Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 17, 2016

P. Thatcher
Google
M. Zanaty
S. Nandakumar
Cisco Systems
A. Roach
Mozilla
B. Burman
Ericsson
B. Campen
Mozilla

October 15, 2015

RTP Payload Format Constraints
draft-pthatcher-mmusic-rid-01

Abstract

In this specification, we define a framework for identifying Source RTP Streams with the constraints on its payload format in the Session Description Protocol. This framework uses "rid" SDP attribute to: a) effectively identify the Source RTP Streams within a RTP Session, b) constrain their payload format parameters in a codec-agnostic way beyond what is provided with the regular Payload Types and c) enable unambiguous mapping between the Source RTP Streams to their media format specification in the SDP.

Note-1: The name 'rid' is not yet finalized. Please refer to Section 12 for more details on the naming.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Payload Type (PT) in RTP provides mapping between the format of the RTP payload and the media format description specified in the signaling. For applications that use SDP for signaling, the constructs rtpmap and/or fmtp describe the characteristics of the media that is carried in the RTP payload, mapped to a given PT.

Recent advances in standards such as RTCWEB and NETVC have given rise to rich multimedia applications requiring support for multiple RTP Streams with in a RTP session [I-D.ietf-mmusic-sdp-bundle-negotiation], [I-D.ietf-mmusic-sdp-simulcast] or having to support multiple codecs, for example. These demands have unearthed challenges inherent with:

- The restricted RTP PT space in specifying the various payload configurations,

- The codec-specific constructs for the payload formats in SDP,

- Missing or underspecied payload format parameters,

- Ambiguity in mapping between the individual Source RTP Streams and their equivalent format specification in the SDP.

This specification defines a new SDP framework for constraining Source RTP Streams (Section 2.1.10 [I-D.ietf-avtext-rtp-grouping-taxonomy]), called "Restriction Identifier (rid)", along with the SDP attributes to constrain their payload formats in a codec-agnostic way. The "rid" framework can be thought of as complementary extension to the way the media format parameters are specified in SDP today, via the "a=fmtp" attribute. This specification also proposes a new RTP header extension to carry the "rid" value, to provide correlation between the RTP Packets and their format specification in the SDP.

Note that the "rid" parameters only serve to further constrain the parameters that are established on a PT format. They do not relax any existing constraints.

As described in Section 7.2.1, this mechanism achieves backwards compatibility via the normal SDP processing rules, which require unknown a= parameters to be ignored. This means that implementations need to be prepared to handle successful offers and answers from other implementations that neither indicate nor honor the constraints requested by this mechanism.

Further, as described in Section 7 and its subsections, this mechanism achieves extensibility by: (a) having offerers include all supported constraints in their offer, abd (b) having answerers ignore a=rid lines that specify unknown constraints.

2. Key Words for Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]

3. Terminology

The terms Source RTP Stream, Endpoint, RTP Session, and RTP Stream are used as defined in [I-D.ietf-avtext-rtp-grouping-taxonomy].

[RFC4566] and [RFC3264] terminology is also used where appropriate.

4. Motivation

This section summarizes several motivations for proposing the "rid" framework.

1. RTP PT Space Exhaustion: [RFC3550] defines payload type (PT) that identifies the format of the RTP payload and determine its interpretation by the application. [RFC3550] assigns 7 bits for the PT in the RTP header. However, the assignment of static mapping of payload codes to payload formats and multiplexing of RTP with other protocols (such as RTCP) could result in limited number of payload type numbers available for the application usage. In scenarios where the number of possible RTP payload configurations exceed the available PT space within a RTP Session, there is need a way to represent the additional payload configurations and also to effectively map a Source RTP Stream to its configuration in the signaling.

2. Codec-Specific Media Format Specification in SDP: RTP Payload configuration is typically specified using rtpmap and fmtp SDP attributes. The rtpmap attribute provides the media format to RTP PT mapping and the ftmp attribute describes the media format specific parameters. The syntax for the fmtp attribute is

tightly coupled to a specific media format (such as H.264, H.265, VP8). This has resulted in a myriad ways for defining the attributes that are common across different media formats. Additionally, with the advent of new standards efforts such as NETVC, one can expect more media formats to be standardized in the future. Thus, there is a need to define common media characteristics in a codec-agnostic way in order to reduce the duplicated efforts and to simplify the syntactic representation across the different codec standards.

