

Internet Engineering Task Force (IETF)  
Request for Comments: 9628  
Category: Standards Track  
ISSN: 2070-1721

J. Uberti  
OpenAI  
S. Holmer  
M. Flodman  
D. Hong  
Google  
J. Lennox  
8x8 / Jitsi  
March 2025

## RTP Payload Format for VP9 Video

### Abstract

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

### 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/rfc9628>.

### Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

### Table of Contents

1. Introduction
2. Conventions
3. Media Format Description
4. Payload Format
  - 4.1. RTP Header Usage
  - 4.2. VP9 Payload Descriptor
    - 4.2.1. Scalability Structure (SS)
  - 4.3. Frame Fragmentation
  - 4.4. Scalable Encoding Considerations
  - 4.5. Example of a VP9 RTP Stream

- 4.5.1. Reference Picture Use for Scalable Structure
- 5. Feedback Messages and Header Extensions
  - 5.1. Reference Picture Selection Indication (RPSI)
  - 5.2. Full Intra Request (FIR)
  - 5.3. Layer Refresh Request (LRR)
- 6. Payload Format Parameters
  - 6.1. SDP Parameters
    - 6.1.1. Mapping of Media Subtype Parameters to SDP
    - 6.1.2. Offer/Answer Considerations
- 7. Media Type Definition
- 8. Security Considerations
- 9. Congestion Control
- 10. IANA Considerations
- 11. References
  - 11.1. Normative References
  - 11.2. Informative References
- Acknowledgments
- Authors' Addresses

## 1. Introduction

This document describes an RTP [RFC3550] 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 to VP8, VP9 is also designed in a way that allows spatially scalable video encoding.

## 2. Conventions

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.

## 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 keyframes.

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 zero 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 the spatial or temporal dimension, 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 and/or quality, and "picture" to refer to all the representations (frames) at a single instant in time. Thus, a picture consists of one or more frames, encoding different spatial layers.

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

Given the 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 keyframe with a temporal-layer ID equal to zero.

In flexible mode, each packet can contain up to three 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 scalability structures, which are changing dynamically.

In non-flexible mode, frames are encoded using a fixed, recurring pattern of dependencies; the set of pictures that recur in this pattern is known as a "Picture Group" (or "PG"). In this mode, the inter-picture dependencies (the reference indices) of the PG MUST be pre-specified as part of the Scalability Structure (SS) data. 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.

| Note: A "Picture Group" or "PG", as used in this document, is  
| not the same thing as the term "Group of Pictures" as it is  
| commonly used in video coding, i.e., to mean an independently  
| decodable run of pictures beginning with a keyframe.

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 this specification are encoded as unsigned integers in network octet order.

##### 4.1. RTP Header Usage

The general RTP payload format for VP9 is depicted below.

