Deep Redundancy for the Opus Codec
draft-valin-opus-dred-01

Presenter: Jean-Marc Valin
jmvalin@amazon.com

IETF 117
Deep REDundancy (DRED)

- Goal: make Opus robust to long bursts of packet loss
- Proposal: code large amounts of redundant audio
  - Use DNN to maximize compression
  - Can code up to 1 second per 20-ms packet (50x)
Forward-Backward Compression

- Don’t re-encode from scratch
- LIFO decoder
- 40-ms “chunks”
  - Updated every 20 ms

\[ \text{forward encoding} \]
\[ \text{odd} \]
\[ \text{even} \]
\[ \text{odd} = 20 \text{ ms frame} \]
\[ s_1 s_2 s_3 s_4 s_5 s_6 s_7 s_8 s_9 s_{10} s_{11} s_{12} s_{13} s_{14} s_{15} s_{16} \]
\[ \text{backward decoding} \]
\[ \text{odd} \]
\[ \text{even} \]
\[ \text{odd} \]
\[ s \]
\[ = \text{initial state (IS)} \]
Results

- Mean Opinion Score (MOS) on Deep PLC Challenge data
- Examples:
  - Opus w/o loss
  - Opus+LBRR
  - Opus+LBRR+DRED
Proposed Format

- Use extension code 32 (temporarily 127)
- Offset: position of redundancy in packet (5 bits)
  - 2.5 ms resolution from -40 ms to +37.5 ms
- Quantizer: 4 bits
- Quantizer slope: 3 bits
- Initial state (~96 bits)
- Quantized latent vectors
- Decode until fewer than 8 bits remain
Normative Aspects

• Balancing act
  - Ensure all implementations are inter-operable
  - Leave as much flexibility as possible

• Normative spec for bits-to-feature decoder
  - All decoder weights are frozen
  - Definition of the acoustic features (how?)

• Encoder and vocoder are left unspecified
Running Code

- DRED-enabled version of libopus (experimental) at https://gitlab.xiph.org/xiph/opus in opus-ng branch
  - Works best for wideband SILK, but other modes supported
  - Takes 5-10% of laptop CPU core (improvements underway)
- Blog post: https://www.amazon.science/blog/neural-encoding-enables-more-efficient-recovery-of-lost-audio-packets
Open Questions

• What should be the maximum duration allowed?
• What are the lowest and highest useful bitrates?