Source-Specific Media Attributes

Jonathan Lennox
Joerg Ott
Thomas Schierl

draft-lennox-mmusic-sdp-source-attributes-01.txt

Source-Specific Attributes: Review

- RTP allows multiple sources in an RTP session, but SDP has no way to signal this.
- Solution: define an SDP attribute for characteristics of a source.

```
m=video 49170 RTP/AVP 96
a=rtpmap:96 H264/90000
a=ssrc:12345 cname:stream1@example.com
a=ssrc:67890 cname:stream2@example.com
```

- Map SDP “source-specific” attributes into the `ssrc` attribute.
- This generalizes material that was previously in the RTP Single-Source Multicast draft.

Motivation

- Avoid clashes with the SSRC id of a single media sender.
  - This is needed for Single-Source Multicast.
- Make SSRC multiplexing explicit.
  - Differentiate between multiple SSRCs coming from the same endpoint in the same RTP session.
  - Indicate relationships among SSRCs in an RTP session.
  - Examples:
    - Multiple cameras
    - FEC
    - Retransmission
    - RTP translators

Terminology Confusion: SDP vs. RTP

- SDP and RTP terminology is inconsistent.
- An SDP *Media Stream* describes an RTP *Session.*

SDP	RTP
Multimedia Session	–
Media Stream, Media Description, m= line	RTP Session
Media Source	(Audio/Video) Stream, RTP Stream, RTP Source, ssrc

- Introduce the term *Media Source* for SDP.

RTP Architectural Issues

- Discussed in AVT in Prague.
- Essentially, no change to RTP stack behaviors, except to constrain an endpoint's choice of SSRC.
  - Still MUST NOT multiplex (e.g.) audio and video streams on a single RTP session.
  - Still a single set of payload type numbers per RTP session.
  - Still MUST do collision detection (with slight variant that an explicit source "wins").
  - If you only know about a source from SDP, it doesn't count for group size estimation, and you don't send SR/RR report blocks for it.

Open Issues

- IANA registration issues
  - Need to define rules for attribute registry.
    - Proposal: same as existing SDP attribute registries.
  - Some media-level parameters (group semantics) are re-used at the source level.
    - These should have a separate IANA registry definition.
- Needs to work with capability negotiation.
  - “Mutually oblivious” standards are probably fine.
  - Treat a=ssrc: like any other attribute.
- MIKEY (RFC 3830, signaled in SDP with RFC 4567) also specifies SSRCs in SDP.
  - Proposal: if you use both, they MUST be consistent.
  - Does more need to be said?

Backward Compatibility

- Many (point-to-point) endpoints won't correctly handle RTP sessions with more than one source.
- Even endpoints which correctly interpret multiple sources might not have the resources to decode them.
- Should there be an attribute for "I can handle at most *N* sources"?
- If so, recommend that if your receiver doesn't send this parameter, you SHOULD send only one source at a time.
- This could be in a separate draft.

Source-specific fmtp

- Describes source-specific codec parameters.
  - Parameters describing the *sender's* bitstream.
  - Motivation: H.264 sprop-* parameters
- If you have multiple sources, out-of-band parameters may not be the same for each source.
  - E.g., video switching, multiple cameras.

```
m=video 49170 RTP/AVP 96
a=rtpmap:96 H264/90000
a=ssrc:12345 cname:stream1@example.com
a=ssrc:12345 fmtp:96 sprop-parameter-sets=XXX
a=ssrc:67890 cname:stream2@example.com
a=ssrc:67890 fmtp:96 sprop-parameter-sets=YYY
```

- This draft doesn't define any (codec-specific) usages of source-specific fmtp.
- Need to figure out backward compatibility issues.

Next Steps

- Need to move quickly on the base spec (RTCP-SSM dependency).
  - Can this be a working group item of MMUSIC?
- What additional work is desirable?
  - Should any of it be added to this draft?