

AVTCore Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 August 2021

J. Uberti  
S. Holmer  
M. Flodman  
D. Hong  
Google  
J. Lennox  
8x8 / Jitsi  
2 February 2021

## **RTP Payload Format for VP9 Video draft-ietf-payload-vp9-11**

### 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

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 August 2021.

### Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Simplified BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Conventions, Definitions and Acronyms . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Media Format Description . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Payload Format . . . . .	<a href="#">5</a>
<a href="#">4.1.</a>	RTP Header Usage . . . . .	<a href="#">5</a>
<a href="#">4.2.</a>	VP9 Payload Descriptor . . . . .	<a href="#">6</a>
<a href="#">4.2.1.</a>	Scalability Structure (SS): . . . . .	<a href="#">11</a>
<a href="#">4.3.</a>	Frame Fragmentation . . . . .	<a href="#">12</a>
<a href="#">4.4.</a>	Scalable encoding considerations . . . . .	<a href="#">13</a>
<a href="#">4.5.</a>	Examples of VP9 RTP Stream . . . . .	<a href="#">13</a>
<a href="#">4.5.1.</a>	Reference picture use for scalable structure . . . . .	<a href="#">13</a>
<a href="#">5.</a>	Feedback Messages and Header Extensions . . . . .	<a href="#">14</a>
<a href="#">5.1.</a>	Reference Picture Selection Indication (RPSI) . . . . .	<a href="#">14</a>
<a href="#">5.2.</a>	Full Intra Request (FIR) . . . . .	<a href="#">15</a>
<a href="#">5.3.</a>	Layer Refresh Request (LRR) . . . . .	<a href="#">15</a>
<a href="#">6.</a>	Payload Format Parameters . . . . .	<a href="#">16</a>
<a href="#">6.1.</a>	Media Type Definition . . . . .	<a href="#">16</a>
<a href="#">6.2.</a>	SDP Parameters . . . . .	<a href="#">18</a>
<a href="#">6.2.1.</a>	Mapping of Media Subtype Parameters to SDP . . . . .	<a href="#">19</a>
<a href="#">6.2.2.</a>	Offer/Answer Considerations . . . . .	<a href="#">19</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">20</a>
<a href="#">8.</a>	Congestion Control . . . . .	<a href="#">20</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">21</a>
<a href="#">10.</a>	Acknowledgments . . . . .	<a href="#">21</a>
<a href="#">11.</a>	References . . . . .	<a href="#">21</a>
<a href="#">11.1.</a>	Normative References . . . . .	<a href="#">21</a>
<a href="#">11.2.</a>	Informative References . . . . .	<a href="#">22</a>
	Authors' Addresses . . . . .	<a href="#">23</a>

## [1.](#) Introduction

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

The VP9 video codec was developed by Google, and is the successor to its earlier VP8 [[RFC6386](#)] codec. Above the compression improvements and other general enhancements above VP8, VP9 is also designed in a way that allows spatially-scalable video encoding.



## **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](#)].

## **3. Media Format Description**

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

VP9 also allows a frame to use another frame of a different resolution as a reference frame. (Specifically, a frame may use any references whose width and height are between 1/16th that of the current frame and twice that of the current frame, inclusive.) 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, and is sometimes referred to as the "base" temporal or spatial layer.

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



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

Within a picture, a frame with spatial layer ID equal to  $SID$ , where  $SID > 0$ , can depend on a frame of the same picture 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 picture and if inter-layer dependency is used, a spatial layer  $SID$  frame can depend only on the immediately previous spatial layer  $SID-1$  frame, when  $S > 0$ . Additionally, if inter-picture dependency is used, a spatial layer  $SID$  frame is assumed to only depend on a previously coded spatial layer  $SID$  frame.

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  $SID$  frame depends on the spatial layer  $SID-1$  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 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 Picture Group (PG) MUST be pre-specified as part of the scalability structure (SS) data. In this mode, each packet has an index to refer to one of the described pictures in the PG, from which the pictures referenced by the picture transmitted in the current packet for inter-picture prediction can be identified.



(Editor's Note: A "Picture Group", as used in this document, is not the same thing as a the term "Group of Pictures" as it is traditionally used in video coding, i.e. to mean an independently-decodable run of pictures beginning with a keyframe. Suggestions for better terminology are welcome.)

