

RTP Payload Format Constraints

draft-pthatcher-mmusic-rid-02

Adam Roach

Yokohama, Japan

Monday, November 2nd, 2015

Overview

- Defines a new identifier, “RID”
- In SDP, negotiates constraints to be applied to a Source RTP Stream
- In RTCP (and RTP headers), carries an identifier to bind a Source Stream to its negotiated constraints
- Key motivator is constraining and identifying multiple encodings in Simulcast

Example: Sending two encodings

m=video 10000 RTP/SAVPF 98

a=rtpmap:98 VP8/90000

...

a=rid:1 send max-width=1280;max-height=720;max-fps=30

a=rid:2 recv max-width=1280;max-height=720;max-fps=30

a=rid:3 send max-width=320;max-height=180;max-fps=15

a=simulcast:send rid=1;3 recv rid=2

Context: We're in a time crunch

- The W3C WebRTC working group is putting the final touches on the WebRTC 1.0 specification.
- Anything that we can't get defined by the time it's ready to be published won't be included.
- Unless we have SDP signaling defined to do Simulcast when WebRTC is published, WebRTC won't have SDP syntax to use.
 - Yes, this is a tautology.
- I plan to drive this draft to completion on the list in the coming weeks. If you care about the outcome of this work, please keep up.

Open Issue 1: Declarative SDP?

- Currently only define semantics for offer/answer (RFC3264) usage.
- The RID mechanism is primarily motivated by simulcast *negotiation* cases.
- Is the utility of being able to declare RIDs sufficiently useful that we want to define its usage at this time?
 - We could always add it later in a different document if we decide not to but change our minds.
- Recommendation: offer/answer only for now.

Open Issue 2: Definition of Bitrate

- Currently, we have our own definition, which we believe is compatible with (but more complete than) TIAS:
 - max-br, for bit rate in bits per second. The restriction applies to the media payload only, and does not include overhead introduced by other layers (e.g., RTP, UDP, IP, or Ethernet). The exact means of keeping within this limit are left up to the implementation, and instantaneous excursions outside the limit are permissible. For any given one-second sliding window, however, the total number of bits in the payload portion of RTP SHOULD NOT exceed the value specified in "max-br."
- Is there something external we should reference?

Open Issue 3: Escaping in Extension Parameters

- The parameters on an “a=rid:” line are extensible. The syntax for these is:

rid-param-other = 1*(alpha-numeric / "-")

["=" param-val]

param-val = *(%x20-58 / %x60-7E)

; Any printable character except semicolon

- If an extension has values that can contain semicolons, they need an escaping mechanism.

Open Issue 3: Escaping in Extension Parameters, cont.

- Note that this is not an issue for any currently defined parameters, as they all take numeric values only.
- Options:
 1. Change extension syntax to only allow numeric values
 2. Define a universal escaping mechanism for all extensions to use
 3. Leave this problem for the first extension parameter – if any – to define value in a way that might allow a semicolon
- Recommendation: Since the chance of a constraint being non-numeric – much less allowing a semicolon – seems quite low, #3 seems to make the most sense.

Background: Negotiating Supported Parameters, cont.

- Where the offer contains specific values, the answer can make them more constrained; e.g., with an offer of:
`a=rid:1 send max-width=1024;max-height=768`
- The answer can tweak them down:
`a=rid:1 recv max-width=640;max-height=480`

Background: Negotiating Supported Parameters

- Mechanism allows offerer to indicate supported parameters without constraining them:

```
a=rid:1 send max-width;max-height;max-goats
```

- This allows answer to propose constraints where the offerer doesn't:

```
a=rid:1 recv max-width=1024;max-height=768
```

Background: Negotiating Supported Parameters, cont.

- If the offer contains a parameter the answerer does not understand:
 - a=rid:1 recv max-br=64000;max-goats=72
 - a=rid:2 recv max-br=64000
 - a=rid:3 recv max-br=32000;max-goats=16
 - a=rid:4 recv max-br=32000
 - a=simulcast:recv rid=1,2;3,4 *Semicolons and commas mean very different things!*
- The answer removes all the rids with unknown parameters:
 - a=rid:2 send max-br=64000
 - a=rid:4 send max-br=32000
 - a=simulcast:send rid=2;4

Background: Negotiating Supported Parameters, cont.

- This all works, but two specific optimizations have been proposed on the mailing list.
 - One allows for a syntax that makes constraints “best effort,” which allows for fewer SDP gyrations for situations like the one on the previous slide
 - The other proposes that unknown parameters in the send direction can be safely ignored

Proposed Enhancement 1: Soft Constraints

- If an offerer would like to specify a constraint to be honored if the remote party understands it *and ignored if the remote party does not*, we add a doodad to the syntax to indicate this fact.
- The previous example collapses to:
a=rid:1 recv max-br=64000;max-goats?72
a=rid:3 recv max-br=32000;max-goats?16
a=simulcast:recv rid=1;3
- The answer removes soft constraints it doesn't know, but keeps the associated RIDs:
a=rid:1 send max-br=64000
a=rid:3 send max-br=32000
a=simulcast:send rid=1;3

Proposed Enhancement 2: Asymmetric Handling

- It has been observed that it should not matter if the answerer doesn't understand a constraint on a stream that it will receive
 - If the stream is being constrained in a way it doesn't understand, why should it care?
- Based on this, it would seem to make sense to have the answerer simply remove unsupported parameters from any received "a=rid:x send" lines (i.e. for streams the answerer will be receiving)
 - This makes the rules *different* based on whether the line is "send" or "recv"

Proposed Enhancement 2: Asymmetric Handling, example 1

- Offer:
a=rid:1 rcv max-br=64000;max-goats=72
a=rid:2 rcv max-br=64000
a=rid:3 rcv max-br=32000;max-goats=16
a=rid:4 rcv max-br=32000
a=simulcast:rcv rid=1,2;3,4
- Answer (doesn't know max-goats):
a=rid:2 send max-br=64000
a=rid:4 send max-br=32000
a=simulcast:send rid=2;4

Proposed Enhancement 2: Asymmetric Handling, example 2

- Offer:
 - a=rid:1 send max-br=64000;max-goats=72
 - a=rid:2 send max-br=64000
 - a=rid:3 send max-br=32000;max-goats=16
 - a=rid:4 send max-br=32000
 - a=simulcast:send rid=1,2;3,4
- Answer (doesn't know max-goats):
 - a=rid:1 recv max-br=64000
 - a=rid:2 recv max-br=64000
 - a=rid:3 recv max-br=32000
 - a=rid:4 recv max-br=32000
 - a=simulcast:recv rid=1,2;3,4