Internet Engineering Task Force (IETF)
Request for Comments: 8851
Updates: 4855
Category: Standards Track
ISSN: 2070-1721

A.B. Roach, Ed.
Mozilla

January 2021

RTP Payload Format Restrictions

Abstract

In this specification, we define a framework for specifying restrictions on RTP streams in the Session Description Protocol (SDP). This framework defines a new "rid" ("restriction identifier") SDP attribute to unambiguously identify the RTP streams within an RTP session and restrict the streams' payload format parameters in a codec-agnostic way beyond what is provided with the regular payload types.

This specification updates RFC 4855 to give additional guidance on choice of Format Parameter (fmtp) names and their relation to the restrictions defined by this document.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8851.

Table of Contents

1. Terminology

The terms "source RTP stream", "endpoint", "RTP session", and "RTP stream" are used as defined in [RFC7656].

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

2. Introduction

The payload type (PT) field in RTP provides a mapping between the RTP payload format and the associated SDP media description. For a given PT, the SDP rtpmap and/or fmtp attributes are used to describe the properties of the media that is carried in the RTP payload.

Recent advances in standards have given rise to rich multimedia applications requiring support for either multiple RTP streams within an RTP session [RFC8843] [RFC8853] or a large number of codecs. 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 underspecified payload format parameters

* Overloading of PTs to indicate not just codec configurations, but individual streams within an RTP session

To expand on these points: [RFC3550] assigns 7 bits for the PT in the RTP header. However, the assignment of static mapping of RTP payload

type numbers to payload formats and multiplexing of RTP with other protocols (such as the RTP Control Protocol (RTCP)) could result in a limited number of payload type numbers available for application usage. In scenarios where the number of possible RTP payload configurations exceeds the available PT space within an RTP session, there is a need for a way to represent the additional restrictions on payload configurations and effectively map an RTP stream to its corresponding restrictions. This issue is exacerbated by the increase in techniques -- such as simulcast and layered codecs -- that introduce additional streams into RTP sessions.

This specification defines a new SDP framework for restricting source RTP streams (Section 2.1.10 of [RFC7656]), along with the SDP attributes to restrict payload formats in a codec-agnostic way. This framework can be thought of as a complementary extension to the way the media format parameters are specified in SDP today, via the "a=fmtp" attribute.

The additional restrictions on individual streams are indicated with a new "a=rid" ("restriction identifier") SDP attribute. Note that the restrictions communicated via this attribute only serve to further restrict the parameters that are established on a PT format. They do not relax any restrictions imposed by other mechanisms.

This specification makes use of the RTP Stream Identifier Source Description (SDES) RTCP item defined in [RFC8852] to provide correlation between the RTP packets and their format specification in the SDP.

As described in Section 6.2.1, this mechanism achieves backwards compatibility via the normal SDP processing rules, which require unknown "a=" lines 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 restrictions requested by this mechanism.

Further, as described in Section 6 and its subsections, this mechanism achieves extensibility by: (a) having offerers include all supported restrictions in their offer, and (b) having answerers ignore "a=rid" lines that specify unknown restrictions.

3. Key Words for Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. SDP "a=rid" Media Level Attribute

This section defines new SDP media-level attribute [RFC4566], "a=rid", used to communicate a set of restrictions to be applied to an identified RTP stream. Roughly speaking, this attribute takes the following form (see Section 10 for a formal definition):

a=rid:<rid-id> <direction> [pt=<fmt-list>;]<restriction>=<value>...

An "a=rid" SDP media attribute specifies restrictions defining a unique RTP payload configuration identified via the "rid-id" field. This value binds the restriction to the RTP stream identified by its RTP Stream Identifier Source Description (SDES) item [RFC8852]. Implementations that use the "a=rid" parameter in SDP MUST support the RtpStreamId SDES item described in [RFC8852]. Such implementations MUST send that SDES item for all streams in an SDP media description ("m=") that have "a=rid" lines remaining after

applying the rules in Section 6 and its subsections.