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.

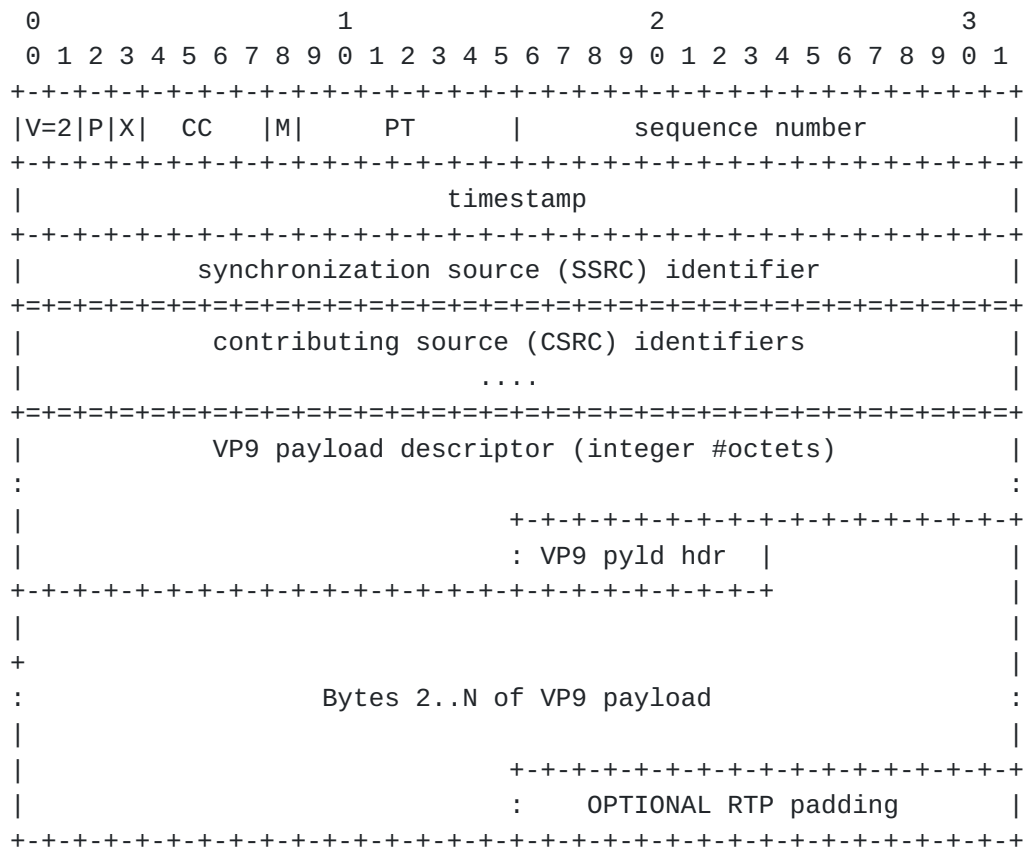






Figure 1

The VP9 payload descriptor will be described in [Section 4.2](#); the VP9 payload header is described in [\[VP9-BITSTREAM\]](#). 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.

Marker bit (M): MUST be set to 1 for the final packet of the highest spatial layer frame (the final packet of the picture), and 0 otherwise. Unless spatial scalability is in use for this picture, 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 (PT): 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) [\[RFC8866\]](#).

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

If a frame has the VP9 show\_frame field set to 0 (i.e., it is meant only to populate a reference buffer, without being output) its timestamp MAY alternately be set to be the same as the subsequent frame with show\_frame equal to 1. (This will be convenient for playing out pre-encoded content packaged with VP9 "superframes", which typically bundle show\_frame==0 frames with a subsequent show\_frame==1 frame.) Every frame with show\_frame==1, however, MUST have a unique timestamp modulo the 2<sup>32</sup> wrap of the field.

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 Descriptor**

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.



