Opus Packet Format (not to scale)

- TOC Byte (code 3)
- Padding Bit
- Opus Frame Length(s)
- Padding (extensions go here)

- Frame Count Byte
- Padding Length (variable)
- Compressed Opus Frames
Opus Extension Format

ID #1

Length for ID=32...127, L=1 (variable)

L Flag

Extension Payload

ID #2...

Extensions for Frame 0

Extensions for Frame 1

Separator (ID=1)

Frame Increment (optional)

Separator (ID=1)

Etc.
Draft Status

- Published as WG draft
  - 00 with no changes from individual draft
  - 01 with updates
Updates Since San Francisco

- Reserved ID 127 for more extensions
  - Length coded same as IDs 32...126 so it can be skipped
  - Contents subject of future draft
- Quoted text from RFC 5576 Section 6.3: “...media-level format parameters MUST NOT be carried over blindly.”
- Clarified support for extension IDs 0 and 1 does not need to be explicitly signaled via a=fmtp
- Asked extensions to specify whether or not they can appear multiple times for the same frame in a packet
- Various other minor edits
Two Future Extension Mechanisms?

- **ID=0, L=0:**
  - Encoder MUST set rest of the padding to zero
  - Decoder MUST ignore rest of the padding
  - Same rules that let us add these extensions
    - Can be extended recursively
    - Would have to repeat separator bytes to assign extensions to individual frames

- **ID=127**
  - Has a defined length
    - Can be skipped and mixed with other extensions
  - Can re-use this draft's separator structure to assign extensions to individual frames
Changes Not Made

- Did not split out IANA registration for L=0 and L=1 modes for IDs 2...31
  - Need to decide if we want to, because of a=fmtp signaling
- Did not switch to QUIC varint for extension IDs
  - Reduces available 1-byte extension IDs from 118 to 30
- Did not reserve “unsafe” extension IDs (e.g., throw away all extensions unless you understand these)
  - No clear use cases
Questions?

- Comments?
- Other feedback?