3. Multi-source and Multi-stream Use Cases: Recently, there is a rising trend with real-time multimedia applications supporting multiple sources per endpoint with various temporal resolutions (Scalable Video Codec) and spatial resolutions (Simulcast) per source. These applications are being challenged by the limited RTP PT space and/or by the underspecified SDP constructs for exercising granular control on configuring the individual Source RTP Streams.

5. SDP 'rid' Media Level Attribute

This section defines new SDP media-level attribute [RFC4566], "a=rid". Roughly speaking, this attribute takes the following form (see Section 10 for a formal definition).

```
a=rid:<rid-identifier> <direction> pt=<fmt-list>;<constraint>=<value>...
```

A given "a=rid" SDP media attribute specifies constraints defining an unique RTP payload configuration identified via the "rid-identifier". A set of codec-agnostic "rid-level" constraints are defined (Section 6) that describe the media format specification applicable to one or more Payload Types speicified by the "a=rid" line.

The 'rid' framework MAY be used in combination with the 'a=fmtp' SDP attribute for describing the media format parameters for a given RTP Payload Type. However in such scenarios, the 'rid-level' constraints (Section 6) further constrains the equivalent 'fmtp' attributes.

The 'direction' identifies the either 'send', 'recv' directionality of the Source RTP Stream.

A given SDP media description MAY have zero or more "a=rid" lines describing various possible RTP payload configurations. A given 'rid-identifier' MUST not be repeated in a given media description.

The 'rid' media attribute MAY be used for any RTP-based media transport. It is not defined for other transports.

Though the 'rid-level' attributes specified by the 'rid' property follow the syntax similar to session-level and media-level attributes, they are defined independently. All 'rid-level' attributes MUST be registered with IANA, using the registry defined in Section 13

Section 10 gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for the "rid" attribute.

The "a=rid" media attribute is not dependent on charset.

6. 'rid-level' constraints

This section defines the 'rid-level' constraints that can be used to constrain the RTP payload encoding format in a codec-agnostic way.

The following constraints are intended to apply to video codecs in a codec-independent fashion.

- max-width, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the width of the stream when a rotation of zero degrees is encoded.

- max-height, for spatial resolution in pixels. In the case that stream orientation signaling is used to modify the intended display orientation, this attribute refers to the width of the stream when a rotation of zero degrees is encoded.

- max-fps, for frame rate in frames per second. For encoders that do not use a fixed framerate for encoding, this value should constrain the minimum amount of time between frames: the time between any two consecutive frames SHOULD not be less than 1/max-fps seconds.

- max-fs, for frame size in pixels per frame.

- 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."

- max-pps, for pixel rate in pixels per second. This value SHOULD be handled identically to max-fps, after performing the following

conversion: max-fps = max-pps / (width * height). If the stream resolution changes, this value is recalculated. Due to this recalculation, excursions outside the specified maximum are possible during near resolution change boundaries.

All the constraints are optional and are subjected to negotiation based on the SDP Offer/Answer rules described in Section 7

This list is intended to be an initial set of constraints; future documents may define additional constraints; see Section 13.4. While this document doesn't define constraints for audio codecs, there is no reason such constraints should be precluded from definition and registration by other documents.

Section 10 provides formal Augmented Backus-Naur Form(ABNF) [RFC5234] grammar for each of the "rid-level" attributes defined in this section.

7. SDP Offer/Answer Procedures

This section describes the SDP Offer/Answer [RFC3264] procedures when using the 'rid' framework.

7.1. Generating the Initial SDP Offer

For each media description in the offer, the offerer MAY choose to include one or more "a=rid" lines to specify a configuration profile for the given set of RTP Payload Types.

In order to construct a given "a=rid" line, the offerer must follow the below steps:

1. It MUST generate a 'rid-identifier' that is unique within a media description

2. It MUST set the direction for the 'rid-identifier' to one of 'send' or 'recv'

3. It MAY include a listing of SDP format tokens (usually corresponding to RTP payload types) to which the constraints expressed by the 'rid-level' attributes apply. Any Payload Types chosen MUST either be defined as part of "a=rtpmap" or "a=fmtp" attributes.