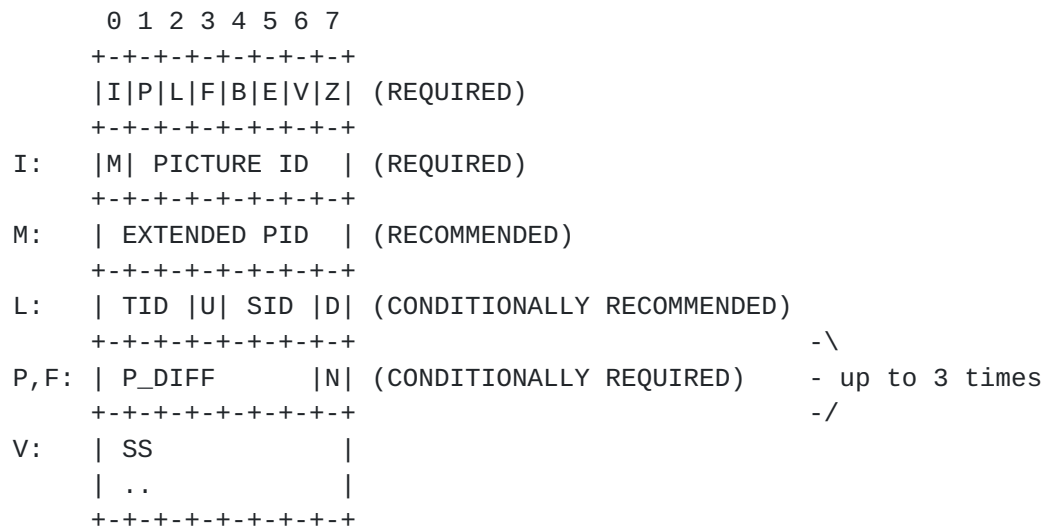


Figure 2

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.

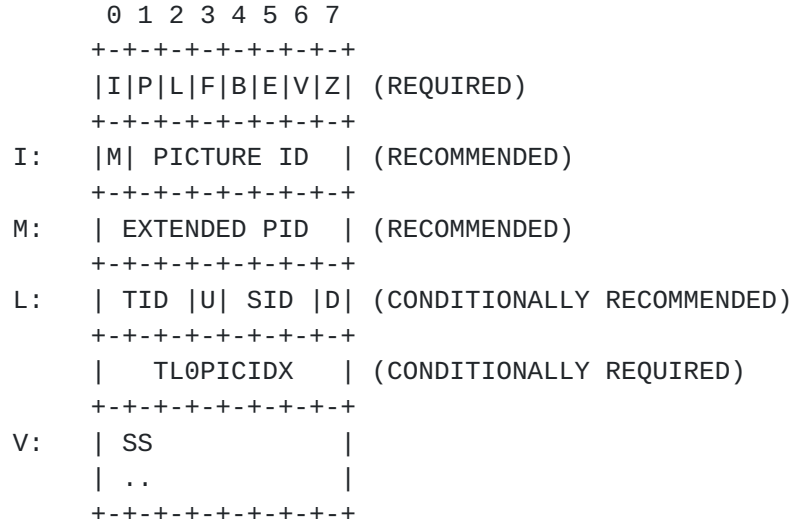


Figure 3

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. If the SS field was present in the stream's most recent start of a keyframe (i.e., non-flexible scalability mode is in use), then the PID MUST also be present in every packet.



- P: Inter-picture predicted frame. When set to zero, the frame does not utilize inter-picture prediction. In this case, up-switching to a 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.ietf-avtext-lrr](#)] message (see [Section 5.3](#)). When P is set to zero, the TID field (described below) MUST also be set to 0 (if present). Note that the P bit does not forbid intra-picture, inter-layer prediction from earlier frames of the same picture, if any.
- 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 MUST only change on the first packet of a key picture. A key picture is a picture whose base spatial layer frame is a key frame, and which thus completely resets the encoder state. This packet will have its P bit equal to zero, SID or D bit (described below) equal to zero, and B bit (described below) equal to 1.
- B: Start of a frame. MUST be set to 1 if the first payload octet of the RTP packet is the beginning of a new VP9 frame, and MUST NOT be 1 otherwise. Note that this frame might not be the first frame of a picture.
- E: End of a frame. MUST be set to 1 for the final RTP packet of a VP9 frame, and 0 otherwise. This enables a decoder to finish decoding the frame, where it otherwise may need to wait for the next packet to explicitly know that the frame is complete. Note that, if spatial scalability is in use, more frames from the same picture 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.



