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

Presenter: Jean-Marc Valin
jmvalin@jmvalin.ca

IETF 120

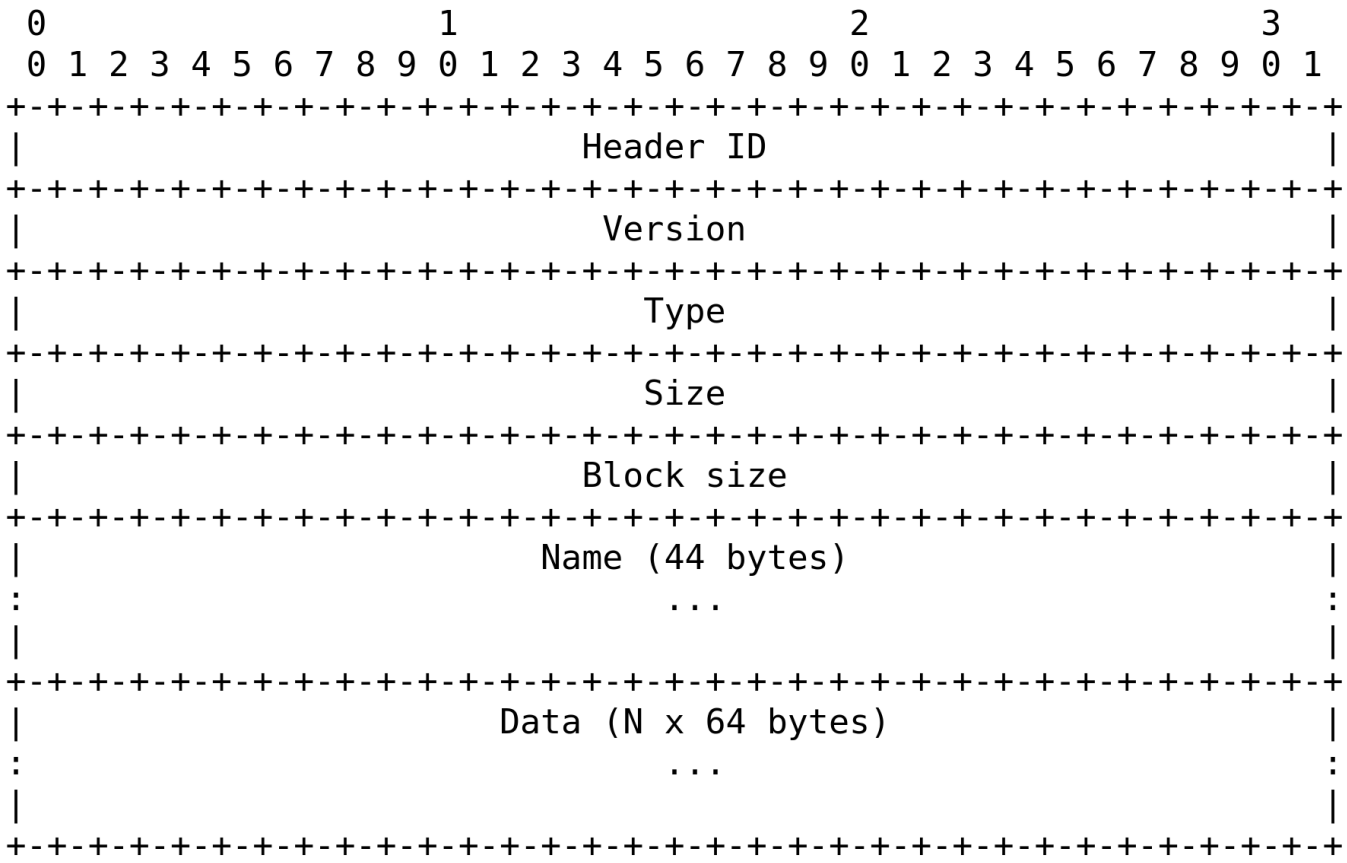
Changes Since IETF 119

- Amount of redundancy unbounded
- SHOULD have one redundancy per packet and SHOULD be first frame
- MUST NOT have more than one extension per frame
- SHOULD NOT encode leading/trailing silence
- SFU SHOULD forward DRED from last active source
- Mixers SHOULD forward or re-encode

DRED Specification (strawman)

- Normative feature decoder, encoder provided as reference
  - Define the computation
  - Compliance based on testvectors (like Opus)?
- Distribution parameters (p0, r) in the draft as tables (8 pages)
- Decoder architecture defined as text/math/pseudocode
- Decoder weights defined in binary file
  - Same as in reference code to minimize errors
  - Floating point or fixed-point?
  - Where do we publish the file?

Binary Weights Format



Vocoder

- No normative vocoder algorithm
  - The vocoder dominates complexity
  - It’s possible to switch vocoder
  - Vocoders are still improving rapidly
- Options:
  1) Define minimal objective metrics on vocoder synthesis
  2) “Feature analysis on vocoder output must match input”
  3) Leave unspecified (rely on feature definition)

Open Questions

- Handling the decoder block-sparse weight matrices
  1) Specify using dense format (i.e. include zeros)
  2) Specify using sparse format (skip zeros)
  3) Don’t use sparse matrices
- Do we say anything normative about how features are computed?