Implementations that use the "a=rid" parameter in SDP and make use of redundancy RTP streams [RFC7656] -- e.g., RTP RTX [RFC4588] or Forward Error Correction (FEC) [RFC5109] -- for any of the source RTP streams that have "a=rid" lines remaining after applying the rules in Section 6 and its subsections MUST support the RepairedRtpStreamId SDES item described in [RFC8852] for those redundancy RTP streams. RepairedRtpStreamId MUST be used for redundancy RTP streams to which it can be applied. Use of RepairedRtpStreamId is not applicable for redundancy formats that directly associate RTP streams through shared synchronization sources (SSRCs) -- for example, [RFC8627] -- or other cases that RepairedRtpStreamId cannot support, such as referencing multiple source streams.

RepairedRtpStreamId is used to provide the binding between the redundancy RTP stream and its source RTP stream by setting the RepairedRtpStreamId value for the redundancy RTP stream to the RtpStreamId value of the source RTP stream. The redundancy RTP stream MAY (but need not) have an "a=rid" line of its own, in which case the RtpStreamId SDES item value will be different from the corresponding source RTP stream.

It is important to note that this indirection may result in the temporary inability to correctly associate source and redundancy data when the SSRC associated with the RtpStreamId or RepairedRtpStreamId is dynamically changed during the RTP session. This can be avoided if all RTP packets, source and repair, include their RtpStreamId or RepairedRtpStreamId, respectively, after the change. To maximize the probability of reception and utility of redundancy information after such a change, all the source packets referenced by the first several repair packets SHOULD include such information. It is RECOMMENDED that the number of such packets is large enough to give a high probability of actual updated association. Section 4.1.1 of [RFC8285] provides relevant guidance for RTP header extension transmission considerations. Alternatively, to avoid this issue, redundancy mechanisms that directly reference its source data may be used, such as [RFC8627].

The "direction" field identifies the direction of the RTP stream packets to which the indicated restrictions are applied. It may be either "send" or "recv". Note that these restriction directions are expressed independently of any "inactive", "sendonly", "recvonly", or "sendrecv" attributes associated with the media section. It is, for example, valid to indicate "recv" restrictions on a "sendonly" stream; those restrictions would apply if, at a future point in time, the stream were changed to "sendrecv" or "recvonly".

The optional "pt=<fmt-list>" lists one or more PT values that can be used in the associated RTP stream. If the "a=rid" attribute contains no "pt", then any of the PT values specified in the corresponding "m=" line may be used.

The list of zero or more codec-agnostic restrictions (Section 5) describes the restrictions that the corresponding RTP stream will conform to.

This framework MAY be used in combination with the "a=fmtp" SDP attribute for describing the media format parameters for a given RTP payload type. In such scenarios, the "a=rid" restrictions (Section 5) further restrict the equivalent "a=fmtp" attributes.

A given SDP media description MAY have zero or more "a=rid" lines describing various possible RTP payload configurations. A given "rid-id" MUST NOT be repeated in a given media description ("m=" section).

The "a=rid" media attribute MAY be used for any RTP-based media transport. It is not defined for other transports, although other documents may extend its semantics for such transports.

Though the restrictions specified by the "rid" restrictions follow a syntax similar to session-level and media-level parameters, they are defined independently. All "rid" restrictions MUST be registered with IANA, using the registry defined in Section 12.

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.

5. "a=rid" Restrictions

This section defines the "a=rid" restrictions that can be used to restrict the RTP payload encoding format in a codec-agnostic way. Please also see the preceding section for a description of how the "pt" parameter is used.

The following restrictions 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 height 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 frame rate for encoding, this value is used to restrict 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. This is the product of frame width and frame height, in pixels, for rectangular frames.

* *max-br*, for bitrate 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 near resolution-change boundaries.

* *max-bpp*, for maximum number of bits per pixel, calculated as an average of all samples of any given coded picture. This is expressed as a floating point value, with an allowed range of 0.0001 to 48.0. These values MUST NOT be encoded with more than four digits to the right of the decimal point.

* *depend*, to identify other streams that the stream depends on. The value is a comma-separated list of rid-ids. These rid-ids identify RTP streams that this stream depends on in order to allow for proper interpretation. The mechanism defined in this document allows for such dependencies to be expressed only when the streams are in the same media section.

All the restrictions are optional and subject to negotiation based on the SDP offer/answer rules described in Section 6.

