Payload Working Group   J. Uberti
Internet-Draft   S. Holmer
Intended status: Standards Track   M. Flodman
Expires: April 21, 2016   Google
                                                               J. Lennox
                                                                 D. Hong
                                                                   Vidyo
                                                        October 19, 2015

RTP Payload Format for VP9 Video
draft-ietf-payload-vp9-01

Abstract

This memo describes an RTP payload format for the VP9 video codec. The payload format has wide applicability, as it supports applications from low bit-rate peer-to-peer usage, to high bit-rate video conferences. It includes provisions for temporal and spatial scalability.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This memo describes an RTP payload specification applicable to the transmission of video streams encoded using the VP9 video codec [I-D.grange-vp9-bitstream]. The format described in this document can be used both in peer-to-peer and video conferencing applications.

TODO: VP9 description. Please see [I-D.grange-vp9-bitstream].

2. Conventions, Definitions and Acronyms

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].

TODO: Cite terminology from [I-D.grange-vp9-bitstream].

3. Media Format Description

The VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame.

VP9 also allows a reference frame to be resampled and used as a reference for another frame of a different resolution. This allows internal resolution changes without requiring the use of key frames.

These features together enable an encoder to implement various forms of coarse-grained scalability, including temporal, spatial and quality scalability modes, as well as combinations of these, without the need for explicit scalable coding tools.

Temporal layers define different frame rates of video; spatial and quality layers define different and possibly dependent representations of a single input frame. Spatial layers allow a frame to be encoded at different resolutions, whereas quality layers allow a frame to be encoded at the same resolution but at different qualities (and thus with different amounts of coding error). VP9 supports quality layers as spatial layers without any resolution changes; hereinafter, the term "spatial layer" is used to represent both spatial and quality layers.

This payload format specification defines how such temporal and spatial scalability layers can be described and communicated.

Temporal and spatial scalability layers are associated with non-negative integer IDs. The lowest layer of either type has an ID of 0.

Layers are designed (and MUST be encoded) such that if any layer, and all higher layers, are removed from the bitstream along any of the two dimensions, the remaining bitstream is still correctly decodable.

For terminology, this document uses the term "layer frame" to refer to a single encoded VP9 frame for a particular resolution/quality, and "super frame" to refer to all the representations (layer frames) at a single instant in time. A super frame thus consists of one or more layer frames, encoding different spatial layers.

Within a super frame, a layer frame with spatial layer ID equal to S, where S > 0, can depend on a layer frame of the same super frame with a lower spatial layer ID. This "inter-layer" dependency can result in additional coding gain compared to the case where only traditional "inter-picture" dependency is used, where a frame depends on

previously coded frame in time. For simplicity, this payload format assumes that, within a super frame and if inter-layer dependency is used, a spatial layer S frame can only depend on spatial layer S-1 frame when S > 0. Additionally, if inter-picture dependency is used, spatial layer S frame is assumed to only depend on previously coded spatial layer S frame.

TODO: Describe how simulcast can be supported?

Given above simplifications for inter-layer and inter-picture dependencies, a flag (the D bit described below) is used to indicate whether a spatial layer S frame depends on spatial layer S-1 frame. Given the D bit, a receiver only needs to additionally know the inter-picture dependency structure for a given spatial layer frame in order to determine its decodability. Two modes of describing the inter-picture dependency structure are possible: "flexible mode" and "non-flexible mode". An encoder can only switch between the two on the very first packet of a key frame with temporal layer ID equal to 0.

In flexible mode, each packet can contain up to 3 reference indices, which identify all frames referenced by the frame transmitted in the current packet for inter-picture prediction. This (along with the D bit) enables a receiver to identify if a frame is decodable or not and helps it understand the temporal layer structure. Since this is signaled in each packet it makes it possible to have very flexible temporal layer hierarchies and patterns which are changing dynamically.

In non-flexible mode, the inter-picture dependency (the reference indices) of a group of frames (GOF) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet MUST have an index to refer to one of the described frames in the GOF, from which the frames referenced by the frame transmitted in the current packet for inter-picture prediction can be identified.