Z: Not a reference frame for upper spatial layers. If set to 1, indicates that frames with higher spatial layers SID+1 of the current and following pictures do not depend on the current spatial layer SID frame. This enables a decoder which is targeting a higher spatial layer to know that it can safely discard this packet's frame without processing it, without having to wait for the "D" bit in the higher-layer frame (see below).

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 picture group (PG) specified in the SS data below. In this mode, the PID of the key frame corresponds to the first specified frame in the PG. Then subsequent PIDs are mapped to subsequently specified frames in the PG (modulo N\_G, specified in the SS data below), respectively.

All frames of the same picture **MUST** have the same PID value.

Frames (and their corresponding pictures) with the VP9 `show_frame` field equal to 0 **MUST** have distinct PID values from subsequent pictures with `show_frame` equal to 1. Thus, a Picture as defined in this specification is different than a VP9 Superframe.

All frames of the same picture **MUST** have the same value for `show_frame`.

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 (TID) and spatial layer ID (SID) 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 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 TID and SID fields indicate the temporal and spatial layers and can help middleboxes and endpoints quickly identify which layer a packet belongs to.

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

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

SID: The spatial layer ID of current frame. Note that frames with spatial layer SDI > 0 may be dependent on decoded spatial layer SID-1 frame within the same picture. Different frames of the same picture MUST have distinct spatial layer IDs, and frames' spatial layers MUST appear in increasing order within the frame.

D: Inter-layer dependency used. MUST be set to one if current spatial layer SID frame depends on spatial layer SID-1 frame of the same picture. MUST only be set to zero if current spatial layer SID frame does not depend on spatial layer SID-1 frame of the same picture. For the base layer frame (with SID 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 pictures, i.e., the pictures with TID set to 0. If TID is larger than 0, TL0PICIDX indicates which temporal base layer picture the current picture depends on. TL0PICIDX MUST be incremented when TID 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 picture. For example, when P\_DIFF=3 on a packet containing the picture with PID 112 means that the picture refers back to the picture 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 frame within a picture as well as the inter-picture dependencies for a picture group (PG). 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: | TID |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: PG 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 pictures in a Picture Group (PG). 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 PG, each picture's temporal layer ID (TID), switch up point (U), and the R reference indices (P\_DIFFs) are specified.

The first picture specified in the PG MUST have TID 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 picture, all 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, SID or D bit equal to zero, and B bit equal to 1. The SS data MUST only be changed on the picture that corresponds to the first picture specified in the previous SS data's PG (if the previous SS data's N\_G was greater than 0).

#### **4.3. 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 E bit set. There is no mechanism for finer-grained access to parts of a VP9 frame.



#### **4.4. Scalable encoding considerations**

In addition to the use of reference frames, VP9 has several additional forms of inter-frame dependencies, largely involving probability tables for the entropy and tree encoders. In VP9 syntax, the syntax element "error\_resilient\_mode" resets this additional inter-frame data, allowing a frame's syntax to be decoded independently.

Due to the requirements of scalable streams, a VP9 encoder producing a scalable stream needs to ensure that a frame does not depend on a previous frame (of the same or a previous picture) that can legitimately be removed from the stream. Thus, a frame that follows a removable frame (in full decode order) **MUST** be encoded with "error\_resilient\_mode" set to true.

For spatially-scalable streams, this means that "error\_resilient\_mode" needs to be turned on for the base spatial layer; it can however be turned off for higher spatial layers, assuming they are sent with inter-layer dependency (i.e. with the "D" bit set). For streams that are only temporally-scalable without spatial scalability, "error\_resilient\_mode" can additionally be turned off for any picture that immediately follows a temporal layer 0 frame.

#### **4.5. Examples of VP9 RTP Stream**

##### **4.5.1. Reference picture use for scalable structure**

As discussed in [Section 3](#), the VP9 codec can maintain up to eight reference frames, of which up to three can be referenced or updated by any new frame. This section illustrates one way that a scalable structure (with three spatial layers and three temporal layers) can be constructed using these reference frames.





Temporal	Spatial	References	Updates
0	0	0	0
0	1	0,1	1
0	2	1,2	2
2	0	0	6
2	1	1,6	7
2	2	2,7	-
1	0	0	3
1	1	1,3	4
1	2	2,4	5
2	0	3	6
2	1	4,6	7
2	2	5,7	-

Table 1: Example scalability structure

This structure is constructed such that the "U" bit can always be set.

