

## RTP Payload Format for Video Codec 1 (VC-1)

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2006).

### Abstract

This memo specifies an RTP payload format for encapsulating Video Codec 1 (VC-1) compressed bit streams, as defined by the Society of Motion Picture and Television Engineers (SMPTE) standard, SMPTE 421M. SMPTE is the main standardizing body in the motion imaging industry, and the SMPTE 421M standard defines a compressed video bit stream format and decoding process for television.

## Table of Contents

1. Introduction .....	2
1.1. Conventions Used in This Document .....	3
2. Definitions and Abbreviations .....	3
3. Overview of VC-1 .....	5
3.1. VC-1 Bit Stream Layering Model .....	6
3.2. Bit-stream Data Units in Advanced Profile .....	7
3.3. Decoder Initialization Parameters .....	7
3.4. Ordering of Frames .....	8
4. Encapsulation of VC-1 Format Bit Streams in RTP .....	9
4.1. Access Units .....	9
4.2. Fragmentation of VC-1 frames .....	10
4.3. Time Stamp Considerations .....	11
4.4. Random Access Points .....	13
4.5. Removal of HRD Parameters .....	14
4.6. Repeating the Sequence Layer Header .....	14
4.7. Signaling of Media Type Parameters .....	15
4.8. The "mode=1" Media Type Parameter .....	16
4.9. The "mode=3" Media Type Parameter .....	16
5. RTP Payload Format Syntax .....	17
5.1. RTP Header Usage .....	17
5.2. AU Header Syntax .....	18
5.3. AU Control Field Syntax .....	19
6. RTP Payload Format Parameters .....	20
6.1. Media type Registration .....	20
6.2. Mapping of media type parameters to SDP .....	28
6.3. Usage with the SDP Offer/Answer Model .....	29
6.4. Usage in Declarative Session Descriptions .....	31
7. Security Considerations .....	32
8. Congestion Control .....	33
9. IANA Considerations .....	34
10. References .....	34
10.1. Normative References .....	34
10.2. Informative References .....	35

## 1. Introduction

This memo specifies an RTP payload format for the video coding standard Video Codec 1, also known as VC-1. The specification for the VC-1 bit stream format and decoding process is published by the Society of Motion Picture and Television Engineers (SMPTE) as SMPTE 421M [1].

VC-1 has a broad applicability, as it is suitable for low bit rate Internet streaming applications to High Definition Television (HDTV) broadcast and Digital Cinema applications with nearly lossless coding. The overall performance of VC-1 is such that bit rate

savings of more than 50% are reported [9] when compared with MPEG-2. See [9] for further details about how VC-1 compares with other codecs, such as MPEG-4 and H.264/AVC. (In [9], VC-1 is referred to by its earlier name, VC-9.)

VC-1 is widely used for downloading and streaming movies on the Internet, in the form of Windows Media Video 9 (WMV-9) [9], because the WMV-9 codec is compliant with the VC-1 standard. VC-1 has also recently been adopted as a mandatory compression format for the high-definition DVD formats HD DVD and Blu-ray.

SMPTE 421M defines the VC-1 bit stream syntax and specifies constraints that must be met by VC-1 conformant bit streams. SMPTE 421M also specifies the complete process required to decode the bit stream. However, it does not specify the VC-1 compression algorithm, thus allowing for different ways of implementing a VC-1 encoder.

The VC-1 bit stream syntax has three profiles. Each profile has specific bit stream syntax elements and algorithms associated with it. Depending on the application in which VC-1 is used, some profiles may be more suitable than others. For example, Simple profile is designed for low bit rate Internet streaming and for playback on devices that can only handle low-complexity decoding. Advanced profile is designed for broadcast applications, such as digital TV, HD DVD, or HDTV. Advanced profile is the only VC-1 profile that supports interlaced video frames and non-square pixels.

Section 2 defines the abbreviations used in this document. Section 3 provides a more detailed overview of VC-1. Sections 4 and 5 define the RTP payload format for VC-1, and section 6 defines the media type and SDP parameters for VC-1. See section 7 for security considerations, and section 8 for congestion control requirements.

### 1.1. Conventions Used in This Document

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 BCP 14, RFC 2119 [2].

## 2. Definitions and Abbreviations

This document uses the definitions in SMPTE 421M [1]. For convenience, the following terms from SMPTE 421M are restated here:

### B-picture:

A picture that is coded using motion compensated prediction from past and/or future reference fields or frames. A B-picture cannot be used for predicting any other picture.