This list is intended to be an initial set of restrictions. Future documents may define additional restrictions; see Section 12.2. While this document does not define restrictions for audio codecs or any media types other than video, there is no reason such restrictions 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 "a=rid" restrictions defined in this section.

6. SDP Offer/Answer Procedures

This section describes the SDP offer/answer procedures [RFC3264] when using this framework.

Note that "rid-id" values are only required to be unique within a media section ("m=" line); they do not necessarily need to be unique within an entire RTP session. In traditional usage, each media section is sent on its own unique 5-tuple (that is: combination of sending address, sending port, receiving address, receiving port, and transport protocol), which provides an unambiguous scope. Similarly, when using BUNDLE [RFC8843], Media Identification (MID) values associate RTP streams uniquely to a single media description. When restriction identifier (RID) is used with the BUNDLE mechanism, streams will be associated with both MID and RID SDES items.

6.1. Generating the Initial SDP Offer

For each RTP 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 these steps:

1. It MUST generate a "rid-id" that is unique within a media description.

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

3. It MAY include a listing of SDP media formats (usually corresponding to RTP payload types) allowed to appear in the RTP stream. Any payload type chosen MUST be a valid payload type for the media section (that is, it must be listed on the "m=" line). The order of the listed formats is significant; the alternatives are listed from (left) most preferred to (right) least preferred. When using RID, this preference overrides the normal codec preference as expressed by format type ordering on the "m=" line, using regular SDP rules.

4. The offerer then chooses zero or more "a=rid" restrictions (Section 5) to be applied to the RTP stream and adds them to the "a=rid" line.

5. If the offerer wishes the answerer to have the ability to specify a restriction but does not wish to set a value itself, it includes the name of the restriction 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 "a=rid" restrictions will further restrict the equivalent "a=fmtp" parameters for the given payload type for the specified RTP stream.

If a given codec would require an "a=fmtp" line when used without "a=rid", then the offer MUST include a valid corresponding "a=fmtp" line even when using "a=rid".

6.2. Answerer Processing the SDP Offer

6.2.1. "a=rid"-Unaware Answerer

If the receiver doesn’t support the framework defined in this specification, the entire "a=rid" line is ignored following the standard offer/answer rules [RFC3264].

Section 6.1 requires the offer to include a valid "a=fmtp" line for any media formats that otherwise require it (in other words, the "a=rid" line cannot be used to replace "a=fmtp" configuration). As a result, ignoring the "a=rid" line is always guaranteed to result in a valid session description.

6.2.2. "a=rid"-Aware Answerer

If the answerer supports the "a=rid" attribute, the following verification steps are executed, in order, for each "a=rid" line in a received offer:

1. The answerer ensures that the "a=rid" line is syntactically well formed. In the case of a syntax error, the "a=rid" line is discarded.

2. The answerer extracts the rid-id from the "a=rid" line and verifies its uniqueness within a media section. In the case of a duplicate, the entire "a=rid" line, and all "a=rid" lines with rid-ids that duplicate this line, are discarded and MUST NOT be included in the SDP answer.

3. If the "a=rid" line contains a "pt=", the list of payload types is verified against the list of valid payload types for the media section (that is, those listed on the "m=" line). Any PT missing from the "m=" line is discarded from the set of values in the "pt=". If no values are left in the "pt=" parameter after this processing, then the "a=rid" line is discarded.

4. If the "direction" field is "recv", the answerer ensures that the specified "a=rid" restrictions are supported. In the case of an unsupported restriction, the "a=rid" line is discarded.

5. If the "depend" restriction is included, the answerer MUST make sure that the listed rid-ids unambiguously match the rid-ids in the media description. Any "depend" "a=rid" lines that do not are discarded.

6. The answerer verifies that the restrictions are consistent with at least one of the codecs to be used with the RTP stream. If the "a=rid" line contains a "pt=", it contains the list of such codecs; otherwise, the list of such codecs is taken from the associated "m=" line. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec

properties for all codecs, then the "a=rid" line is discarded.

Note that the answerer does not need to understand every restriction present in a "send" line: if a stream sender restricts the stream in a way that the receiver does not understand, this causes no issues with interoperability.

6.3. Generating the SDP Answer

Having performed verification of the SDP offer as described in Section 6.2.2, the answerer shall perform the following steps to generate the SDP answer.