4. The Offerer then chooses the 'rid-level' constraints (Section 6) to be applied for the rid, and adds them to the "a=rid" line. If it wishes the answer to have the ability to specify a constraint, but does not wish to set a value itself, it MUST include the name

of the constraint in the "a=rid" line, but without any indicated value.

Note: If an 'a=fmtp' attribute is also used to provide media-format-specific parameters, then the 'rid-level' attributes will further constrain the equivalent 'fmtp' parameters for the given Payload Type for those streams associated with the 'rid'.

7.2. Answerer processing the SDP Offer

For each media description in the offer, and for each "a=rid" attribute in the media description, the receiver of the offer will perform the following steps:

7.2.1. 'rid' unaware Answerer

If the receiver doesn't support the 'rid' framework proposed in this specification, the entire "a=rid" line is ignored following the standard [RFC3264] Offer/Answer rules. If a given codec would require 'a=fmtp' line when used without "a=rid" then the offer still needs to include that even when using RID.

7.2.2. 'rid' aware Answerer

If the answerer supports 'rid' framework, the following steps are executed, in order, for each "a=rid" line in a given media description:

1. Extract the rid-identifier from the "a=rid" line and verify its uniqueness. In the case of a duplicate, the entire "a=rid" line is rejected and MUST not be included in the SDP Answer.

2. If the "a=rid" line contains a "pt=" parameter, the list of payload types is verified against the list obtained from "a=rtpmap" and/or "a=fmtp" attributes. If there is no match for the Payload Type listed in the "a=rid" line, then remove the "a=rid" line.

3. The answerer ensures that "rid-level" parameters listed are supported and syntactically well formed. In the case of a syntax error or an unsupported parameter, the "a=rid" line is removed.

4. If the 'depend' rid-level attribute is included, the answerer MUST make sure that the rid-identifiers listed unambiguously match the rid-identifiers in the SDP offer. Any lines that do not are removed.

5. if the "a=rid" line contains a "pt=" parameter, the answerer verifies that the attribute values provided in the "rid-level" attributes are consistent with the corrsponding codecs and their other parameters. See Section 9 for more detail. If the rid-level parameters are incompatible with the other codec properties, then the "a=rid" line is removed.

7.3. Generating the SDP Answer

Having performed the verification of the SDP offer as described, the answerer shall perform the following steps to generate the SDP answer.

For each "a=rid" line:

1. The answerer MAY choose to modify specific 'rid-level' attribute value in the answer SDP. In such a case, the modified value MUST be more constrained than the ones specified in the offer. The answer MUST NOT include any constraints that were not present in the offer.

2. The answerer MUST NOT modify the 'rid-identifier' present in the offer.

3. The answerer is allowed to remove one or more media formats from a given 'a=rid' line. If the answerer chooses to remove all the media format tokens from an "a=rid" line, the answerer MUST remove the entire "a=rid" line.

4. In cases where the answerer is unable to support the payload configuration specified in a given "a=rid" line in the offer, the answerer MUST remove the corresponding "a=rid" line. This includes situations in which the answerer does not understand one or more of the constraints in the "a=rid" line that has an associated value.

7.4. Offering Processing of the SDP Answer

The offerer shall follow the steps similar to answerer's offer processing with the following exceptions

1. The offerer MUST ensure that the 'rid-identifiers' aren't changed between the offer and the answer. If so, the offerer MUST consider the corresponding 'a=rid' line as rejected.

2. If there exist changes in the 'rid-level' attribute values, the offerer MUST ensure that the modifications can be supported or else consider the "a=rid" line as rejected.

3. If the SDP answer contains any "rid-identifier" that doesn't match with the offer, the offerer MUST ignore the corresponding "a=rid" line.

4. if the "a=rid" line contains a "pt=" parameter, the offerer verifies that the attribute values provided in the "rid-level" attributes are consistent with the corrsponding codecs and their other parameters. See Section 9 for more detail. If the rid-level parameters are incompatible with the other codec properties, then the "a=rid" line is removed.