**BI-picture:**

A B-picture that is coded using information only from itself.

A BI-picture cannot be used for predicting any other picture.

**Bit-stream data unit (BDU):**

A unit of the compressed data which may be parsed (i.e., syntax decoded) independently of other information at the same hierarchical level. A BDU can be, for example, a sequence layer header, an entry-point header, a frame, or a slice.

**Encapsulated BDU (EBDU):**

A BDU that has been encapsulated using the encapsulation mechanism described in Annex E of SMPTE 421M [1], to prevent emulation of the start code prefix in the bit stream.

**Entry-point:**

A point in the bit stream that offers random access.

**frame:**

A frame contains lines of spatial information of a video signal. For progressive video, these lines contain samples starting from one time instant and continuing through successive lines to the bottom of the frame. For interlaced video, a frame consists of two fields, a top field and a bottom field. One of these fields will commence one field period later than the other.

**interlace:**

The property of frames where alternating lines of the frame represent different instances in time. In an interlaced frame, one of the fields is meant to be displayed first.

**I-picture:**

A picture coded using information only from itself.

**level:**

A defined set of constraints on the values that may be taken by the parameters (such as bit rate and buffer size) within a particular profile. A profile may contain one or more levels.

**P-picture:**

A picture that is coded using motion compensated prediction from past reference fields or frames.

**picture:**

For progressive video, a picture is identical to a frame, while for interlaced video, a picture may refer to a frame, or the top field or the bottom field of the frame depending on the context.

**profile:**

A defined subset of the syntax of VC-1 with a specific set of coding tools, algorithms, and syntax associated with it. There are three VC-1 profiles: Simple, Main, and Advanced.

**progressive:**

The property of frames where all the samples of the frame represent the same instance in time.

**random access:**

A random access point in the bit stream is defined by the following guarantee: If decoding begins at this point, all frames needed for display after this point will have no decoding dependency on any data preceding this point, and they are also present in the decoding sequence after this point. A random access point is also called an entry-point.

**sequence:**

A coded representation of a series of one or more pictures. In VC-1 Advanced profile, a sequence consists of a series of one or more entry-point segments, where each entry-point segment consists of a series of one or more pictures, and where the first picture in each entry-point segment provides random access. In VC-1 Simple and Main profiles, the first picture in each sequence is an I-picture.

**slice:**

A consecutive series of macroblock rows in a picture, which are encoded as a single unit.

**start codes (SC):**

Unique 32-bit codes that are embedded in the coded bit stream and identify the beginning of a BDU. Start codes consist of a unique three-byte Start Code Prefix (SCP), and a one-byte Start Code Suffix (SCS).

### 3. Overview of VC-1

The VC-1 bit stream syntax consists of three profiles: Simple, Main, and Advanced. Simple profile is designed for low bit rates and for low complexity applications, such as playback of media on personal digital assistants. The maximum bit rate supported by Simple profile

is 384 kbps. Main profile targets high bit rate applications, such as streaming and TV over IP. Main profile supports B-pictures, which provide improved compression efficiency at the cost of higher complexity.

Certain features that can be used to achieve high compression efficiency, such as non-square pixels and support for interlaced pictures, are only included in Advanced profile. The maximum bit rate supported by the Advanced profile is 135 Mbps, making it suitable for nearly lossless encoding of HDTV signals.

Only Advanced profile supports carrying user-data (meta-data) in-band with the compressed bit stream. The user-data can be used for closed captioning support, for example.

Of the three profiles, only Advanced profile allows codec configuration parameters, such as the picture aspect ratio, to be changed through in-band signaling in the compressed bit stream.

For each of the profiles, a certain number of "levels" have been defined. Unlike a "profile", which implies a certain set of features or syntax elements, a "level" is a set of constraints on the values of parameters in a profile, such as the bit rate or buffer size. VC-1 Simple profile has two levels, Main profile has three, and Advanced profile has five. See Annex D of SMPTE 421M [1] for a detailed list of the profiles and levels.

### 3.1. VC-1 Bit Stream Layering Model

The VC-1 bit stream is defined as a hierarchy of layers. This is conceptually similar to the notion of a protocol stack of networking protocols. The outermost layer is called the sequence layer. The other layers are entry-point, picture, slice, macroblock, and block.