The SS data can also be used to specify the resolution of each spatial layer present in the VP9 stream for both flexible and non-flexible modes.

4. Payload Format

This section describes how the encoded VP9 bitstream is encapsulated in RTP. To handle network losses usage of RTP/AVPF [RFC4585] is RECOMMENDED. All integer fields in the specifications are encoded as unsigned integers in network octet order.

4.1. RTP Header Usage

The general RTP payload format for VP9 is depicted below.

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V=2|P|X|  CC   |M|     PT      |       sequence number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           timestamp                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           synchronization source (SSRC) identifier            |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            contributing source (CSRC) identifiers             |
|                             ....                              |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|            VP9 payload descriptor (integer #octets)           |
:                                                               :
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               : VP9 pyld hdr  |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
+                                                               |
:                   Bytes 2..N of VP9 payload                   :
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               :    OPTIONAL RTP padding       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The VP9 payload descriptor and VP9 payload header will be described in Section 4.2 and Section 4.3. OPTIONAL RTP padding MUST NOT be included unless the P bit is set. The figure specifically shows the format for the first packet in a frame. Subsequent packets will not contain the VP9 payload header, and will have later octets in the frame payload.

Figure 1

Marker bit (M): MUST be set to 1 for the final packet of the highest spatial layer frame (the final packet of the super frame), and 0 otherwise. Unless spatial scalability is in use for this super frame, this will have the same value as the E bit described below. Note this bit MUST be set to 1 for the target spatial layer frame if a stream is being rewritten to remove higher spatial layers.

Payload Type (TP): In line with the policy in Section 3 of [RFC3551], applications using the VP9 RTP payload profile MUST assign a dynamic payload type number to be used in each RTP

session and provide a mechanism to indicate the mapping. See Section 6.2 for the mechanism to be used with the Session Description Protocol (SDP) [RFC4566].

Timestamp: The RTP timestamp indicates the time when the input frame was sampled, at a clock rate of 90 kHz. If the input frame is encoded with multiple layer frames, all of the layer frames of the super frame MUST have the same timestamp.

The remaining RTP Fixed Header Fields (V, P, X, CC, sequence number, SSRC and CSRC identifiers) are used as specified in Section 5.1 of [RFC3550].

4.2. VP9 Payload Description

In flexible mode (with the F bit below set to 1), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

```
      0 1 2 3 4 5 6 7
     +-+-+-+-+-+-+-+-+
     |I|P|L|F|B|E|V|-| (REQUIRED)
     +-+-+-+-+-+-+-+-+
I:   |M| PICTURE ID  | (REQUIRED)
     +-+-+-+-+-+-+-+-+
M:   | EXTENDED PID  | (RECOMMENDED)
     +-+-+-+-+-+-+-+-+
L:   |  T  |U|  S  |D| (CONDITIONALLY RECOMMENDED)
     +-+-+-+-+-+-+-+-+                             -\
P,F: | P_DIFF      |N| (CONDITIONALLY REQUIRED)    - up to 3 times
     +-+-+-+-+-+-+-+-+                             -/
V:   | SS            |
     | ..            |
     +-+-+-+-+-+-+-+-+
```

Figure 2

In non-flexible mode (with the F bit below set to 0), The first octets after the RTP header are the VP9 payload descriptor, with the following structure.

```
      0 1 2 3 4 5 6 7
     +-+-+-+-+-+-+-+-+
     |I|P|L|F|B|E|V|-| (REQUIRED)
     +-+-+-+-+-+-+-+-+
I:   |M| PICTURE ID  | (RECOMMENDED)
     +-+-+-+-+-+-+-+-+
M:   | EXTENDED PID  | (RECOMMENDED)
     +-+-+-+-+-+-+-+-+
L:   |  T  |U|  S  |D| (CONDITIONALLY RECOMMENDED)
     +-+-+-+-+-+-+-+-+
     |   TL0PICIDX   | (CONDITIONALLY REQUIRED)
     +-+-+-+-+-+-+-+-+
V:   | SS            |
     | ..            |
     +-+-+-+-+-+-+-+-+
```

Figure 3

I: Picture ID (PID) present. When set to one, the OPTIONAL PID MUST be present after the mandatory first octet and specified as below. Otherwise, PID MUST NOT be present.

