

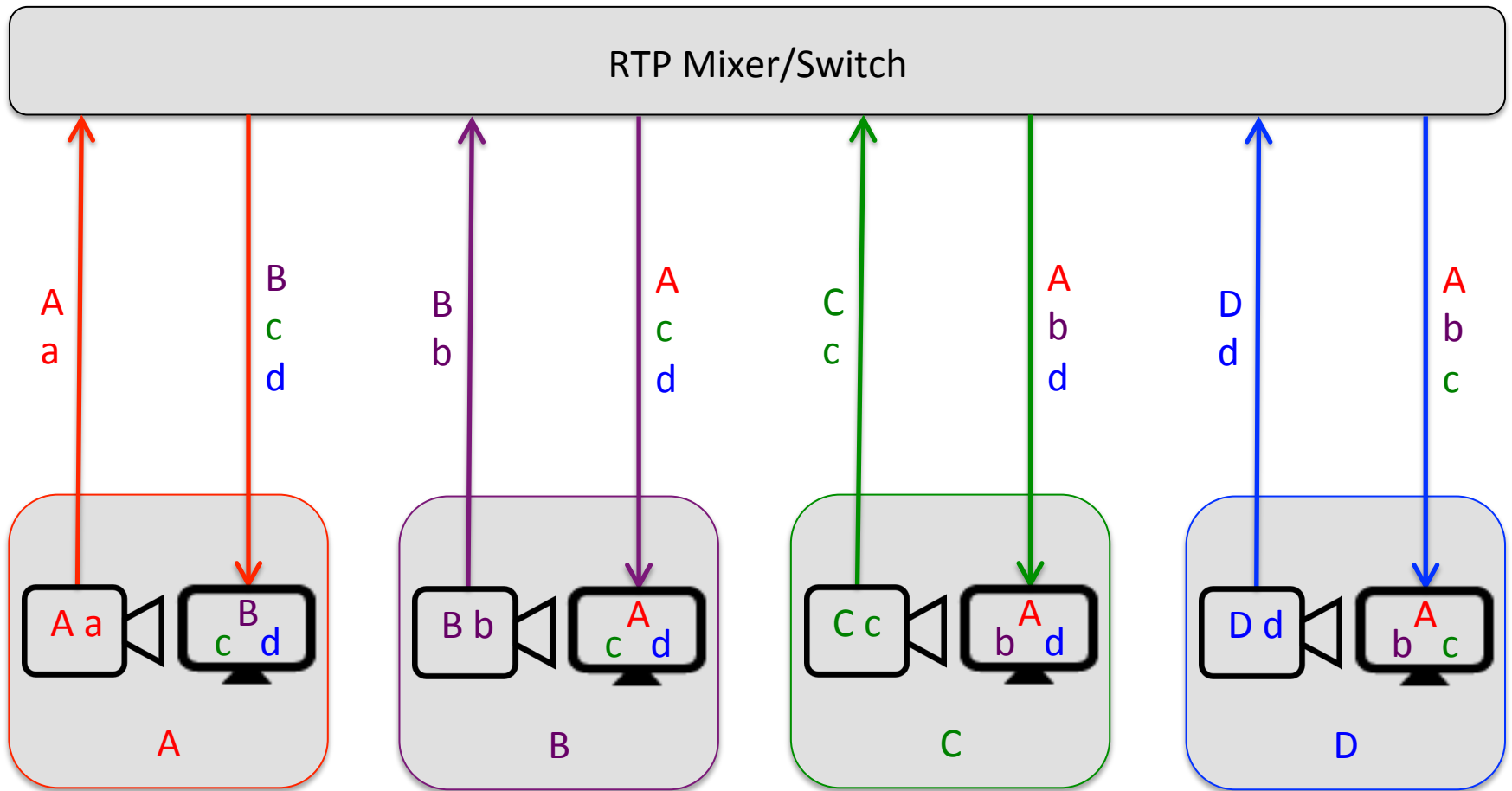
# MMUSIC RTP Simulcast

draft-westerlund-avtcore-rtp-simulcast-04

Bo Burman, Magnus Westerlund, Suhas Nandakumar

IETF 90

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



# Changes since -03

- Significantly simplified approach
  - Removed a=config-id concept
    - Rely on PT to fully specify media format of each simulcast stream
  - Removed a=sim-send-cap/sim-recv-cap
  - Replaced a=sim-send/sim-recv with simpler syntax
    - a=simulcast [send|recv|sendrecv] PT1; PT2...
- Support independent encodings and dependent scalable layers with similar syntax and semantics
- Support simulcast with bundled media using RTP mid
  - Defined in BUNDLE and reused here

# Simplified Approach

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)

# Simulcast Attribute

**a=simulcast [send|recv|sendrecv] PT1a,PT1b; PT2a,PT2b;...**

- A list of payload types for the indicated direction(s)
  - Comma-separated alternatives for the same simulcast stream
  - Semicolon-separated different simulcast 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; 97; recv 96**

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

# Simulcast Alternatives

**a=simulcast [send | rcv | sendrcv] PT1a,PT1b; PT2a,PT2b;...**

**a=simulcast [send | rcv | sendrcv] PT3; PT4;...**

- Comma-separated alternatives on same line for any combination
- Separate alternative lines for fixed combinations

- Example: PT=96-99=H264, PT=100-103=VP8

- Alternative lines for fixed combinations:

**a=simulcast send 96; 97; 98; 99; (4 streams of H264, or)**

**a=simulcast send 100; 101; 102; 103; (4 streams of VP8)**

- Alternatives on the same line for any combination:

**a=simulcast send 96,100; 97,101; 98,102; 99,103;**

**(4 streams of H264 or VP8 each)**

# Dependent Scalable Layers

**a=depend:PT1 lay mid:PT2;**

**a=simulcast [send | rcv | sendrcv] PT1; PT2;...**

- Support dependent scalable layers (sent as separate SSRCs) as simulcast
  - a=depend attribute from RFC 5583 signals the dependencies

m=video 10000 RTP/AVP 96 97

a=mid:v1

a=rtpmap:96 H264-SVC/90000 (1280x720p 30fps)

a=fmtp:96 profile-level-id=42400d; max-fs=3600; max-fps=3000; max-  
mbps=108000; mst-mode=NI-TC;

a=rtpmap:97 H264/90000 (1280x720p 15fps)

a=fmtp:97 profile-level-id=42400d; max-fs=3600; max-fps=1500; max-  
mbps=54000;

**a=depend:96 lay v1:97;**

**a=simulcast sendrcv 96; 97;**

**(30 and 15 fps temporal layers in both directions)**

# Simulcast with Bundled Media

- Use RTP mid (defined in BUNDLE) to identify simulcast streams from the same source, and associate the source to its media description (m-line).

m=video 10000 RTP/AVP 96 97

**a=mid:v1**

**a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid**

a=rtpmap:96 VP8/90000 (HD)

a=rtpmap:97 VP8/90000 (thumbnail)

**a=simulcast send 96; 97; recv 96**

m=video 10000 RTP/AVP 96 97

**a=mid:v2**

**a=extmap:1 urn:ietf:params:rtp-hdext:sdes:mid**

a=rtpmap:96 VP8/90000 (HD)

a=rtpmap:97 VP8/90000 (thumbnail)

**a=simulcast send 96; 97; recv 96**



# Next Steps

- Are the simulcast use cases compelling to progress the work?
- Are the proposed simulcast semantics sufficient for most use cases?
- Repair flows
  - Retransmission already works, since it is already based on PT bindings.
  - FEC needs more work, many variants.