

Simulcast

draft-ietf-mmusic-sdp-simulcast-00

Bo Burman, Magnus Westerlund,
Suhas Nandakumar, Mo Zanaty

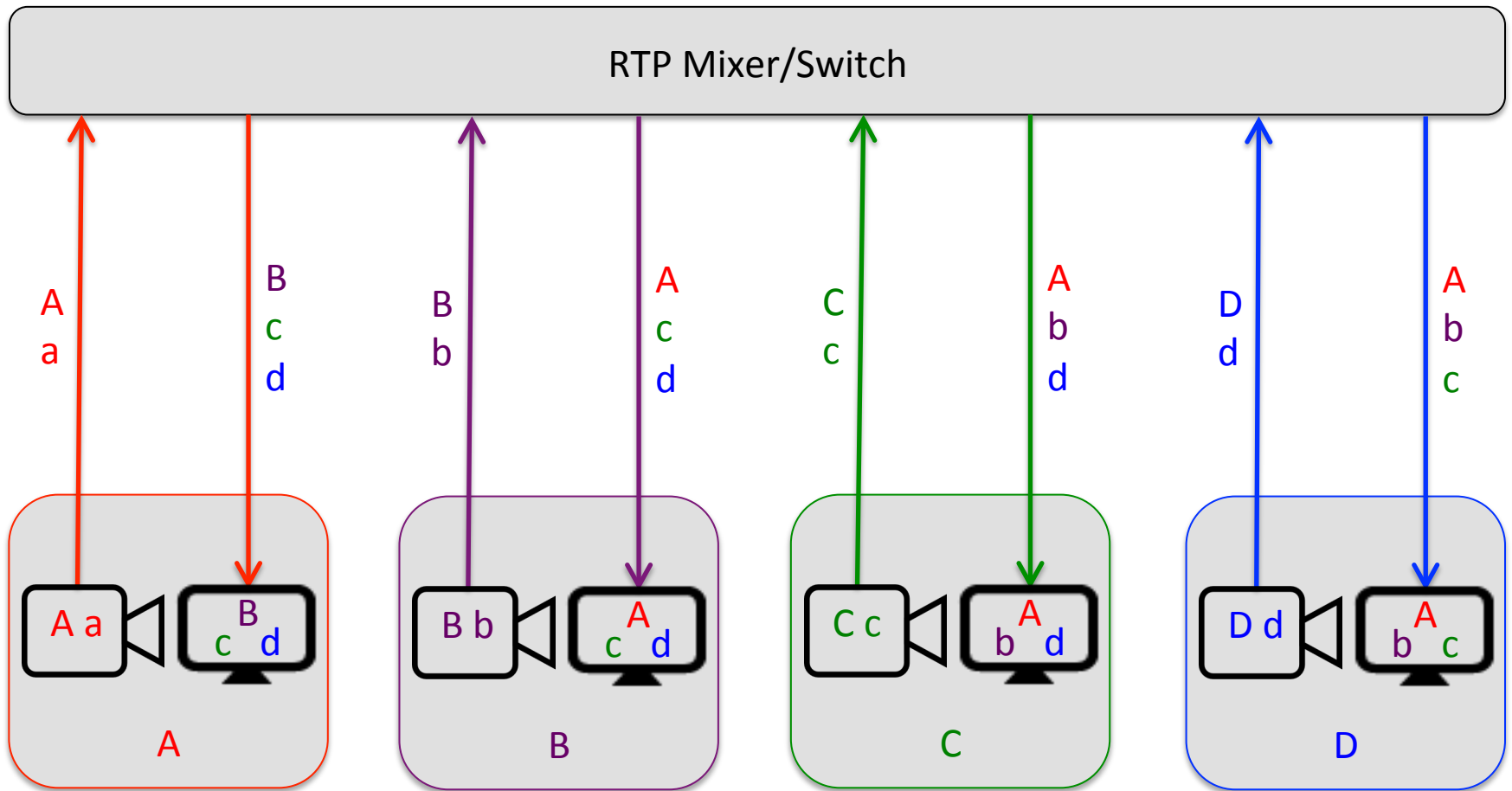
IETF 92

Note Well

IPR disclosure
from Microsoft
filed on this draft

<https://datatracker.ietf.org/ipr/2536/>

Review Common Use Case: Simulcast of HD + thumbnail to Mixer



Review Simulcast SDP

- Simulcast attribute expresses **concurrent** RTP streams, as a semi-colon separated list, in each direction.
- **Payload type** fully specifies each unique encoding for each simulcast stream.

```
m=video 10000 RTP/AVP 96 97
a=rtpmap:96 VP8/90000 (HD)
a=rtpmap:97 VP8/90000 (thumbnail)
a=fmtp:96 max-fs=3600;max-fr=30 (1280x720p30)
a=fmtp:97 max-fs=240;max-fr=15 (320x180p15)
a=simulcast send 96; 97; recv 96
(send HD + thumbnail, receive HD)
```