P: Inter-picture predicted layer frame. When set to zero, the layer frame does not utilize inter-picture prediction. In this case, up-switching to current spatial layer's frame is possible from directly lower spatial layer frame. P SHOULD also be set to zero when encoding a layer synchronization frame in response to an LRR [I-D.lennox-avtext-lrr]. When P is set to zero, the T bit (described below) MUST also be set to 0 (if present).

L: Layer indices present. When set to one, the one or two octets following the mandatory first octet and the PID (if present) is as described by "Layer indices" below. If the F bit (described below) is set to 1 (indicating flexible mode), then only one octet is present for the layer indices. Otherwise if the F bit is set to 0 (indicating non-flexible mode), then two octets are present for the layer indices.

F: Flexible mode. F set to one indicates flexible mode and if the P bit is also set to one, then the octets following the mandatory first octet, the PID, and layer indices (if present) are as described by "Reference indices" below. This MUST only be set to 1 if the I bit is also set to one; if the I bit is set to zero,

then this MUST also be set to zero and ignored by receivers. The value of this F bit CAN ONLY CHANGE on the very first packet of a key picture. This is a packet with the P bit equal to zero, S or D bit (described below) equal to zero, and B bit (described below) equal to 1.

B: Start of a layer frame. MUST be set to 1 if the first payload octet of the RTP packet is the beginning of a new VP9 layer frame, and MUST NOT be 1 otherwise. Note that this layer frame might not be the very first layer frame of a super frame.

E: End of a layer frame. MUST be set to 1 for the final RTP packet of a VP9 layer frame, and 0 otherwise. This enables a decoder to finish decoding the layer frame, where it otherwise may need to wait for the next packet to explicitly know that the layer frame is complete. Note that, if spatial scalability is in use, more layer frames from the same super frame may follow; see the description of the M bit above.

V: Scalability structure (SS) data present. When set to one, the OPTIONAL SS data MUST be present in the payload descriptor. Otherwise, the SS data MUST NOT be present.

-: Bit reserved for future use. MUST be set to zero and MUST be ignored by the receiver.

The mandatory first octet is followed by the extension data fields that are enabled:

M: The most significant bit of the first octet is an extension flag. The field MUST be present if the I bit is equal to one. If set, the PID field MUST contain 15 bits; otherwise, it MUST contain 7 bits. See PID below.

Picture ID (PID): Picture ID represented in 7 or 15 bits, depending on the M bit. This is a running index of the pictures. The field MUST be present if the I bit is equal to one. If M is set to zero, 7 bits carry the PID; else if M is set to one, 15 bits carry the PID in network byte order. The sender may choose between a 7- or 15-bit index. The PID SHOULD start on a random number, and MUST wrap after reaching the maximum ID. The receiver MUST NOT assume that the number of bits in PID stay the same through the session.

In the non-flexible mode (when the F bit is set to 0), this PID is used as an index to the GOF specified in the SS data bleow. In this mode, the PID of the key frame corresponds to the very first specified frame in the GOF. Then subsequent PIDs are mapped to

subsequently specified frames in the GOF (modulo N_G, specified in the SS data below), respectively.

Layer indices: This information is optional but recommended whenever encoding with layers. For both flexible and non-flexible modes, one octet is used to specify a layer frame's temporal layer ID (T) and spatial layer ID (S) as shown both in Figure 2 and Figure 3. Additionally, a bit (U) is used to indicate that the current frame is a "switching up point" frame. Another bit (D) is used to indicate whether inter-layer prediction is used for the current layer frame.

In the non-flexible mode (when the F bit is set to 0), another octet is used to represent temporal layer 0 index (TL0PICIDX), as depicted in Figure 3. The TL0PICIDX is present so that all minimally required frames - the base temporal layer frames - can be tracked.

The T and S fields indicate the temporal and spatial layers and can help middleboxes and and endpoints quickly identify which layer a packet belongs to.