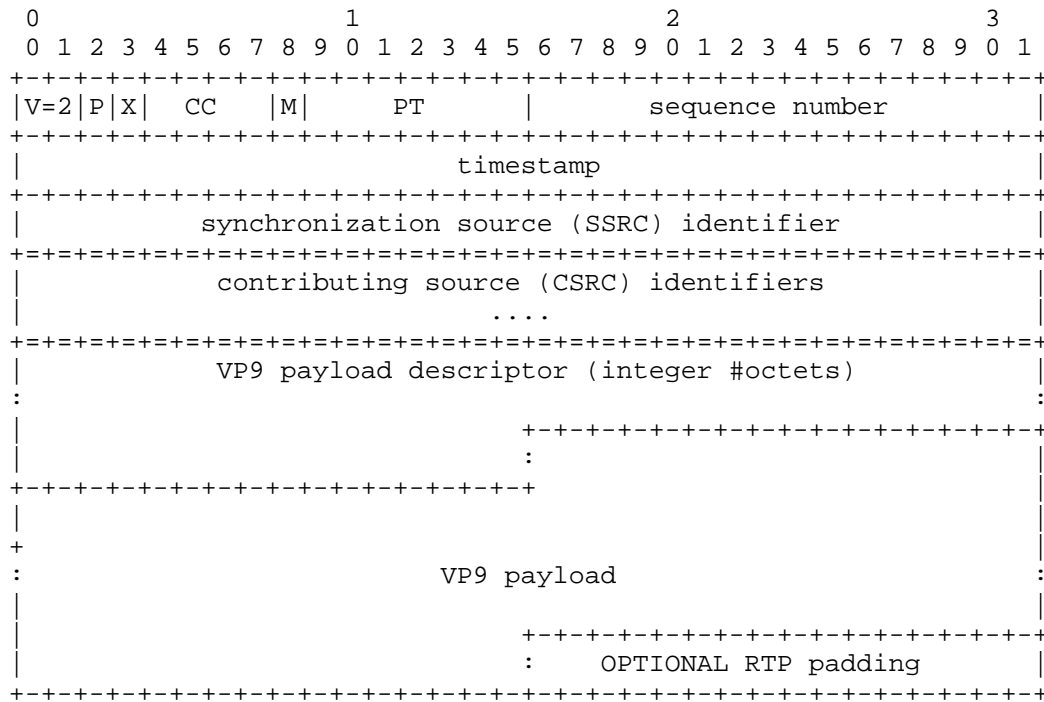


Figure 1: General RTP Payload Format for VP9

See Section 4.2 for more information on the VP9 payload descriptor; the VP9 payload is described in [VP9-BITSTREAM]. OPTIONAL RTP padding MUST NOT be included unless the P bit is set.

**Marker bit (M):** This bit MUST be set to one for the final packet of the highest spatial-layer frame (the final packet of the picture); otherwise, it is zero. Unless spatial scalability is in use for this picture, this bit will have the same value as the E bit described in Section 4.2. Note this bit MUST be set to one 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 6.1 for the mechanism to be used with the Session Description Protocol (SDP) [RFC8866].

**Timestamp:** The RTP timestamp [RFC3550] indicates the time when the input frame was sampled, at a clock rate of 90 kHz. If the input picture is encoded with multiple frames, all of the frames of the picture MUST have the same timestamp.

If a frame has the VP9 show\_frame field set to zero (i.e., it is

meant only to populate a reference buffer without being output), its timestamp MAY alternatively be set to be the same as the subsequent frame with show\_frame equal to one. (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 picture containing a 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 one), the first octets after the RTP header are the VP9 payload descriptor, with the following structure.