Open Issues

- Terminology should align with taxonomy
 - Simulcast “versions” are concurrent Source RTP Streams.
 - Simulcast “alternatives” are Payload Type format alternatives just like non-simulcast SDP, for dynamic switching not concurrent transmission.
- Paused Streams
- Bandwidth
- FEC Repair Streams
- Encoded/RTP Stream ID

Paused Streams

Current draft text in section 6.1:

Editor's note: Consider adding the possibility to put an RTP stream in "paused" state [I-D.ietf-avtext-rtp-stream-pause] from the beginning of the session, possibly starting it at a later point in time by applying RTP/RTCP level procedures from that specification.

Paused Streams

Proposed new text in section 6.1:

A simulcast version, and all its alternatives, may be put into a "paused" initial state by including the optional "paused" keyword after the media format, or after the list of comma-separated alternatives. The "paused" keyword is a declarative receiver preference that MAY be used for the simulcast direction of "recv" or "sendrecv", where it only applies to the receive direction, and MUST NOT be used for the simulcast direction of "send". Paused RTP streams may later be resumed by RTP/RTCP level procedures, as specified in [PAUSE]. The paused RTP sender MUST send an RTCP PAUSED (or TMMBN 0) notification of the local paused state to provide receivers with a target SSRC to resume.

Paused Streams

m=video 10000 RTP/AVP 96 97

a=rtpmap:96 VP8/90000 (HD)

a=rtpmap:97 VP8/90000 (thumbnail)

a=fmtp:96 max-fs=3600;max-fr=30 (1280x720p30)

a=fmtp:97 max-fs=240;max-fr=15 (320x180p15)

a=simulcast send 96; rcv 96 **paused; 97**

(send HD, receive thumbnail + **paused HD)**

Does PT work for BW, FEC, ESID?

- Bandwidth
 - VP8/VP9 lack bandwidth per PT
 - Consider a new codec-agnostic BW attribute?
 - Bind to PT directly or indirectly via other ID (ESID?)
- Repair flows
 - Retransmission already works, since it is already based on PT bindings.
 - FEC may use non-PT binding, perhaps ESID?

