF_PLC flag to the decoder
- ✓ If we miss the n-th frame , then pick n+1 frame from the Jitter Buffer if it's present :
 - if we have the frame then try to do FEC on it
 - if we don't have the frame we do PLC and return



Freeswitch & Opus: Congestion control / Bitrate adjustment.

“an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone”

```
<param name="adjust-bitrate" value="1" />
```



- Based on RTCP Feedback
- Senses sudden changes in average of Packet loss and RTT and adjusts encoder bitrate via codec control
- Varies the bitrate in 0.4 kb steps (1200 bitrates) . Min: 6kb Max: 512 kb
- Hard to model
- Experimental code

- “adjust-bitrate” setting enables a form of congestion control and it's based on estimators and detectors
- we pass the “future” packet loss percentage to the encoder

FS CLI>opus_debug on



```
[DEBUG] mod_opus.c:453 decode: opus_frames [1] samples [320] audio
  bandwidth [NARROWBAND] bytes [87] FEC[yes] channels[1]
[DEBUG] mod_opus.c:764 Missing SEQ 482 Checking JB
[DEBUG] mod_opus.c:769 Lookahead frame found: 1364160:483
[DEBUG] mod_opus.c:781 FEC info available in packet with SEQ: 483
  LEN: 91
[DEBUG] mod_opus.c:798 MISSING FRAME: Look-ahead FEC
[DEBUG] mod_opus.c:453 FEC correction: opus_frames [1] samples [320]
  audio bandwidth [NARROWBAND] bytes [91] FEC[yes] channels[1]
```


Logged at the end of a call. Eg:



- *Opus decoder stats:* Frames[7038] PLC[2247] FEC[29]
- *Opus encoder stats:* Frames[1219] Bytes encoded[35490]
Encoded length ms[24380] Average encoded bitrate
bps[11830] FEC frames [755]

MULTIDSLA – test equipment



- opus@16000h (Opus at 16000 Hz, together with AMR-WB helps connect the WebRTC World with the IMS World) - no resampling when transcoding
- mod_opusfile (module to encode and decode OGG/OPUS files) - upstream PR

- Codec settings on dialplan (eg: different SIP trunks might need different OPUS fntp settings)
- RTCP – XR ?
- Ptime change during the call ? (hard to achieve, much easier with client side software)
- Kalman filter: aggregate information (packet loss % , RTT, jitter) – eg: comparable with sensor fusion – come up with a single value which would describe how congested is the network (how bad is it)

Opus FAQ - <https://wiki.xiph.org/OpusFAQ>

FreeSWITCH And The Opus Audio Codec Confluence Page (written by me and Giacomo Vacca)

<https://freeswitch.org/confluence/display/FREESWITCH/FreeSWITCH+And+The+Opus+Audio+Codec>

Definition of the Opus Audio Codec - <https://tools.ietf.org/html/rfc6716>

Opus Interactive Audio Codec - <https://opus-codec.org/>



THANK YOU!

QUESTIONS?