FEC for WebRTC

uberti-rtcweb-fec-00

FEC Can Mean Multiple Things

- Separate stream of recovery packets (e.g. RFC 5109)
- Redundant encoding (RFC2198)
- Codec-specific mechanisms (e.g. Opus)
- Different approaches make sense for different media types

Audio

- Opus in-band FEC recommended
  - a=fmtp:XX useinbandfec=1
- For other codecs, RED with duplicated payloads
  - Each packet contains current frame and a prev frame
  - a=rtpmap:XX red/8000
- Separate-stream FEC discouraged
  - Excessive overhead in most cases

Video

- Send FEC on a separate SSRC from primary stream (like RTX)
- Use 'flexfec' packetization from Varun
  - RFC5109 doesn't work with muxing
  - flexfec works with all FECFRAME schemes; Parity is MUST implement
- Use unified plan semantics for SDP

Example SDP

```
m=audio 20000 UDP/TLS/RTP/SAVPF 96
a=mid:a1
a=rtpmap:96 opus/48000/2
a=fmtp:96 useinbandfec=1
a=ssrc:1111

m=video 30000 UDP/TLS/RTP/SAVPF 100 110
c=IN IP4 233.252.0.1
a=mid:v1
a=rtpmap:100 VP8/90000
a=rtpmap:110 interleaved-parityfec/90000
a=fmtp:110 L:5; D:10; ToP:0; repair-window:200000
a=ssrc:1234
a=ssrc:2345
a=ssrc-group:FEC-FR 1234 2345
```

Out of Scope For Now

- Protection of multiple streams with a single FEC stream
  - Violates Unified Plan