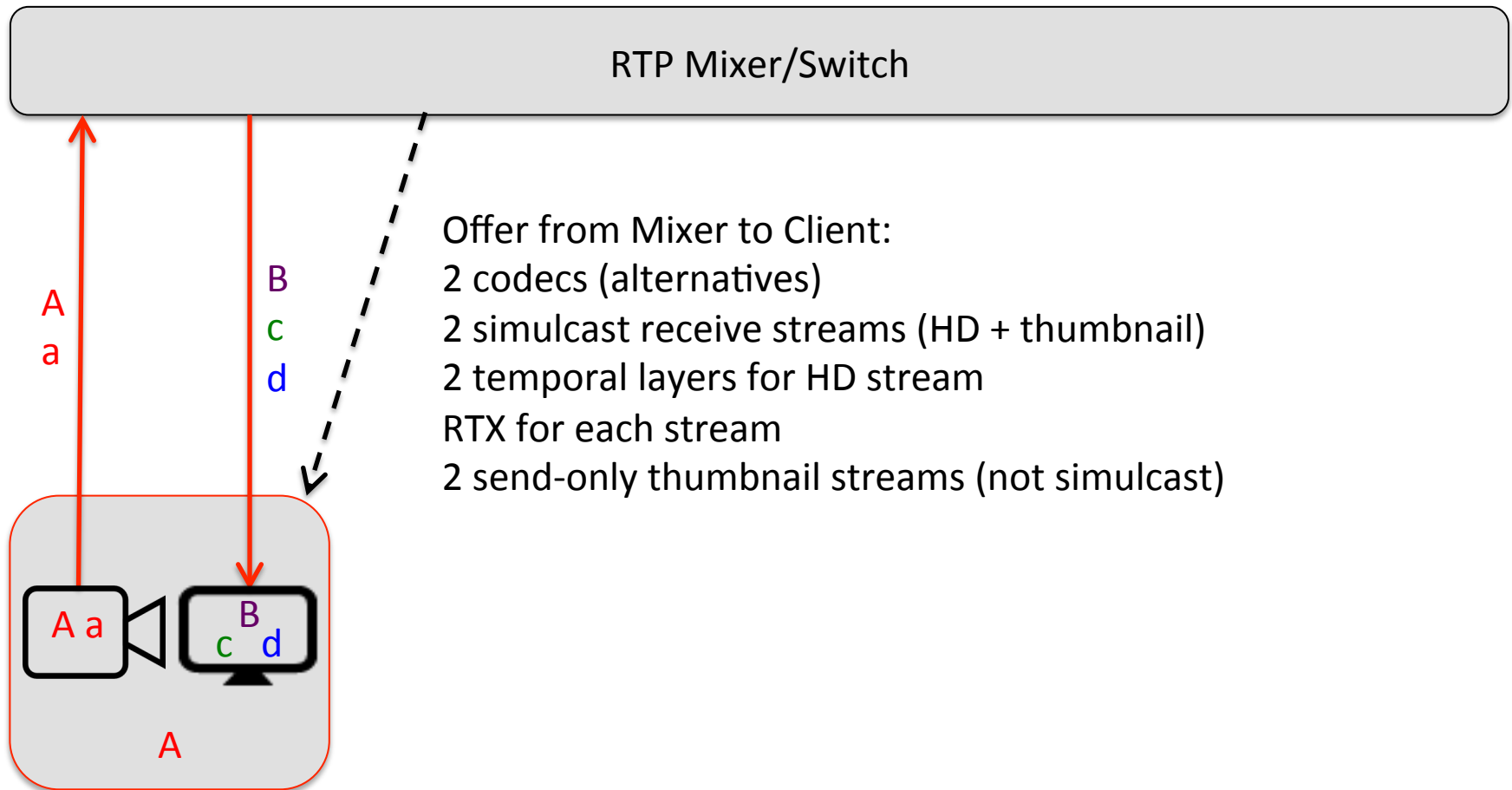
Next Steps

- Fix terminology to align with taxonomy.
- Add support for paused streams.
- Revisit PT binding approach for BW, FEC, ESID.

Backup Slides

PT vs. Other-ID

Review Common Use Case: Simulcast of HD + thumbnail to Mixer



PT approach (current draft)

```
m=video 10000 RTP/AVP 96 97 98 101 102 103 104 105 106 107 108 109
a=rtpmap:96 VP8/90000 (fmp: thumbnail 180p 15fps)
a=rtpmap:97 VP8/90000 (fmp: HD 720p 15fps)
a=rtpmap:98 VP8/90000 (fmp: HD 720p 30fps)
a=rtpmap:104 RTX/90000 (fmp: apt=96)
a=rtpmap:105 RTX/90000 (fmp: apt=97)
a=rtpmap:106 RTX/90000 (fmp: apt=98)
a=rtpmap:101 H264/90000 (fmp: thumbnail 180p 15fps)
a=rtpmap:102 H264/90000 (fmp: HD 720p 15fps)
a=rtpmap:103 H264/90000 (fmp: HD 720p 30fps)
a=rtpmap:107 RTX/90000 (fmp: apt=101)
a=rtpmap:108 RTX/90000 (fmp: apt=102)
a=rtpmap:109 RTX/90000 (fmp: apt=103)
a=depend:98 lay mid:97; 103 lay mid:102
a=simulcast recv 96,101; sendrecv 97,102; 98,103 (receive thumbnail; send+receive HD 15+30)
```

```
m=video 10000 RTP/AVP 96 101 // thumbnail 1 – not simulcast!
a=sendonly
m=video 10000 RTP/AVP 96 101 // thumbnail 2 – not simulcast!
a=sendonly
```

Other-ID approach (ESID?)

```
m=video 10000 RTP/AVP 96 97 98 101 102 103 104 105 106 107 108 109
a=rtpmap:96 VP8/90000 (fmp: thumbnail 180p 15fps)
a=rtpmap:97 VP8/90000 (fmp: HD 720p 15fps)
a=rtpmap:98 VP8/90000 (fmp: HD 720p 30fps)
a=rtpmap:104 RTX/90000 (fmp: apt=96)
a=rtpmap:105 RTX/90000 (fmp: apt=97)
a=rtpmap:106 RTX/90000 (fmp: apt=98)
a=rtpmap:101 H264/90000 (fmp: thumbnail 180p 15fps)
a=rtpmap:102 H264/90000 (fmp: HD 720p 15fps)
a=rtpmap:103 H264/90000 (fmp: HD 720p 30fps)
a=rtpmap:107 RTX/90000 (fmp: apt=101)
a=rtpmap:108 RTX/90000 (fmp: apt=102)
a=rtpmap:109 RTX/90000 (fmp: apt=103)
a=depend:98 lay mid:97; 103 lay mid:102
a=other-id:1 max-fs=180p max-fr=15
a=other-id:1 max-fs=180p max-fr=15
a=other-id:1 max-fs=180p max-fr=15
a=simulcast rcv 1:97,1:102; sendrcv 97,102; 98,103 (receive thumbnail; send+receive HD 15+30)
```

```
m=video 10000 RTP/AVP 96 101 // thumbnail 1 – not simulcast!
a=sendonly
m=video 10000 RTP/AVP 96 101 // thumbnail 2 – not simulcast!
a=sendonly
```