Deep Redundancy for the Opus Codec
draft-valin-opus-dred-05

Presenter: Jean-Marc Valin
jmvalin@jmvalin.ca

IETF 119
DRED Recap

- Code large amounts of redundant audio in Opus packet
  - Use DNN to maximize compression
  - Can code 1 second per 20-ms packet (50x)
Changes Since IETF 118

- Added extended offset
  - Optional extra byte signals up to 20 second offset
  - Used to trim silence (more efficient, better DTX)
  - Potential use in SFU just after switch

- Added Qmax field
  - Makes it possible to cap the quantizer
  - Variable size
  - Costs one bit when unused
### Format

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Additional formatting not visible in the image.
Normative Aspects

- Normative spec for bits-to-feature decoder
  - All decoder weights are frozen
    - Publish as simple binary format
  - Definition of the acoustic features
    - How do we specify neural pitch estimator?

- Encoder is left unspecified

- Minimal constraints on vocoder
Implementation Update

- Changes landed in new Opus main branch
- Released latest implementation in Opus 1.5
  - https://www.opus-codec.org/demo/opus-1.5/
  - DRED disabled by default
- WebRTC patch set with DRED support
  - https://github.com/xiph/webrtc-opus-ng/
Results

![Graph showing the relationship between PLCMOS score and percentage loss with different methods: None, LBRR, DRED, and LBRR+DRED. The graph indicates that LBRR+DRED performs the best, followed by DRED, LBRR, and None.]
Open Questions

- Should there be a maximum duration allowed?
  - Technically we could do up to ~10 minutes
  - Proposal: no hard limit, since receiver can ignore the rest

- What are the lowest and highest useful bitrates?
  - Currently support 10 to 100 kb/s for 1 second redundancy
  - Equivalent to 200 b/s to 2 kb/s effective bitrate