7.5. Modifying the Session

Offers and answers inside an existing session follow the rules for initial session negotiation. Such an offer MAY propose a change the number of RIDs in use. To avoid race conditions with media, any RIDs with proposed changes SHOULD use a new ID, rather than re-using one from the previous offer/answer exchange. RIDs without proposed changes SHOULD re-use the ID from the previous exchange.

8. Usage of 'rid' in RTP and RTCP

The RTP fixed header includes the payload type number and the SSRC values of the RTP stream. RTP defines how you de-multiplex streams within an RTP session, but in some use cases applications need further identifiers in order to effectively map the individual RTP Streams to their equivalent payload configurations in the SDP.

This specification defines a new RTP header extension [RFC5285] to include the 'rid-identifier'. This makes it possible for a receiver to associate received RTP packets (identifying the Source RTP Stream) with a media description having the format constraint specified. This specification also defines a new RTCP SDES item [RFC3550], 'RID', which is used to carry rids within RTCP SDES packets.

8.1. RTP 'rid' Header Extension

The payload, containing the identification-tag, of the RTP 'rid-identifier' header extension element can be encoded using either the one-byte or two-byte header [RFC5285]. The identification-tag payload is UTF-8 encoded, as in SDP.

As the identification-tag is included in an RTP header extension, there should be some consideration about the packet expansion caused by the identification-tag. To avoid Maximum Transmission Unit (MTU) issues for the RTP packets, the header extension's size needs to be taken into account when the encoding media. Note that set of header

extensions included in the packet needs to be padded to the next 32-bit boundary using zero bytes [RFC5285]

It is recommended that the identification-tag is kept short. Due to the properties of the RTP header extension mechanism, when using the one-byte header, a tag that is 1-3 bytes will result in that a minimal number of 32-bit words are used for the RTP header extension, in case no other header extensions are included at the same time. In many cases, a one-byte tag will be sufficient; it is RECOMMENDED that implementations use the shortest tag that fits their purposes.

8.2. RTCP 'RID' SDES Extension

```
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      MID=TBD  |     length    | rid                         ...
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The rid payload is UTF-8 encoded and is not null-terminated.

> RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

9. Interaction with Other Techniques

Historically, a number of other approaches have been defined that allow constraining media streams via SDP parameters. These include:

- Codec-specific configuration set via format parameters ("a=fmtp"); for example, the H.264 "max-fs" format parameter

- Size restrictions imposed by image attribute attributes ("a=imgattr") [RFC6236]

When the mechanism described in this document is used in conjunction with these other restricting mechanisms, it is intended to impose additional restrictions beyond those communicated in other techniques.

In an offer, this means that a=rid lines, when combined with other restrictions on the media stream, are expected to result in a non-empty union. For example, if image attributes are used to indicate that a PT has a minimum width of 640, then specification of "max-width=320" in an "a=rid" line that is then applied to that PT is nonsensical. According to the rules of Section 7.2.2, this will result in the corresponding "a=rid" line being ignored by the recipient.

Similarly, an answer the a=rid lines, when combined with the other restrictions on the media stream, are also expected to result in a non-empty union. If the implementation generating an answer wishes to restrict a property of the stream below that which would be allowed by other parameters (e.g., those specified in "a=fmtp" or "a=imgattr"), its only recourse is to remove the "a=rid" line altogether, as described in Section 7.3. If it instead attempts to constrain the stream beyond what is allowed by other mechanisms, then the offerer will ignore the corresponding "a=rid" line, as described in Section 7.4.

10. Formal Grammar

This section gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar for each of the new media and rid-level attributes defined in this document.

```
   rid-syntax        = "a=rid:" rid-identifier SP rid-dir
                       [ rid-pt-param-list / rid-param-list ]

   rid-identifier    = 1*(alpha-numeric / "-" / "_")

   rid-dir           = "send" / "recv"

   rid-pt-param-list = SP rid-fmt-list *(";" rid-param)

   rid-param-list    = SP rid-param *(";" rid-param)

   rid-fmt-list      = "pt=" fmt *( "," fmt )
                        ; fmt defined in {{RFC4566}}

   rid-param         = rid-width-param
                       / rid-height-param
                       / rid-fps-param
                       / rid-fs-param
                       / rid-br-param
                       / rid-pps-param
                       / rid-depend-param
                       / rid-param-other