For each "a=rid" line that has not been discarded by previous processing:

1. The value of the "direction" field is reversed: "send" is changed to "recv", and "recv" is changed to "send".

2. The answerer MAY choose to modify specific "a=rid" restriction values in the answer SDP. In such a case, the modified value MUST be more restrictive than the ones specified in the offer. The answer MUST NOT include any restrictions that were not present in the offer.

3. The answerer MUST NOT modify the "rid-id" present in the offer.

4. If the "a=rid" line contains a "pt=", the answerer is allowed to discard one or more media formats from a given "a=rid" line. If the answerer chooses to discard all the media formats from an "a=rid" line, the answerer MUST discard the entire "a=rid" line. If the offer did not contain a "pt=" for a given "a=rid" line, then the answer MUST NOT contain a "pt=" in the corresponding line.

5. In cases where the answerer is unable to support the payload configuration specified in a given "a=rid" line with a direction of "recv" in the offer, the answerer MUST discard the corresponding "a=rid" line. This includes situations in which the answerer does not understand one or more of the restrictions in an "a=rid" line with a direction of "recv".

Note: In the case that the answerer uses different PT values to represent a codec than the offerer did, the "a=rid" values in the answer use the PT values that are present in its answer.

6.4. Offerer Processing of the SDP Answer

The offerer SHALL follow these steps when processing the answer:

1. The offerer matches the "a=rid" line in the answer to the "a=rid" line in the offer using the "rid-id". If no matching line can be located in the offer, the "a=rid" line is ignored.

2. If the answer contains any restrictions that were not present in the offer, then the offerer SHALL discard the "a=rid" line.

3. If the restrictions have been changed between the offer and the answer, the offerer MUST ensure that the modifications are more restrictive than they were in the original offer and that they can be supported; if not, the offerer SHALL discard the "a=rid" line.

4. If the "a=rid" line in the answer contains a "pt=" but the offer did not, the offerer SHALL discard the "a=rid" line.

5. If the "a=rid" line in the answer contains a "pt=" and the offer did as well, the offerer verifies that the list of payload types is a subset of those sent in the corresponding "a=rid" line in the offer. Note that this matching must be performed semantically rather than on literal PT values, as the remote end may not be using symmetric PTs. For the purpose of this comparison: for each PT listed on the "a=rid" line in the answer, the offerer looks up the corresponding "a=rtpmap" and "a=fmtp" lines in the answer. It then searches the list of "pt=" values indicated in the offer and attempts to find one with an equivalent set of "a=rtpmap" and "a=fmtp" lines in the offer. If all PTs in the answer can be matched, then the "pt=" values pass validation; otherwise, it fails. If this validation fails, the offerer SHALL discard the "a=rid" line. Note that this semantic comparison necessarily requires an understanding of the meaning of codec parameters, rather than a rote byte-wise comparison of their values.

6. If the "a=rid" line contains a "pt=", the offerer verifies that the attribute values provided in the "a=rid" attributes are consistent with the corresponding codecs and their other parameters. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec properties, then the offerer SHALL discard the "a=rid" line.

7. The offerer verifies that the restrictions are consistent with at least one of the codecs to be used with the RTP stream. If the "a=rid" line contains a "pt=", it contains the list of such codecs; otherwise, the list of such codecs is taken from the associated "m=" line. See Section 8 for more detail. If the "a=rid" restrictions are incompatible with the other codec properties for all codecs, then the offerer SHALL discard the "a=rid" line.

Any "a=rid" line present in the offer that was not matched by step 1 above has been discarded by the answerer and does not form part of the negotiated restrictions on an RTP stream. The offerer MAY still apply any restrictions it indicated in an "a=rid" line with a direction field of "send", but it is not required to do so.

It is important to note that there are several ways in which an offer can contain a media section with "a=rid" lines, although the corresponding media section in the response does not. This includes situations in which the answerer does not support "a=rid" at all or does not support the indicated restrictions. Under such circumstances, the offerer MUST be prepared to receive a media stream to which no restrictions have been applied.

6.5. Modifying the Session

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

7. Use with Declarative SDP

This document does not define the use of a RID in declarative SDP. If concrete use cases for RID in declarative SDP use are identified in the future, we expect that additional specifications will address such use.

