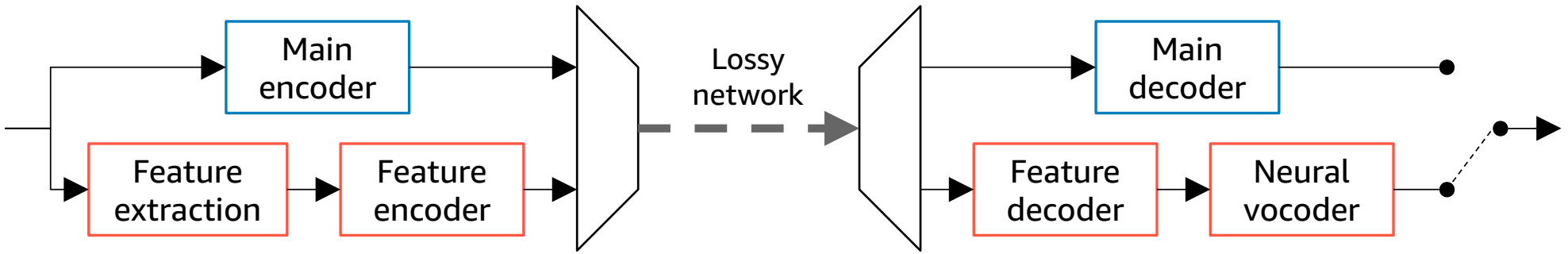
Deep Redundancy for the Opus Codec
`draft-valin-opus-dred-02`

Presenter: Jean-Marc Valin
jmvalin@amazon.com

IETF 118

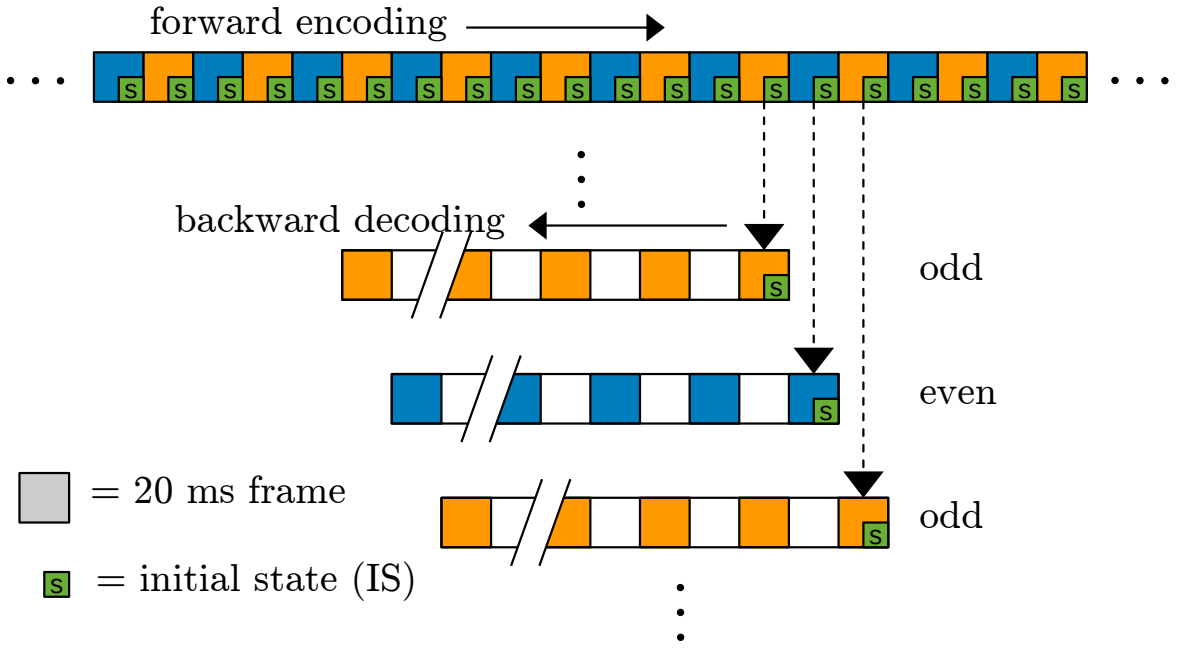
Deep REDundancy (DRED)

- Goal: make Opus robust to long bursts of packet loss
- Proposal: code large amounts of redundant audio
  - Use DNN to maximize compression
  - Can code 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



Proposed Format

- Use extension code 32 (temporarily 126)
- Offset: position of redundancy in packet (5 bits)
  - 2.5 ms resolution from -40 ms to +37.5 ms
- Quantizer selection (bitrate): 4 bits
- Quantizer slope: 3 bits
- Initial state now entropy-coded (previously fixed rate)
- Latent vectors entropy-coded
- 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 (how do we publish them?)
  - Definition of the acoustic features
    - How do we specify neural pitch estimator?
- Encoder is left unspecified
- Minimal constraints on vocoder

Implementation Update

- Improved quality from new vocoder
- Complexity reduced from 10% to 3% CPU for high loss
  - No cost under no loss conditions
- Size of weights down from 17 MB to 4 MB
- DRED-enabled version of libopus at https://gitlab.xiph.org/xiph/opus in opus-ng branch

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