

webrtcbin progress update

Who am I?

WEBRTC BIN

- Current implementation in -bad plugins, API subject to change
- `libnice` for Interactive Connectivity Establishment (ICE)
- `srtp` plugins for secure RTP profile

NEW FEATURES

- BUNDLE
- Forward Error Correction (FEC)
- Data channels

BUNDLE

- <https://tools.ietf.org/html/draft-ietf-mmusic-sdp-bundle-negotiation-53>
- Multiplexing media on a single transport
- Negotiation based on policy

SDP

```
a=group:BUNDLE audio video  
[...]  
m=audio [...]  
m=video [...]
```

IMPLEMENTATION

- No change required in `rtpbin \o/`
- Usage of Pexip's `rtpfunnel`, now lives in -good (thanks!)

RESULTS

- Ice candidates gathered for a single transport
- Average time to data transfer significantly reduced
- Reduced footprint

FORWARD ERROR CORRECTION (FEC)

- Transport of a "protection stream"
- Only ULPFEC for now, API for new mechanisms not exposed yet

ULPFEC

- Uneven Level Protection
- Same ssrc, different payload type
- "Spliced" into the protected stream sequence number domain :(

SDP

```
m=video [...] 96 97  
[...]  
a=rtpmap:96 VP8/90000  
a=rtpmap:97 ulpfec/90000
```

IMPLEMENTATION

- New elements implemented by Pexip (thanks again)
- `rtpjitterbuffer` signals lost packets, decoder reconstructs
- Implementing a few `rtpbin` signals

RESULTS

- Latency / bandwidth tradeoff
- Improved resilience when faced with "normal" losses
- Works well with H264 and VP8, probably others
- Tested and working with audio (now)

DATA CHANNELS

- Matthew should present / have presented that

CROSS-BROWSER TESTING

- Known to work with major browsers (Chrome / Firefox / Safari)
- Recently shown to work with Microsoft Edge and react-native-webrtc
- Per-browser quirks :(

NEXT STEPS

- Congestion control
- Support for missing bundlePolicy (balanced)
- Better handling of browser-specific quirks
- FLEX FEC
- More renegotiation testing
- More usage!