8. Interaction with Other Techniques

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

* Codec-specific configuration set via format parameters ("a=fmtp") -- for example, the H.264 "max-fs" format parameter [RFC6184]

* Size restrictions imposed by the "a=imageattr" attribute [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 intersection. 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 6.2.2, this will result in the corresponding "a=rid" line being ignored by the recipient.

In 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 intersection. 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=imageattr"), its only recourse is to discard the "a=rid" line altogether, as described in Section 6.3. If it instead attempts to restrict the stream beyond what is allowed by other mechanisms, then the offerer will ignore the corresponding "a=rid" line, as described in Section 6.4.

The following subsections demonstrate these interactions using commonly used video codecs. These descriptions are illustrative of the interaction principles outlined above and are not normative.

8.1. Interaction with VP8 Format Parameters

[RFC7741] defines two format parameters for the VP8 codec. Both correspond to restrictions on receiver capabilities and never indicate sending restrictions.

8.1.1. max-fr - Maximum Frame Rate

The VP8 "max-fr" format parameter corresponds to the "max-fps" restriction defined in this specification. If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fps" parameter, then the sent stream will conform to the smaller of the two values.

8.1.2. max-fs - Maximum Frame Size, in VP8 Macroblocks

The VP8 "max-fs" format parameter corresponds to the "max-fs" restriction defined in this document, by way of a conversion factor of the number of pixels per macroblock (typically 256). If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fs" parameter, then the sent stream will conform to the smaller of the two values; that is, the number of pixels per frame will not exceed:

```
min(rid_max_fs, fmtp_max_fs * macroblock_size)
```

This fmtp parameter also has bearing on the max-height and max-width

parameters. Section 6.1 of [RFC7741] requires that the width and height of the frame in macroblocks be less than int(sqrt(fmtp_max_fs * 8)). Accordingly, the maximum width of a transmitted stream will be limited to:

```
min(rid_max_width, int(sqrt(fmtp_max_fs * 8)) * macroblock_width)
```

Similarly, the stream's height will be limited to:

```
min(rid_max_height, int(sqrt(fmtp_max_fs * 8)) * macroblock_height)
```

8.2. Interaction with H.264 Format Parameters

[RFC6184] defines format parameters for the H.264 video codec. The majority of these parameters do not correspond to codec-independent restrictions:

* deint-buf-cap
* in-band-parameter-sets
* level-asymmetry-allowed
* max-rcmd-nalu-size
* max-cpb
* max-dpb
* packetization-mode
* redundant-pic-cap
* sar-supported
* sar-understood
* sprop-deint-buf-req
* sprop-init-buf-time
* sprop-interleaving-depth
* sprop-level-parameter-sets
* sprop-max-don-diff
* sprop-parameter-sets
* use-level-src-parameter-sets

Note that the max-cpb and max-dpb format parameters for H.264 correspond to restrictions on the stream, but they are specific to the way the H.264 codec operates, and do not have codec-independent equivalents.

The [RFC6184] codec format parameters covered in the following sections correspond to restrictions on receiver capabilities and never indicate sending restrictions.

8.2.1. profile-level-id and max-recv-level - Negotiated Subprofile

These parameters include a "level" indicator, which acts as an index into Table A-1 of [H264]. This table contains a number of parameters, several of which correspond to the restrictions defined in this document. [RFC6184] also defines format parameters for the

H.264 codec that may increase the maximum values indicated by the negotiated level. The following sections describe the interaction between these parameters and the restrictions defined by this document. In all cases, the H.264 parameters being discussed are the maximum of those indicated by [H264] Table A-1 and those indicated in the corresponding "a=fmtp" line.

8.2.2. max-br / MaxBR - Maximum Video Bitrate

The H.264 "MaxBR" parameter (and its equivalent "max-br" format parameter) corresponds to the "max-bps" restriction defined in this specification, by way of a conversion factor of 1000 or 1200; see [RFC6184] for details regarding which factor gets used under differing circumstances.