T: The temporal layer ID of current frame. In the case of non-flexible mode, if PID is mapped to a frame in a specified GOF, then the value of T MUST match the corresponding T value of the mapped frame in the GOF.

U: Switching up point. If this bit is set to 1 for the current frame with temporal layer ID equal to T, then "switch up" to a higher frame rate is possible as subsequent higher temporal layer frames will not depend on any frame before the current frame (in coding time) with temporal layer ID greater than T.

S: The spatial layer ID of current frame. Note that frames with spatial layer S > 0 may be dependent on decoded spatial layer S-1 frame within the same super frame.

D: Inter-layer dependency used. MUST be set to one if current spatial layer S frame depends on spatial layer S-1 frame of the same super frame. MUST only be set to zero if current spatial layer S frame does not depend on spatial layer S-1 frame of the same super frame. For the base layer frame with S equal to 0, this D bit MUST be set to zero.

TL0PICIDX: 8 bits temporal layer zero index. TL0PICIDX is only present in the non-flexible mode (F = 0). This is a running index for the temporal base layer frames, i.e., the frames with T set to 0. If T is larger than 0, TL0PICIDX indicates which

temporal base layer frame the current frame depends on. TL0PICIDX MUST be incremented when T is equal to 0. The index SHOULD start on a random number, and MUST restart at 0 after reaching the maximum number 255.

Reference indices: When P and F are both set to one, indicating a non-key frame in flexible mode, then at least one reference index has to be specified as below. Additional reference indices (total of up to 3 reference indices are allowed) may be specified using the N bit below. When either P or F is set to zero, then no reference index is specified.

P_DIFF: The reference index (in 7 bits) specified as the relative PID from the current frame. For example, when P_DIFF=3 on a packet containing the frame with PID 112 means that the frame refers back to the frame with PID 109. This calculation is done modulo the size of the PID field, i.e., either 7 or 15 bits.

N: 1 if there is additional P_DIFF following the current P_DIFF.

4.2.1. Scalability Structure (SS):

The scalability structure (SS) data describes the resolution of each layer frame within a super frame as well as the inter-picture dependencies for a group of frames (GOF). If the VP9 payload descriptor's "V" bit is set, the SS data is present in the position indicated in Figure 2 and Figure 3.

```
     +-+-+-+-+-+-+-+-+
V:   | N_S |Y|G|-|-|-|
     +-+-+-+-+-+-+-+-+              -\
Y:   |     WIDTH     | (OPTIONAL)    .
     +               +               .
     |               | (OPTIONAL)    .
     +-+-+-+-+-+-+-+-+               . - N_S + 1 times
     |     HEIGHT    | (OPTIONAL)    .
     +               +               .
     |               | (OPTIONAL)    .
     +-+-+-+-+-+-+-+-+              -/            -\
G:   |      N_G      | (OPTIONAL)
     +-+-+-+-+-+-+-+-+                            -\
N_G: |  T  |U| R |-|-| (OPTIONAL)                 .
     +-+-+-+-+-+-+-+-+              -\            . - N_G times
     |    P_DIFF     | (OPTIONAL)    . - R times  .
     +-+-+-+-+-+-+-+-+              -/            -/
```

Figure 4

N_S: N_S + 1 indicates the number of spatial layers present in the VP9 stream.

Y: Each spatial layer's frame resolution present. When set to one, the OPTIONAL WIDTH (2 octets) and HEIGHT (2 octets) MUST be present for each layer frame. Otherwise, the resolution MUST NOT be present.

G: GOF description present flag.

-: Bit reserved for future use. MUST be set to zero and MUST be ignored by the receiver.

N_G: N_G indicates the number of frames in a GOF. If N_G is greater than 0, then the SS data allows the inter-picture dependency structure of the VP9 stream to be pre-declared, rather than indicating it on the fly with every packet. If N_G is greater than 0, then for N_G pictures in the GOF, each frame's temporal layer ID (T), switch up point (U), and the R reference indices (P_DIFFs) are specified.

The very first frame specified in the GOF MUST have T set to 0.