## 5. Feedback Messages and Header Extensions

### 5.1. Reference Picture Selection Indication (RPSI)

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.

Note: because all frames of the same picture must have the same inter-picture reference structure, there is no need for a message to specify which frame is being selected.

### 5.2. Full Intra Request (FIR)

The Full Intra Request (FIR) [RFC5104] RTCP feedback message allows a receiver to request a full state refresh of an encoded stream.

Upon receipt of an FIR request, a VP9 sender MUST send a picture with a keyframe for its spatial layer 0 layer frame, and then send frames without inter-picture prediction (P=0) for any higher layer frames.

### 5.3. Layer Refresh Request (LRR)

The Layer Refresh Request [I-D.ietf-avtext-ldrr] allows a receiver to request a single layer of a spatially or temporally encoded stream to be refreshed, without necessarily affecting the stream's other layers.

```

+-----+-----+
|0|1|2|3|4|5|6|7|0|1|2|3|4|5|6|7|
+-----+-----+
|  RES  | TID | RES  | SID |
+-----+-----+

```

Figure 5

Figure 5 shows the format of LRR's layer index fields for VP9 streams. The two "RES" fields MUST be set to 0 on transmission and ignored on reception. See [Section 4.2](#) for details on the TID and SID fields.

Identification of a layer refresh frame can be derived from the reference IDs of each frame by backtracking the dependency chain until reaching a point where only decodable frames are being referenced. Therefore it's recommended for both the flexible and the non-flexible mode that, when upgrade frames are being encoded in response to a LRR, those packets should contain layer indices and the reference fields so that the decoder or an MCU can make this derivation.



Example:

LRR {1,0}, {2,1} is sent by an MCU when it is currently relaying {1,0} to a receiver and which wants to upgrade to {2,1}. In response the encoder should encode the next frames in layers {1,1} and {2,1} by only referring to frames in {1,0}, or {0,0}.

In the non-flexible mode, periodic upgrade frames can be defined by the layer structure of the SS, thus periodic upgrade frames can be automatically identified by the picture ID.

## **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  $\text{int}(\text{sqrt}(\text{max-fs} * 8))$  - for instance, a max-fs of 1200 (capable of supporting 640x480 resolution) will support widths and heights up to 1552 pixels (97 macroblocks).



profile-id: The value of profile-id is an integer indicating the default coding profile, the subset of coding tools that may have been used to generate the stream or that the receiver supports). Table 2 lists all of the profiles defined in section 7.2 of [\[VP9-BITSTREAM\]](#) and the corresponding integer values to be used.

If no profile-id is present, Profile 0 MUST be inferred.

Informative note: See Table 3 for capabilities of coding profiles defined in section 7.2 of [\[VP9-BITSTREAM\]](#).

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 [\[VP9-BITSTREAM\]](#) 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:

Jonathan Lennox <[jonathan.lennox@8x8.com](mailto:jonathan.lennox@8x8.com)>

Intended usage:

COMMON





**Restrictions on usage:**

This media type depends on RTP framing, and hence is only defined for transfer via RTP [[RFC3550](#)].

**Author:**

Jonathan Lennox <[jonathan.lennox@8x8.com](mailto:jonathan.lennox@8x8.com)>

**Change controller:**

IETF AVTCore Working Group delegated from the IESG.

Profile	profile-id
0	0
1	1
2	2
3	3

Table 2: Table 1.  
Table of profile-id  
integer values  
representing the VP9  
profile corresponding  
to the set of coding  
tools supported.

Profile	Bit Depth	SRGB Colorspace	Chroma Subsampling
0	8	No	YUV 4:2:0
1	8	Yes	YUV 4:2:0, 4:4:0 or 4:4:4
2	10 or 12	No	YUV 4:2:0
3	10 or 12	Yes	YUV 4:2:0, 4:4:0 or 4:4:4

Table 3: Table 2. Table of profile capabilities.

## 6.2. SDP Parameters

The receiver MUST ignore any fmp 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) [[RFC8866](#)] 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.
- \* The OPTIONAL parameter profile-id, when present, SHOULD be included in the "a=fmtp" line of SDP. This parameter is expressed as a media subtype string, in the form of a parameter=value pair. When the parameter is not present, a value of 0 MUST be used for profile-id.