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fps" parameter, then the sent stream will conform to the smaller of the two values -- that is:

```
min(rid_max_br, h264_MaxBR * conversion_factor)
```

8.2.3. max-fs / MaxFS - Maximum Frame Size, in H.264 Macroblocks

The H.264 "MaxFs" parameter (and its equivalent "max-fs" format parameter) corresponds roughly to the "max-fs" restriction defined in this document, by way of a conversion factor of 256 (the number of pixels per macroblock).

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-fs" parameter, then the sent stream will conform to the smaller of the two values -- that is:

```
min(rid_max_fs, h264_MaxFs * 256)
```

8.2.4. max-mbps / MaxMBPS - Maximum Macroblock Processing Rate

The H.264 "MaxMBPS" parameter (and its equivalent "max-mbps" format parameter) corresponds roughly to the "max-pps" restriction defined in this document, by way of a conversion factor of 256 (the number of pixels per macroblock).

If an RTP sender is generating a stream using a format defined with this format parameter, and the sending restrictions defined via "a=rid" include a "max-pps" parameter, then the sent stream will conform to the smaller of the two values -- that is:

```
min(rid_max_pps, h264_MaxMBPS * 256)
```

8.2.5. max-smbps - Maximum Decoded Picture Buffer

The H.264 "max-smbps" format parameter operates the same way as the "max-mbps" format parameter, under the hypothetical assumption that all macroblocks are static macroblocks. It is handled by applying the conversion factor described in Section 8.1 of [RFC6184], and the result of this conversion is applied as described in Section 8.2.4.

8.3. Redundancy Formats and Payload Type Restrictions

Section 4 specifies that redundancy formats using redundancy RTP streams bind the redundancy RTP stream to the source RTP stream with either the RepairedRtpStreamId SDES item or other mechanisms. However, there exist redundancy RTP payload formats that result in the redundancy being included in the source RTP stream. An example of this is "RTP Payload for Redundant Audio Data" [RFC2198], which

encapsulates one source stream with one or more redundancy streams in the same RTP payload. Formats defining the source and redundancy encodings as regular RTP payload types require some consideration for how the "a=rid" restrictions are defined. The "a=rid" line "pt=" parameter can be used to indicate whether the redundancy RTP payload type and/or the individual source RTP payload type(s) are part of the restriction.

Example (SDP excerpt):

```
m=audio 49200 RTP/AVP 97 98 99 100 101 102
a=mid:foo
a=rtpmap:97 G711/8000
a=rtpmap:98 LPC/8000
a=rtpmap:99 OPUS/48000/1
a=rtpmap:100 RED/8000/1
a=rtpmap:101 CN/8000
a=rtpmap:102 telephone-event/8000
a=fmtp:99 useinbandfec=1; usedtx=0
a=fmtp:100 97/98
a=fmtp:102 0-15
a=ptime:20
a=maxptime:40
a=rid:5 send pt=99,102;max-br=64000
a=rid:6 send pt=100,97,101,102
```

The RID with ID=6 restricts the payload types for this RID to 100 (the redundancy format), 97 (G.711), 101 (Comfort Noise), and 102 (dual-tone multi-frequency (DTMF) tones). This means that RID 6 can either contain the Redundant Audio Data (RED) format, encapsulating encodings of the source media stream using payload type 97 and 98, 97 without RED encapsulation, Comfort noise, or DTMF tones. Payload type 98 is not included in the RID, and can thus not be sent except as redundancy information in RED encapsulation. If 97 were to be excluded from the pt parameter, it would instead mean that payload types 97 and 98 are only allowed via RED encapsulation.

9. Format Parameters for Future Payloads

Registrations of future RTP payload format specifications that define media types that have parameters matching the RID restrictions specified in this memo SHOULD name those parameters in a manner that matches the names of those RID restrictions and SHOULD explicitly state what media-type parameters are restricted by what RID restrictions.

10. Formal Grammar

This section gives a formal Augmented Backus-Naur Form (ABNF) [RFC5234] grammar, with the case-sensitive extensions described in [RFC7405], for each of the new media and "a=rid" attributes defined in this document.