In Simple and Main profiles, a sequence in the sequence layer consists of a series of one or more coded pictures. In Advanced profile, a sequence consists of one or more entry-point segments, where each entry-point segment consists of a series of one or more pictures, and where the first picture in each entry-point segment provides random access. A picture is decomposed into macroblocks. A slice comprises one or more contiguous rows of macroblocks.

The entry-point and slice layers are only present in Advanced profile. In Advanced profile, the start of each entry-point layer segment indicates a random access point. In Simple and Main profiles, each I-picture is a random access point.

Each picture can be coded as an I-picture, P-picture, skipped picture, BI-picture, or as a B-picture. These terms are defined in [section 2](#) of this document and in [section 4.12](#) of SMPTE 421M [1].

### 3.2. Bit-stream Data Units in Advanced Profile

In Advanced profile, each picture and slice is considered a Bit-stream Data Unit (BDU). A BDU is always byte-aligned and is defined as a unit that can be parsed (i.e., syntax decoded) independently of other information in the same layer.

The beginning of a BDU is signaled by an identifier called Start Code (SC). Sequence layer headers and entry-point headers are also BDUs and thus can be easily identified by their Start Codes. See Annex E of SMPTE 421M [1] for a complete list of Start Codes. Blocks and macroblocks are not BDUs and thus do not have a Start Code and are not necessarily byte-aligned.

The Start Code consists of four bytes. The first three bytes are 0x00, 0x00 and 0x01. The fourth byte is called the Start Code Suffix (SCS) and it is used to indicate the type of BDU that follows the Start Code. For example, the SCS of a sequence layer header (0x0F) is different from the SCS of an entry-point header (0x0E). The Start Code is always byte-aligned and is transmitted in network byte order.

To prevent accidental emulation of the Start Code in the coded bit stream, SMPTE 421M defines an encapsulation mechanism that uses byte stuffing. A BDU that has been encapsulated by this mechanism is referred to as an Encapsulated BDU, or EBDU.

### 3.3. Decoder Initialization Parameters

In VC-1 Advanced profile, the sequence layer header contains parameters that are necessary to initialize the VC-1 decoder.

The parameters apply to all entry-point segments until the next occurrence of a sequence layer header in the coded bit stream.

The parameters in the sequence layer header include the Advanced profile level, the maximum dimensions of the coded frames, the aspect ratio, interlace information, the frame rate and up to 31 leaky bucket parameter sets for the Hypothetical Reference Decoder (HRD).

[Section 6.1](#) of SMPTE 421M [1] provides the formal specification of the sequence layer header.

A sequence layer header is not defined for VC-1 Simple and Main profiles. For these profiles, decoder initialization parameters **MUST** be conveyed out-of-band. The decoder initialization parameters for Simple and Main profiles include the maximum dimensions of the coded frames and a leaky bucket parameter set for the HRD. [Section 4.7](#) specifies how the parameters are conveyed by this RTP payload format.

Each leaky bucket parameter set for the HRD specifies a peak transmission bit rate and a decoder buffer capacity. The coded bit stream is restricted by these parameters. The HRD model does not mandate buffering by the decoder. Its purpose is to limit the encoder's bit rate fluctuations according to a basic buffering model so that the resources necessary to decode the bit stream are predictable. The HRD has a constant-delay mode and a variable-delay mode. The constant-delay mode is appropriate for broadcast and streaming applications, while the variable-delay mode is designed for video-conferencing applications.

Annex C of SMPTE 421M [1] specifies the usage of the hypothetical reference decoder for VC-1 bit streams. A general description of the theory of the HRD can be found in [10].

For Simple and Main profiles, the current buffer fullness value for the HRD leaky bucket is signaled using the BF syntax element in the picture header of I-pictures and BI-pictures.

For Advanced profile, the entry-point header specifies current buffer fullness values for the leaky buckets in the HRD. The entry-point header also specifies coding control parameters that are in effect until the occurrence of the next entry-point header in the bit stream. The concept of an entry-point layer applies only to VC-1 Advanced profile. See [Section 6.2](#) of SMPTE 421M [1] for the formal specification of the entry-point header.

### 3.4. Ordering of Frames

Frames are transmitted in the same order in which they are captured, except if B-pictures or BI-pictures are present in the coded bit stream. A BI-picture is a special kind of B-picture, and in the remainder of this section the terms B-picture and B-frame also apply to BI-pictures and BI-frames, respectively.

When B-pictures are present in the coded bit stream, the frames are transmitted such that the frames that the B-pictures depend on are transmitted first. This is referred to as the coded order of the frames.