   rid-width-param   = "max-width" [ "=" int-param-val ]

   rid-height-param  = "max-height" [ "=" int-param-val ]

   rid-fps-param     = "max-fps" [ "=" int-param-val ]

   rid-fs-param      = "max-fs" [ "=" int-param-val ]

   rid-br-param      = "max-br" [ "=" int-param-val ]

   rid-pps-param     = "max-pps" [ "=" int-param-val ]

   rid-depend-param  = "depend=" rid-list

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

   rid-list          = rid-identifier *( "," rid-identifier )

   int-param-val     = 1*DIGIT

   param-val         = *( %x20-58 / %x60-7E )
                       ; Any printable character except semicolon
```

11. SDP Examples

11.1. Many Bundled Streams using Many Codecs

In this scenario, the offerer supports the Opus, G.722, G.711 and DTMF audio codecs, and VP8, VP9, H.264 (CBP/CHP, mode 0/1), H.264-SVC (SCBP/SCHP) and H.265 (MP/M10P) for video. An 8-way video call (to a mixer) is supported (send 1 and receive 7 video streams) by offering 7 video media sections (1 sendrecv at max resolution and 6 recvonly at smaller resolutions), all bundled on the same port, using 3 different resolutions. The resolutions include:

- 1 receive stream of 720p resolution is offered for the active speaker.

- 2 receive streams of 360p resolution are offered for the prior 2 active speakers.

- 4 receive streams of 180p resolution are offered for others in the call.

Expressing all these codecs and resolutions using 32 dynamic PTs (2 audio + 10x3 video) would exhaust the primary dynamic space (96-127). RIDs are used to avoid PT exhaustion and express the resolution constraints.

NOTE: The SDP given below skips few lines to keep the example short and focused, as indicated by either the "..." or the comments inserted.

Example 1

```
Offer:
...
m=audio 10000 RTP/SAVPF 96 9 8 0 123
a=rtpmap:96 OPUS/48000
a=rtpmap:9 G722/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:123 telephone-event/8000
a=mid:a1
...
m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
a=rtpmap:98 VP8/90000
a=fmtp:98 max-fs=3600; max-fr=30
a=rtpmap:99 VP9/90000
a=fmtp:99 max-fs=3600; max-fr=30
```

```
   a=rtpmap:100 H264/90000
   a=fmtp:100 profile-level-id=42401f; packetization-mode=0
   a=rtpmap:101 H264/90000
   a=fmtp:101 profile-level-id=42401f; packetization-mode=1
   a=rtpmap:102 H264/90000
   a=fmtp:102 profile-level-id=640c1f; packetization-mode=0
   a=rtpmap:103 H264/90000
   a=fmtp:103 profile-level-id=640c1f; packetization-mode=1
   a=rtpmap:104 H264-SVC/90000
   a=fmtp:104 profile-level-id=530c1f
   a=rtpmap:105 H264-SVC/90000
   a=fmtp:105 profile-level-id=560c1f
   a=rtpmap:106 H265/90000
   a=fmtp:106 profile-id=1; level-id=93
   a=rtpmap:107 H265/90000
   a=fmtp:107 profile-id=2; level-id=93
   a=sendrecv
   a=mid:v1 (max resolution)
   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
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v2 (medium resolution)
   a=rid:3 recv max-width=640;max-height=360;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v3 (medium resolution)
   a=rid:3 recv max-width=640;max-height=360;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   ...same rtpmap/fmtp as above...
   a=recvonly
   a=mid:v4 (small resolution)
   a=rid:4 recv max-width=320;max-height=180;max-fps=15
   ...
   m=video 10000 RTP/SAVPF 98 99 100 101 102 103 104 105 106 107
   ...same rtpmap/fmtp as above...
   ...same rid:4 as above for mid:v5,v6,v7 (small resolution)...
   ...