G set to 0 or N_G set to 0 indicates that either there is only one temporal layer or no fixed inter-picture dependency information is present going forward in the bitstream.

Note that for a given super frame, all layer frames follow the same inter-picture dependency structure. However, the frame rate of each spatial layer can be different from each other and this can be controlled with the use of the D bit described above. The specified dependency structure in the SS data MUST be for the highest frame rate layer.

In a scalable stream sent with a fixed pattern, the SS data SHOULD be included in the first packet of every key frame. This is a packet with P bit equal to zero, S or D bit equal to zero, and B bit equal to 1. The SS data MUST only be changed on the frame that corresponds to the very first frame specified in the previous SS data's GOF (if the previous SS data's N_G was greater than 0).

4.3. VP9 Payload Header

TODO: need to describe VP9 payload header.

4.4. Frame Fragmentation

VP9 frames are fragmented into packets, in RTP sequence number order, beginning with a packet with the B bit set, and ending with a packet with the RTP marker bit M set. There is no mechanism for finer-grained access to parts of a VP9 frame.

4.5. Examples of VP9 RTP Stream

TODO

5. Using VP9 with RPSI and SLI Feedback

The VP9 payload descriptor defined in Section 4.2 above contains an optional PictureID parameter. One use of this parameter is to enable use of the reference picture selection index (RPSI) and slice loss indication (SLI) RTCP feedback messages, both defined in [RFC4585].

5.1. RPSI

TODO: Update to indicate which frame within the picture.

The reference picture selection index is a payload-specific feedback message defined within the RTCP-based feedback format. The RPSI message is generated by a receiver and can be used in two ways. Either it can signal a preferred reference picture when a loss has been detected by the decoder -- preferably then a reference that the decoder knows is perfect -- or, it can be used as positive feedback information to acknowledge correct decoding of certain reference pictures. The positive feedback method is useful for VP9 used for point to point (unicast) communication. The use of RPSI for VP9 is preferably combined with a special update pattern of the codec's two special reference frames -- the golden frame and the altref frame -- in which they are updated in an alternating leapfrog fashion. When a receiver has received and correctly decoded a golden or altref frame, and that frame had a PictureID in the payload descriptor, the receiver can acknowledge this simply by sending an RPSI message back to the sender. The message body (i.e., the "native RPSI bit string" in [RFC4585]) is simply the PictureID of the received frame.

5.2. SLI

TODO: Update to indicate which frame within the picture.

The slice loss indication is another payload-specific feedback message defined within the RTCP-based feedback format. The SLI message is generated by the receiver when a loss or corruption is

detected in a frame. The format of the SLI message is as follows [RFC4585]:

```
 0                   1                   2                   3
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         First           |        Number             | PictureID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5

Here, First is the macroblock address (in scan order) of the first lost block and Number is the number of lost blocks, as defined in [RFC4585]. PictureID is the six least significant bits of the codec-specific picture identifier in which the loss or corruption has occurred. For VP9, this codec-specific identifier is naturally the PictureID of the current frame, as read from the payload descriptor. If the payload descriptor of the current frame does not have a PictureID, the receiver MAY send the last received PictureID+1 in the SLI message. The receiver MAY set the First parameter to 0, and the Number parameter to the total number of macroblocks per frame, even though only part of the frame is corrupted. When the sender receives an SLI message, it can make use of the knowledge from the latest received RPSI message. Knowing that the last golden or altref frame was successfully received, it can encode the next frame with reference to that established reference.

5.3. Example

TODO: this example is copied from the VP8 payload format specification, and has not been updated for VP9. It may be incorrect.

The use of RPSI and SLI is best illustrated in an example. In this example, the encoder may not update the altref frame until the last sent golden frame has been acknowledged with an RPSI message. If an update is not received within some time, a new golden frame update is sent instead. Once the new golden frame is established and acknowledged, the same rule applies when updating the altref frame.

Event	Sender	Receiver	Established reference
1000	Send golden frame PictureID = 0		
		Receive and decode	