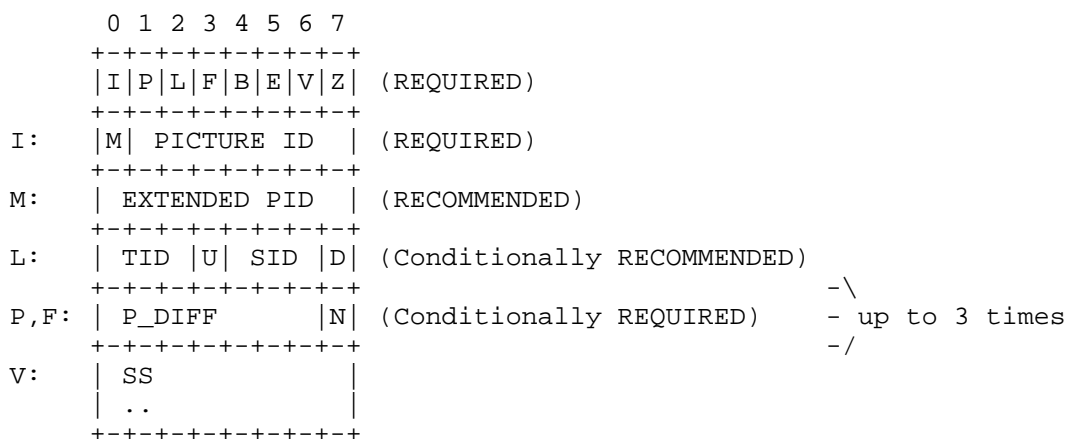


Figure 2: Flexible Mode Format for VP9 Payload Descriptor

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

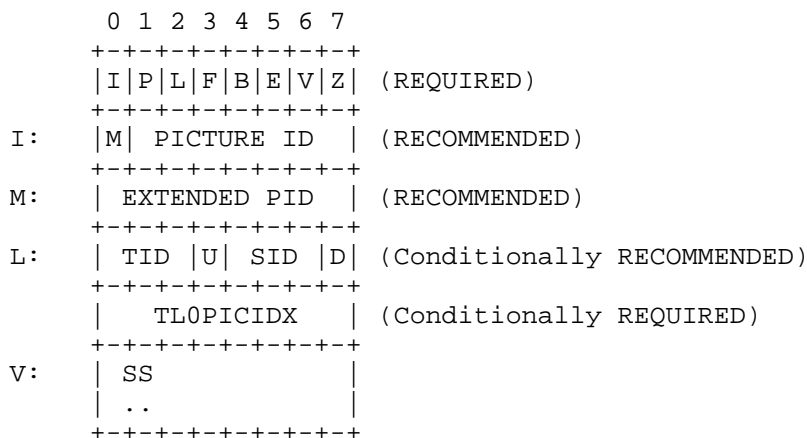


Figure 3: Non-Flexible Mode Format for VP9 Payload Descriptor

Except as noted, the following field descriptions apply to the payload descriptor formats in both Figures 2 and 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 V bit was set in the stream's most recent start of a keyframe (i.e., the SS field was

present) and the F bit is set to zero (i.e., non-flexible scalability mode is in use), then this bit MUST be set on 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 a directly lower spatial-layer frame. P SHOULD also be set to zero when encoding a layer synchronization frame in response to a Layer Refresh Request (LRR) [RFC9627] message (see Section 5.3). When P is set to zero, the Temporal-layer ID (TID) field (described below) MUST also be set to zero (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 one (indicating flexible mode), then only one octet is present for the layer indices. Otherwise, if the F bit is set to zero (indicating non-flexible mode), then two octets are present for the layer indices.
- F: Flexible mode. When set to one, this indicates flexible mode; 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 bit MUST only be set to one if the I bit is also set to one; if the I bit is set to zero, then this bit MUST also be set to zero and ignored by receivers. (Flexible mode's reference indices are defined as offsets from the Picture ID field, so they would have no meaning if I were not set.) The value of the 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 keyframe, and thus one which completely resets the encoder state. This packet will have its P bit equal to zero, SID or L bit (described below) equal to zero, and B bit (described below) equal to one.
- B: Start of Frame. This bit MUST be set to one if the first payload octet of the RTP packet is the beginning of a new VP9 frame; otherwise, it MUST NOT be one. Note that this frame might not be the first frame of a picture.
- E: End of Frame. This bit MUST be set to one for the final RTP packet of a VP9 frame; otherwise, it is zero. 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 B 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 one, indicates that frames with higher spatial layers SID+1 and greater of the current and following pictures do not depend on the current spatial-layer SID frame. This enables a decoder that 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 M is 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, where the sender increments the value by one for each picture it sends. (Note, however, that because a middlebox can discard pictures where permitted by the SS, Picture IDs as received by a receiver might not be contiguous.) This 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 (0x7f or 0x7fff depending on the index size chosen). The receiver MUST NOT assume that the number of bits in the PID stays the same through the session. If this field transitions from 7 bits to 15 bits, the value is zero-extended (i.e., the value after 0x6e is 0x006f); if the field transitions from 15 bits to 7 bits, it is truncated (i.e., the value after 0x1bbe is 0x3f).