   Answer:
   ...same as offer but swap send/recv...
```

11.2. Simulcast

Adding simulcast to the above example allows the mixer to selectively forward streams like an SFU rather than transcode high resolutions to lower ones. Simulcast encodings can be expressed using PTs or RIDs. Using PTs can exhaust the primary dynamic space even faster in simulcast scenarios. So RIDs are used to avoid PT exhaustion and express the encoding constraints. In the example below, 3 resolutions are offered to be sent as simulcast to a mixer/SFU.

Example 2

```
Offer:
...
m=audio ... same as from Example 1 ..
...
m=video ...same as from Example 1 ...
...same rtpmap/fmtp as above...
a=sendrecv
a=mid:v1 (max resolution)
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:5 send max-width=640;max-height=360;max-fps=15
a=rid:6 send max-width=320;max-height=180;max-fps=15
a=simulcast: send rid=1;5;6 recv rid=2
...
...same m=video sections as Example 1 for mid:v2-v7...
...

Answer:
...same as offer but swap send/recv...
```

11.3. Scalable Layers

Adding scalable layers to the above simulcast example gives the SFU further flexibility to selectively forward packets from a source that best match the bandwidth and capabilities of diverse receivers. Scalable encodings have dependencies between layers, unlike independent simulcast streams. RIDs can be used to express these dependencies using the "depend" parameter. In the example below, the highest resolution is offered to be sent as 2 scalable temporal layers (using MRST).

Example 3

```
Offer:
...
m=audio ...same as Example 1 ...
...
m=video ...same as Example 1 ...
...same rtpmap/fmtp as Example 1...
a=sendrecv
a=mid:v1 (max resolution)
a=rid:0 send max-width=1280;max-height=720;max-fps=15
a=rid:1 send max-width=1280;max-height=720;max-fps=30;depend=0
a=rid:2 recv max-width=1280;max-height=720;max-fps=30
a=rid:5 send max-width=640;max-height=360;max-fps=15
a=rid:6 send max-width=320;max-height=180;max-fps=15
a=simulcast: send rid=0;1;5;6 recv rid=2
...
...same m=video sections as Example1 for mid:v2-v7...
...

Answer:
...same as offer but swap send/recv...
```

11.4. Simulcast with Payload Types

This example shows a simulcast Offer SDP that uses rid framework to identify:

- 1 send stream at max resolution,
- 1 recv stream at max resolution,
- 1 recv stream at low resolution

and includes 2 "a=simulcast" lines to identify the simulcast streams with the Payload Types and rid-identifier respectively.

> Note: The exact rules for the usage of rid framework with simulcast is still a work in progress.

Example 4