```
   |       |                   | golden frame            |             |
   |       |                   |                         |             |
   | 1001  |                   | Send RPSI(0)            |             |
   |       |                   |                         |             |
   | 1002  | Receive RPSI(0)   |                         | golden      |
   |       |                   |                         |             |
   | ...   | (sending regular  |                         |             |
   |       | frames)           |                         |             |
   |       |                   |                         |             |
   | 1100  | Send altref frame |                         |             |
   |       | PictureID = 100   |                         |             |
   |       |                   |                         |             |
   |       |                   | Altref corrupted or     | golden      |
   |       |                   | lost                    |             |
   |       |                   |                         |             |
   | 1101  |                   | Send SLI(100)           | golden      |
   |       |                   |                         |             |
   | 1102  | Receive SLI(100)  |                         |             |
   |       |                   |                         |             |
   | 1103  | Send frame with   |                         |             |
   |       | reference to      |                         |             |
   |       | golden            |                         |             |
   |       |                   |                         |             |
   |       |                   | Receive and decode      | golden      |
   |       |                   | frame (decoder state    |             |
   |       |                   | restored)               |             |
   |       |                   |                         |             |
   | ...   | (sending regular  |                         |             |
   |       | frames)           |                         |             |
   |       |                   |                         |             |
   | 1200  | Send altref frame |                         |             |
   |       | PictureID = 200   |                         |             |
   |       |                   |                         |             |
   |       |                   | Receive and decode      | golden      |
   |       |                   | altref frame            |             |
   |       |                   |                         |             |
   | 1201  |                   | Send RPSI(200)          |             |
   |       |                   |                         |             |
   | 1202  | Receive RPSI(200) |                         | altref      |
   |       |                   |                         |             |
   | ...   | (sending regular  |                         |             |
   |       | frames)           |                         |             |
   |       |                   |                         |             |
   | 1300  | Send golden frame |                         |             |
   |       | PictureID = 300   |                         |             |
   |       |                   |                         |             |
   |       |                   | Receive and decode      | altref      |
   |       |                   | golden frame            |             |
```

1301		Send RPSI(300)	altref
1302	RPSI lost		
1400	Send golden frame PictureID = 400		
		Receive and decode golden frame	altref
1401		Send RPSI(400)	
1402	Receive RPSI(400)		golden

Table 1: Example signaling between sender and receiver

Note that the scheme is robust to loss of the feedback messages. If the RPSI is lost, the sender will try to update the golden (or altref) again after a while, without releasing the established reference. Also, if an SLI is lost, the receiver can keep sending SLI messages at any interval allowed by the RTCP sending timing restrictions as specified in [RFC4585], as long as the picture is corrupted.

6. Payload Format Parameters

This payload format has two optional parameters.

6.1. Media Type Definition

This registration is done using the template defined in [RFC6838] and following [RFC4855].

Type name: video

Subtype name: VP9

Required parameters: None.

Optional parameters:
These parameters are used to signal the capabilities of a receiver implementation. If the implementation is willing to receive media, both parameters MUST be provided. These parameters MUST NOT be used for any other purpose.

max-fr: The value of max-fr is an integer indicating the maximum frame rate in units of frames per second that the decoder is capable of decoding.

max-fs: The value of max-fs is an integer indicating the maximum frame size in units of macroblocks that the decoder is capable of decoding.

The decoder is capable of decoding this frame size as long as the width and height of the frame in macroblocks are less than int(sqrt(max-fs * 8)) - for instance, a max-fs of 1200 (capable of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).

Encoding considerations:
This media type is framed in RTP and contains binary data; see Section 4.8 of [RFC6838].

Security considerations: See Section 7 of RFC xxxx.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Interoperability considerations: None.

Published specification: VP9 bitstream format [I-D.grange-vp9-bitstream] and RFC XXXX.
[RFC Editor: Upon publication as an RFC, please replace "XXXX" with the number assigned to this document and remove this note.]

Applications which use this media type:
For example: Video over IP, video conferencing.

Fragment identifier considerations: N/A.

Additional information: None.

Person & email address to contact for further information:
TODO [Pick a contact]

Intended usage: COMMON

Restrictions on usage:
This media type depends on RTP framing, and hence is only defined for transfer via RTP [RFC3550].

Author: TODO [Pick a contact]

Change controller:

IETF Payload Working Group delegated from the IESG.

6.2. SDP Parameters

The receiver MUST ignore any fmtp parameter unspecified in this memo.

6.2.1. Mapping of Media Subtype Parameters to SDP

The media type video/VP9 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.

- The encoding name in the "a=rtpmap" line of SDP MUST be VP9 (the media subtype).

- The clock rate in the "a=rtpmap" line MUST be 90000.

- The parameters "max-fs", and "max-fr", MUST be included in the "a=fmtp" line of SDP if SDP is used to declare receiver capabilities. These parameters are expressed as a media subtype string, in the form of a semicolon separated list of parameter=value pairs.

6.2.1.1. Example

An example of media representation in SDP is as follows:

```
m=video 49170 RTP/AVPF 98
a=rtpmap:98 VP9/90000
a=fmtp:98 max-fr=30; max-fs=3600;
```

6.2.2. Offer/Answer Considerations

TODO: Update this for VP9

7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile such as RTP/AVP [RFC3551], RTP/AVPF [RFC4585], RTP/SAVP [RFC3711], or RTP/SAVPF [RFC5124]. SAVPF [RFC5124]. However, as "Securing the RTP Protocol Framework: Why RTP Does Not Mandate a Single Media Security Solution" [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity and source authenticity for RTP in general. This responsibility lays on anyone

using RTP in an application. They can find guidance on available security mechanisms in Options for Securing RTP Sessions [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this security consideration section discusses the security impacting properties of the payload format itself.

This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing, and thus are unlikely to pose a denial-of-service threat due to the receipt of pathological data. Nor does the RTP payload format contain any active content.

8. Congestion Control

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550], and with any applicable RTP profile; e.g., RFC 3551 [RFC3551]. The congestion control mechanism can, in a real-time encoding scenario, adapt the transmission rate by instructing the encoder to encode at a certain target rate. Media aware network elements MAY use the information in the VP9 payload descriptor in Section 4.2 to identify non-reference frames and discard them in order to reduce network congestion. Note that discarding of non-reference frames cannot be done if the stream is encrypted (because the non-reference marker is encrypted).

9. IANA Considerations

The IANA is requested to register the following values:
- Media type registration as described in Section 6.1.

10. References

10.1. Normative References

[I-D.grange-vp9-bitstream]
Grange, A. and H. Alvestrand, "A VP9 Bitstream Overview", draft-grange-vp9-bitstream-00 (work in progress), February 2013.

[I-D.lennox-avtext-lrr]
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", draft-lennox-avtext-lrr-00 (work in progress), March 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>.

[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>.

[RFC4585] Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey, "Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, DOI 10.17487/RFC4585, July 2006, <http://www.rfc-editor.org/info/rfc4585>.

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

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <http://www.rfc-editor.org/info/rfc6838>.

10.2. Informative References

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <http://www.rfc-editor.org/info/rfc3551>.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <http://www.rfc-editor.org/info/rfc3711>.

[RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <http://www.rfc-editor.org/info/rfc5124>.

[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", RFC 7201, DOI 10.17487/RFC7201, April 2014, <http://www.rfc-editor.org/info/rfc7201>.

[RFC7202] Perkins, C. and M. Westerlund, "Securing the RTP Framework: Why RTP Does Not Mandate a Single Media Security Solution", RFC 7202, DOI 10.17487/RFC7202, April 2014, <http://www.rfc-editor.org/info/rfc7202>.

Authors' Addresses

Justin Uberti
Google, Inc.
747 6th Street South
Kirkland, WA 98033
USA

Email: justin@uberti.name

Stefan Holmer
Google, Inc.
Kungsbron 2
Stockholm 111 22
Sweden

Email: holmer@google.com

Magnus Flodman
Google, Inc.
Kungsbron 2
Stockholm 111 22
Sweden

Email: mflodman@google.com

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Danny Hong
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: danny@vidyo.com