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

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

alpha-numeric     = < as defined in [RFC4566] >

rid-dir           = %s"send" / %s"recv"

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

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

```
rid-fmt-list      = %s"pt=" fmt *( "," fmt )

fmt               = < as defined in [RFC4566] >

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

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

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

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

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

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

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

rid-bpp-param     = %s"max-bpp" [ "=" float-param-val ]

rid-depend-param  = %s"depend=" rid-list

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

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

int-param-val     = 1*DIGIT

float-param-val   = 1*DIGIT "." 1*DIGIT

param-val         = *(%x20-3A / %x3C-7E)
                  ; Any printable character except semicolon
```

11. SDP Examples

Note: See [RFC8853] for examples of RID used in simulcast scenarios.

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.

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

inserted.

The offer for this scenario is shown below.

```
...
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=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
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
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
...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
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
...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
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
...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
a=extmap 1 urn:ietf:params:rtp-hdrext:sdes:rtp-stream-id
...same rtpmap/fmtp as above...
...same rid:4 as above for mid:v5,v6,v7 (small resolution)...
...
```

11.2. Scalable Layers

Adding scalable layers to a session within a multiparty conference gives a selective forwarding unit (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" restriction. In the example below, the highest resolution is offered to be sent as 2 scalable temporal layers (using Multiple RTP Streams on a Single Media Transport (MRST)). See [RFC8853] for additional detail about simulcast usage.

```
Offer:
...
m=audio ...same as previous example ...
...
m=video ...same as previous example ...
...same rtpmap/fmtp as previous example ...
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 previous example for mid:v2-v7...
...
```

12. IANA Considerations

This specification updates [RFC4855] to give additional guidance on choice of Format Parameter (fmtp) names and their relation to RID restrictions.

12.1. New SDP Media-Level Attribute

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

The "rid" attribute is used to identify the properties of an RTP stream within an RTP session. Its format is defined in Section 10.

The formal registration information for this attribute follows.

Contact name, email address, and telephone number
: IETF MMUSIC Working Group
: mmusic@ietf.org
: +1 510 492 4080

Attribute name (as it will appear in SDP)
: rid

Long-form attribute name in English
: Restriction Identifier

Type of attribute (session level, media level, or both)
: Media Level

Whether the attribute value is subject to the charset attribute
: The attribute is not dependent on charset.

A one-paragraph explanation of the purpose of the attribute
   The "rid" SDP attribute is used to unambiguously identify the RTP streams within an RTP session and restrict the streams' payload format parameters in a codec-agnostic way beyond what is provided with the regular payload types.

A specification of appropriate attribute values for this attribute
   Valid values are defined by the ABNF in RFC 8851

Multiplexing (Mux) Category
   SPECIAL

12.2. Registry for RID-Level Parameters

This specification creates a new IANA registry named "RID Attribute Parameters" within the SDP parameters registry. The "a=rid" restrictions 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, restrictions always have to restrict the definition to be a subset of what is otherwise allowable, and never expand it.

New restriction registrations are accepted according to the "Specification Required" policy of [RFC8126]. The registration MUST contain the RID parameter name and a reference to the corresponding specification. The specification itself must contain the following information (not all of which appears in the registry):

* restriction name (as it will appear in SDP)

* an explanation of the purpose of the restriction

* a specification of appropriate attribute values for this restriction

* an ABNF definition of the restriction

The initial set of "a=rid" restriction names, with definitions in Section 5 of this document, is given below:

RID Parameter Name	Reference
pt	RFC 8851
max-width	RFC 8851
max-height	RFC 8851
max-fps	RFC 8851
max-fs	RFC 8851
max-br	RFC 8851
max-pps	RFC 8851
max-bpp	RFC 8851
depend	RFC 8851

Table 1: "a=rid" restriction names

It is conceivable that a future document will want to define RID-level restrictions that contain string values. These extensions need to take care to conform to the ABNF defined for rid-param-other. In particular, this means that such extensions will need to define escaping mechanisms if they want to allow semicolons, unprintable characters, or byte values greater than 127 in the string.

13. Security Considerations

As with most SDP parameters, a failure to provide integrity protection over the "a=rid" attributes gives 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.

The actual identifiers used for RIDs are expected to be opaque. As such, they are not expected to contain information that would be sensitive, were it observed by third parties.

