Jingle & WebRTC

An introduction to audio/video calls with XMPP

Holger & Daniel

XMPP Meetup Berlin

June 10, 2020

Challenges of real time media

- Network
  - Connection establishment (through NAT)
  - Bandwidth managment
  - Stream control (a/v sync, packet loss, resend behaviour, . . .)[1]
  - Encryption
- Client side
  - (Hardware accelerated) encoding
  - Noise and echo detection

[1] The Video Conferencing Problem - Computerphile: youtube.com/watch?v=DkIhI59ysXI

WebRTC

- Web standard (W3C)
- Protocol stack based on other (IETF) standards
  - Interactive Connectivity Establishment (ICE)
  - Real-time Transport Protocol (RTP)
  - DTLS-SRTP (required)
  - Session Description Protocol (SDP)
- OPUS and VP8 + H.264 (optionally VP9, AV1 (future), ...)

libwebrtc

- *One* implementation of WebRTC
- *The original implementation*
  - bought by Google
  - formerly called libjingle
  - used in GTalk
- includes client side code (capturing, encoding, decoding, rendering, echo cancellation, . . .)

libwebrtc · Magic black box

- Does everything (codec selection, candidate selection, playing audio, . . .)
- very few public methods
- not very well documented but follows the browser APIs
- SDP output

Transporting SDP

```
v=0
m=audio 9 UDP/TLS/RTP/SAVPF 109 9 0 8 101
c=IN IP4 0.0.0.0
a=sendrecv
a=extmap:1 urn:ietf:params:rtp-hdrext:ssrc-audio-level
a=extmap:2/recvonly urn:ietf:params:rtp-hdrext:csrc-audio-level
a=extmap:3 urn:ietf:params:rtp-hdrext:sdes:mid
a=fmtp:109 maxplaybackrate=48000;stereo=1;useinbandfec=1
a=fmtp:101 0-15
a=mid:0
a=rtpmap:109 opus/48000/2
a=rtpmap:9 G722/8000/1
a=rtpmap:8 PCMA/8000
```

SDP ⇔ Jingle

- Required
  - XEP-0167: Jingle RTP Sessions
  - XEP-0176: Jingle ICE-UDP Transport Method
  - XEP-0320: Use of DTLS-SRTP in Jingle Sessions
- Extensions
  - XEP-0293: Jingle RTP Feedback Negotiation
  - XEP-0294: Jingle RTP Header Extensions Negotiation
  - XEP-0338: Jingle Grouping Framework
  - XEP-0339: Source-Specific Media Attributes in Jingle

State of XEPs

- Surprisingly clear / straight forward mapping of SDP to Jingle
- Some XEPs already Draft. Council has been calling the other
- `<rtcp-mux/>` has been used forever; Wasn't documented

Compatibility

- "Instantly" compatible with most clients that attempt to use the same stack (ICE, DTLS-SRTP, ...)
  - Movim
  - Siskin + Beagle
- No compatibility with clients that do something else (raw-udp, ZRTP, SRTP via `<encryption/>` as defined in XEP-0167 §7), ...)
  - Monal
  - probably other clients that predate WebRTC
- Compatible clients don't have to use libwebrtc; just reproduce the same stack
  - Jitsi[2]
  - aTalk [3]

[2] Not actually compatible. github.com/jitsi/jitsi/issues/679

[3] Not yet compatible. github.com/cmeng-git/atalk-android/issues/140

XEP-0353: Jingle Message Initiation

- Call (Ring) all devices with message instead of IQ
- Potentially have record of past calls in MAM
- Potentially call offline devices (Push)
- Incomplete (feature, tie break, . . .)

XEP-0215: External Service Discovery

- STUN/TURN server required for most connections
- XEP-0215 not used deployed in the wild
- Existing clients: Don't use STUN, hardcode Google's STUN server, or let the user configure it