#### **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; profile-id=0;
```

### **6.2.2. Offer/Answer Considerations**

When VP9 is offered over RTP using SDP in an Offer/Answer model [[RFC3264](#)] for negotiation for unicast usage, the following limitations and rules apply:

- \* The parameter identifying a media format configuration for VP9 is profile-id. This media format configuration parameter MUST be used symmetrically; that is, the answerer MUST either maintain all configuration parameters or remove the media format (payload type) completely if one or more of the parameter values are not supported.



- \* To simplify the handling and matching of these configurations, the same RTP payload type number used in the offer SHOULD also be used in the answer, as specified in [\[RFC3264\]](#). An answer MUST NOT contain the payload type number used in the offer unless the configuration is exactly the same as in the offer.

## **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 media type registration "video/vp9" as specified in [Section 6.1](#). The media type is also requested to be added to the IANA registry for "RTP Payload Format MIME types" [<http://www.iana.org/assignments/rtp-parameters>](http://www.iana.org/assignments/rtp-parameters).

## **10. Acknowledgments**

Alex Eleftheriadis, Yuki Ito, Won Kap Jang, Sergio Garcia Murillo, Roi Sasson, Timothy Terriberry, Emircan Uysaler, and Thomas Volkert commented on the development of this document and provided helpful comments and feedback.

## **11. References**

### **11.1. Normative References**

- [I-D.ietf-avtext-lrr]  
Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", Work in Progress, Internet-Draft, [draft-ietf-avtext-lrr-07](#), 2 July 2017, [<http://www.ietf.org/internet-drafts/draft-ietf-avtext-lrr-07.txt>](http://www.ietf.org/internet-drafts/draft-ietf-avtext-lrr-07.txt).
- [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>](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>](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>](https://www.rfc-editor.org/info/rfc3550).
- [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, [<https://www.rfc-editor.org/info/rfc4585>](https://www.rfc-editor.org/info/rfc4585).





- [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>>.
- [RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", [RFC 5104](#), DOI 10.17487/RFC5104, February 2008, <<https://www.rfc-editor.org/info/rfc5104>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 6838](#), DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", [RFC 8866](#), DOI 10.17487/RFC8866, January 2021, <<https://www.rfc-editor.org/info/rfc8866>>.
- [VP9-BITSTREAM]  
Grange, A., de Rivaz, P., and J. Hunt, "VP9 Bitstream & Decoding Process Specification", Version 0.6, 31 March 2016, <<https://storage.googleapis.com/downloads.webmproject.org/docs/vp9/vp9-bitstream-specification-v0.6-20160331-draft.pdf>>.

## 11.2. 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, <<https://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, <<https://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, <<https://www.rfc-editor.org/info/rfc5124>>.
- [RFC6386] Bankoski, J., Koleszar, J., Quillio, L., Salonen, J., Wilkins, P., and Y. Xu, "VP8 Data Format and Decoding Guide", [RFC 6386](#), DOI 10.17487/RFC6386, November 2011, <<https://www.rfc-editor.org/info/rfc6386>>.



[RFC7201] Westerlund, M. and C. Perkins, "Options for Securing RTP Sessions", [RFC 7201](#), DOI 10.17487/RFC7201, April 2014, <<https://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, <<https://www.rfc-editor.org/info/rfc7202>>.

#### Authors' Addresses

Justin Uberti  
Google, Inc.  
747 6th Street South  
Kirkland, WA 98033  
United States of America

Email: [justin@uberti.name](mailto:justin@uberti.name)

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

Email: [holmer@google.com](mailto:holmer@google.com)

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

Email: [mflodman@google.com](mailto:mflodman@google.com)

Danny Hong  
Google, Inc.  
1585 Charleston Road  
Mountain View, CA 94043  
United States of America

Email: [dannyhong@google.com](mailto:dannyhong@google.com)



Jonathan Lennox  
8x8, Inc. / Jitsi  
1350 Broadway  
New York, NY 10018  
United States of America

Email: [jonathan.lennox@8x8.com](mailto:jonathan.lennox@8x8.com)