14. References

14.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://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, <https://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, <https://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, <https://www.rfc-editor.org/info/rfc4566>.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

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

[RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", RFC 7405, DOI 10.17487/RFC7405, December 2014, <https://www.rfc-editor.org/info/rfc7405>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8852] Roach, A.B., Nandakumar, S., and P. Thatcher, "RTP Stream Identifier Source Description (SDES)", RFC 8852, DOI 10.17487/RFC8852, January 2021, <https://www.rfc-editor.org/info/rfc8852>.

14.2. Informative References

[H264] International Telecommunication Union, "Advanced video coding for generic audiovisual services", ITU-T Recommendation H.264, June 2019, <https://www.itu.int/rec/T-REC-H.264>.

[RFC2198] Perkins, C., Kouvelas, I., Hodson, O., Hardman, V., Handley, M., Bolot, J.C., Vega-Garcia, A., and S. Fosse-Parisis, "RTP Payload for Redundant Audio Data", RFC 2198, DOI 10.17487/RFC2198, September 1997, <https://www.rfc-editor.org/info/rfc2198>.

[RFC4588] Rey, J., Leon, D., Miyazaki, A., Varsa, V., and R. Hakenberg, "RTP Retransmission Payload Format", RFC 4588, DOI 10.17487/RFC4588, July 2006, <https://www.rfc-editor.org/info/rfc4588>.

[RFC5109] Li, A., Ed., "RTP Payload Format for Generic Forward Error Correction", RFC 5109, DOI 10.17487/RFC5109, December 2007, <https://www.rfc-editor.org/info/rfc5109>.

[RFC6184] Wang, Y.-K., Even, R., Kristensen, T., and R. Jesup, "RTP Payload Format for H.264 Video", RFC 6184, DOI 10.17487/RFC6184, May 2011, <https://www.rfc-editor.org/info/rfc6184>.

[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, <https://www.rfc-editor.org/info/rfc6236>.

[RFC7656] Lennox, J., Gross, K., Nandakumar, S., Salgueiro, G., and B. Burman, Ed., "A Taxonomy of Semantics and Mechanisms for Real-Time Transport Protocol (RTP) Sources", RFC 7656, DOI 10.17487/RFC7656, November 2015, <https://www.rfc-editor.org/info/rfc7656>.

[RFC7741] Westin, P., Lundin, H., Glover, M., Uberti, J., and F. Galligan, "RTP Payload Format for VP8 Video", RFC 7741, DOI 10.17487/RFC7741, March 2016, <https://www.rfc-editor.org/info/rfc7741>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8285] Singer, D., Desineni, H., and R. Even, Ed., "A General Mechanism for RTP Header Extensions", RFC 8285, DOI 10.17487/RFC8285, October 2017, <https://www.rfc-editor.org/info/rfc8285>.

[RFC8627] Zanaty, M., Singh, V., Begen, A., and G. Mandyam, "RTP Payload Format for Flexible Forward Error Correction (FEC)", RFC 8627, DOI 10.17487/RFC8627, July 2019, <https://www.rfc-editor.org/info/rfc8627>.

[RFC8843] Holmberg, C., Alvestrand, H., and C. Jennings, "Negotiating Media Multiplexing Using the Session Description Protocol (SDP)", RFC 8843,

DOI 10.17487/RFC8843, January 2021,
<https://www.rfc-editor.org/info/rfc8843>.

[RFC8853] Burman, B., Westerlund, M., Nandakumar, S., and M. Zanaty, "Using Simulcast in Session Description Protocol (SDP) and RTP Sessions", RFC 8853, DOI 10.17487/RFC8853, January 2021, <https://www.rfc-editor.org/info/rfc8853>.

Acknowledgements

Many thanks to Cullen Jennings, Magnus Westerlund, and Paul Kyzivat for reviewing. Thanks to Colin Perkins for input on future payload type handling.

Contributors

The following individuals have contributed significant text to this document.

Peter Thatcher
Google

Email: pthatcher@google.com

Mo Zanaty
Cisco Systems

Email: mzanaty@cisco.com

Suhas Nandakumar
Cisco Systems

Email: snandaku@cisco.com

Bo Burman
Ericsson

Email: bo.burman@ericsson.com

Byron Campen
Mozilla

Email: bcampen@mozilla.com

Author’s Address

Adam Roach (editor)
Mozilla

Email: adam@nostrum.com