```
Offer:
m=video 10000 RTP/AVP 97 98
a=rtpmap:97 VP8/90000
a=rtpmap:98 VP8/90000
a=fmtp:97 max-fs=3600
a=fmtp:98 max-fs=3600
a=rid:1 send pt=97;max-width=1280;max-height=720;
a=rid:2 recv pt=97;max-width=1280;max-height=720
a=rid:3 recv pt=98;max-width=320;max-height=180
a=simulcast send pt=97 recv
a=simulcast: send rid=1 recv rid=2;3
```

12. Open Issues

12.1. Name of the identifier

The name 'rid' is provisionally used and is open for further discussion.

Here are the few options that were considered while writing this draft

- CID: Constraint ID, which is a rather precise description of what we are attempting to accomplish.

- ESID: Encoded Stream ID, does not align well with taxonomy which defines Encoded Stream as before RTP packetization.

- RSID or RID: RTP Stream ID, aligns better with taxonomy but very vague.

- LID: Layer ID, aligns well for SVC with each layer in a separate stream, but not for other SVC layerings or independent simulcast which is awkward to view as layers.

- EPT or XPT: EXtended Payload Type, conveys XPT.PT usage well, but may be confused with PT, for example people may mistakenly think they can use it in other places where PT would normally be used.

13. IANA Considerations

13.1. New RTP Header Extension URI

This document defines a new extension URI in the RTP Compact Header Extensions subregistry of the Real-Time Transport Protocol (RTP) Parameters registry, according to the following data:

Extension URI:	urn:ietf:params:rtp-hdrext:rid
Description:	RTP Stream Identifier
Contact:	<mmusic@ietf.org>
Reference:	RFCXXXX

13.2. New SDES item

RFC EDITOR NOTE: Please replace RFCXXXX with the RFC number of this document.

RFC EDITOR NOTE: Please replace TBD with the assigned SDES identifier value.

This document adds the MID SDES item to the IANA "RTCP SDES item types" registry as follows:

Value:	TBD
Abbrev.:	RID
Name:	Restriction Identification
Reference:	RFCXXXX

13.3. New SDP Media-Level attribute

This document defines "rid" as SDP media-level attribute. This attribute must be registered by IANA under "Session Description Protocol (SDP) Parameters" under "att-field (media level only)".

The "rid" attribute is used to identify characteristics of RTP stream with in a RTP Session. Its format is defined in Section 10.

13.4. Registry for RID-Level Parameters

This specification creates a new IANA registry named "att-field (rid level)" within the SDP parameters registry. The rid-level parameters MUST be registered with IANA and documented under the same rules as for SDP session-level and media-level attributes as specified in [RFC4566].

Parameters for "a=rid" lines that modify the nature of encoded media MUST be of the form that the result of applying the modification to

the stream results in a stream that still complies with the other parameters that affect the media. In other words, parameters always have to restrict the definition to be a subset of what is otherwise allowable, and never expand it.

New parameter registrations are accepted according to the "Specification Required" policy of [RFC5226], provided that the specification includes the following information:

- contact name, email address, and telephone number
- parameter name (as it will appear in SDP)
- long-form parameter name in English
- whether the parameter value is subject to the charset attribute
- an explanation of the purpose of the parameter
- a specification of appropriate attribute values for this parameter

The initial set of rid-level parameter names, with definitions in Section 6 of this document, is given below:

Type	SDP Name	Reference
att-field	(rid level)	
	max-width	[RFCXXXX]
	max-height	[RFCXXXX]
	max-fps	[RFCXXXX]
	max-fs	[RFCXXXX]
	max-br	[RFCXXXX]
	max-pps	[RFCXXXX]
	depend	[RFCXXXX]

14. Security Considerations

As with most SDP parameters, a failure to provide integrity protection over the a=rid attributes provides attackers a way to modify the session in potentially unwanted ways. This could result in an implementation sending greater amounts of data than a recipient wishes to receive. In general, however, since the "a=rid" attribute can only restrict a stream to be a subset of what is otherwise allowable, modification of the value cannot result in a stream that is of higher bandwidth than would be sent to an implementation that does not support this mechanism.

15. Acknowledgements

Many thanks to review from Cullen Jennings and Magnus Westerlund.

16. References

16.1. Normative References

[I-D.ietf-avtext-rtp-grouping-taxonomy]
Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", draft-ietf-avtext-rtp-grouping-taxonomy-08 (work in progress), July 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <http://www.rfc-editor.org/info/rfc3264>.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <http://www.rfc-editor.org/info/rfc3550>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <http://www.rfc-editor.org/info/rfc5234>.

[RFC5285] Singer, D. and H. Desineni, "A General Mechanism for RTP Header Extensions", RFC 5285, DOI 10.17487/RFC5285, July 2008, <http://www.rfc-editor.org/info/rfc5285>.

16.2. Informative References

[I-D.ietf-mmusic-sdp-bundle-negotiation]
Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", draft-ietf-mmusic-sdp-bundle-negotiation-23 (work in progress), July 2015.

[I-D.ietf-mmusic-sdp-simulcast]
Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in SDP and RTP Sessions", draft-ietf-mmusic-sdp-simulcast-02 (work in progress), October 2015.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <http://www.rfc-editor.org/info/rfc5226>.

[RFC6236] Johansson, I. and K. Jung, "Negotiation of Generic Image Attributes in the Session Description Protocol (SDP)", RFC 6236, DOI 10.17487/RFC6236, May 2011, <http://www.rfc-editor.org/info/rfc6236>.

Authors' Addresses

Peter Thatcher
Google

Email: pthatcher@google.com

Mo Zanaty
Cisco Systems

Email: mzanaty@cisco.com

Suhas Nandakumar
Cisco Systems

Email: snandaku@cisco.com

Adam Roach
Mozilla

Email: adam@nostrum.com

Bo Burman
Ericsson

Email: bo.burman@ericsson.com

Byron Campen
Mozilla

Email: bcampen@mozilla.com