In the non-flexible mode (when the F bit is set to zero), this PID is used as an index to the PG specified in the SS data below. In this mode, the PID of the keyframe 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 zero MUST have distinct PID values from subsequent pictures with show\_frame equal to one. 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 field 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 Figures 2 and 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 zero), another octet is used to represent the Temporal Layer 0 Picture Index (8 bits) (TLOPICIDX), as depicted in Figure 3. The TLOPICIDX 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 the current frame. In the case of non-flexible mode, if a PID is mapped to a picture in a specified PG, then the value of the TID MUST match the corresponding TID value of the mapped picture in the PG.

U: Switching up point. When this bit is set to one, if the current picture has a temporal-layer ID equal to value T, then subsequent pictures with temporal-layer ID values higher than T



Figure 4: VP9 Scalability Structure

N\_S: Number of Spatial Layers Minus 1.  $N\_S + 1$  indicates the number of spatial layers present in the VP9 stream.

Y: Each spatial layer's frame resolution is 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: The PG description present flag.

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

N\_G: N\_G indicates the number of pictures in a PG. If N\_G is greater than zero, 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 zero, then for N\_G pictures in the PG, each picture's Temporal-layer ID (TID), switch up point (U), and reference indices (P\_DIFFs) are specified.

The first picture specified in the PG MUST have a TID set to zero.

G set to zero or N\_G set to zero indicates that either there is only one temporal layer (for non-flexible mode) or no fixed inter-picture dependency information is present (for flexible mode) 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; this can be described 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.

R: The number of P\_DIFF fields that are present.

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 the P bit equal to zero, SID or L bit equal to zero, and B bit equal to one. 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 zero).

#### 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 frame that might be removed (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; however, it can 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. Example of a 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 RPSI 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 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 Picture ID 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 (7- or 15-bit) Picture ID 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 a 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 (LRR) [RFC9627] 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.

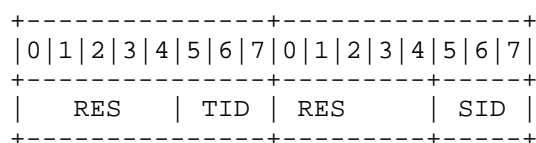


Figure 5: LRR Index Format

Figure 5 shows the format of an LRR's layer index fields for VP9 streams. The two "RES" fields MUST be set to zero 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 switching up points are being encoded in response to an LRR, those packets contain layer indices and the reference field or fields so that the decoder or selective forwarding middleboxes [RFC7667] can make this derivation.

Example:

LRR {1,0}, {2,1} is sent by a Multipoint Control Unit (MCU) when it is currently relaying {1,0} to a receiver that 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 three optional parameters: max-fr, max-fs,

and profile-id.

The max-fr and max-fs 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. A media sender SHOULD NOT send media with a frame rate or frame size exceeding the max-fr and max-fs values signaled. (There may be scenarios, such as pre-encoded media or selective forwarding middleboxes [RFC7667], where a media sender does not have media available that fits within a receiver's max-fs and max-fr values; in such scenarios, a sender MAY exceed the signaled values.)

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 each less than  $\text{int}(\sqrt{\text{max-fs} * 8})$ ; 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. (The profile-id parameter was added relatively late in the development of this specification, so some existing implementations may not send it.)

Informative note: See Table 3 for capabilities of coding profiles defined in Section 7.2 of [VP9-BITSTREAM].

A receiver MUST ignore any parameter unspecified in this specification.

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

Table 2:  
Correspondence between  
profile-id to VP9  
Profile Integer

Profile	Bit Depth	SRGB Colorspace	Chroma Subsampling
0	8	No	YUV 4:2:0

1	8	Yes	YUV 4:2:2,4:4:0 or 4:4:4
2	10 or 12	No	YUV 4:2:0
3	10 or 12	Yes	YUV 4:2:2,4:4:0 or 4:4:4

Table 3: Profile Capabilities

| Note: SRGB (often sRGB) = Standard Red-Green-Blue

## 6.1. SDP Parameters

### 6.1.1. 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-fr and max-fs MUST be included in the "a=fmtp" line of SDP if the receiver wishes to declare its 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 inferred for profile-id.

#### 6.1.1.1. 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
```

### 6.1.2. 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 this configuration parameter or remove the media format (payload type) completely if it is not supported.
- \* The max-fr and max-fs parameters are used declaratively to describe receiver capabilities, even in the Offer/Answer model. The values in an answer are used to describe the answerer's capabilities; thus, their values are set independently of the values in the offer.
- \* 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 and in a subsequent offer, as specified in

[RFC3264]. An answer or subsequent offer MUST NOT contain the payload type number used in the offer unless the profile-id value is exactly the same as in the original offer. However, max-fr and max-fs parameters MAY be changed in subsequent offers and answers, with the same payload type number, if an endpoint wishes to change its declared receiver capabilities.

## 7. Media Type Definition

This registration uses the template defined in [RFC6838] and following [RFC4855].

Type name: video

Subtype name: VP9

Required parameters: N/A

Optional parameters: There are three optional parameters: max-fr, max-fs, and profile-id. See Section 6 for their definition.

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

Security considerations: See Section 8 of RFC 9628.

Interoperability considerations: None

Published specification: VP9 bitstream format [VP9-BITSTREAM] and RFC 9628.

Applications that 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>

Intended usage: COMMON

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

Author: Jonathan Lennox <jonathan.lennox@8x8.com>

Change controller: IETF AVTCore Working Group delegated from the IETF.

## 8. 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]. However, as "Securing the RTP 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 lies with 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.

Implementations of this RTP payload format need to take appropriate security considerations into account. It is extremely important for the decoder to be robust against malicious or malformed payloads and ensure that they do not cause the decoder to overrun its allocated memory or otherwise misbehave. An overrun in allocated memory could lead to arbitrary code execution by an attacker. The same applies to the encoder, even though problems in encoders are (typically) rarer.

This RTP payload format and its media decoder do not exhibit any significant non-uniformity in the receiver-side computational complexity for packet processing; thus, they 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.

## 9. Congestion Control

Congestion control for RTP SHALL be used in accordance with [RFC3550], and with any applicable RTP profile, e.g., [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).

## 10. IANA Considerations

IANA has registered the media type "video/vp9" as specified in Section 7. The media type has also been added to the "RTP Payload Format Media Types" registry of the "Real-Time Transport Protocol (RTP) Parameters" registry group (<https://www.iana.org/assignments/rtp-parameters>) as follows.

Media Type: video  
Subtype: VP9  
Clock Rate (Hz): 90000  
Reference: RFC 9628

## 11. References

### 11.1. 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>>.
- [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>>.

- [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>>.
- [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>>.
- [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>>.
- [RFC9627] Lennox, J., Hong, D., Uberti, J., Holmer, S., and M. Flodman, "The Layer Refresh Request (LRR) RTCP Feedback Message", RFC 9627, DOI 10.17487/RFC9627, March 2025, <<https://www.rfc-editor.org/info/rfc9627>>.
- [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>>.

[RFC7667] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 7667,  
DOI 10.17487/RFC7667, November 2015,  
<<https://www.rfc-editor.org/info/rfc7667>>.

#### 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 feedback.

#### Authors' Addresses

Justin Uberti  
OpenAI  
1455 3rd St  
San Francisco, CA 94158  
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.  
315 Hudson St.  
New York, NY 10013  
United States of America  
Email: [dannyhong@google.com](mailto:dannyhong@google.com)

Jonathan Lennox  
8x8, Inc. / Jitsi  
Jersey City, NJ 07302  
United States of America  
Email: [jonathan.lennox@8x8.com](mailto:jonathan.lennox@8x8.com)