The rules for how a decoder converts frames from the coded order to the display order are stated in [section 5.4](#) of SMPTE 421M [1]. In short, if B-pictures may be present in the coded bit stream, a hypothetical decoder implementation needs to buffer one additional decoded frame. When an I-frame or a P-frame is received, the frame can be decoded immediately but it is not displayed until the next I- or P-frame is received. However, B-frames are displayed immediately.

Figure 1 illustrates the timing relationship between the capture of frames, their coded order, and the display order of the decoded frames, when B-pictures are present in the coded bit stream. The figure shows that the display of frame P4 is delayed until frame P7 is received, while frames B2 and B3 are displayed immediately.

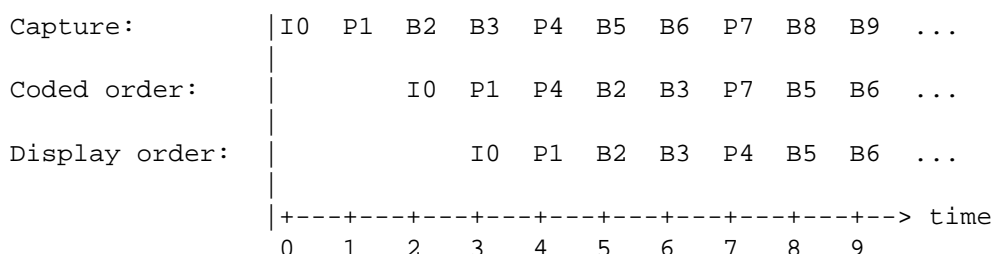


Figure 1. Frame reordering when B-pictures are present

If B-pictures are not present, the coded order and the display order are identical, and frames can then be displayed without the additional delay shown in Figure 1.

#### 4. Encapsulation of VC-1 Format Bit Streams in RTP

##### 4.1. Access Units

Each RTP packet contains an integral number of application data units (ADUs). For VC-1 format bit streams, an ADU is equivalent to one Access Unit (AU). An Access Unit is defined as the AU header (defined in [section 5.2](#)) followed by a variable length payload, with the rules and constraints described in [sections 4.1](#) and [4.2](#). Figure 2 shows the layout of an RTP packet with multiple AUs.

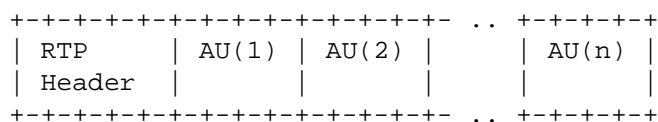


Figure 2. RTP packet structure

Each Access Unit MUST start with the AU header defined in [section 5.2](#). The AU payload MUST contain data belonging to exactly one VC-1 frame. This means that data from different VC-1 frames will always be in different AUs. However, it is possible for a single VC-1 frame to be fragmented across multiple AUs (see [section 4.2](#)).

In the case of interlaced video, a VC-1 frame consists of two fields that may be coded as separate pictures. The two pictures still belong to the same VC-1 frame.

The following rules apply to the contents of each AU payload when VC-1 Advanced profile is used:

- The AU payload MUST contain VC-1 bit stream data in EBDU format (i.e., the bit stream must use the byte-stuffing encapsulation mode defined in Annex E of SMPTE 421M [1].)
- The AU payload MAY contain multiple EBDUs, e.g., a sequence layer header, an entry-point header, a frame (picture) header, a field header, and multiple slices and the associated user-data. However, all slices and their corresponding macroblocks MUST belong to the same video frame.
- The AU payload MUST start at an EBDU boundary, except when the AU payload contains a fragmented frame, in which case the rules in [section 4.2](#) apply.

When VC-1 Simple or Main profiles are used, the AU payload MUST start at the beginning of a frame, except when the AU payload contains a fragmented frame. [Section 4.2](#) describes how to handle fragmented frames.

Access Units MUST be byte-aligned. If the data in an AU (EBDUs in the case of Advanced profile and frame in the case of Simple and Main) does not end at an octet boundary, up to 7 zero-valued padding bits MUST be added to achieve octet-alignment.

#### 4.2. Fragmentation of VC-1 frames

Each AU payload SHOULD contain a complete VC-1 frame. However, if this would cause the RTP packet to exceed the MTU size, the frame SHOULD be fragmented into multiple AUs to avoid IP-level fragmentation. When an AU contains a fragmented frame, this MUST be indicated by setting the FRAG field in the AU header as defined in [section 5.3](#).

AU payloads that do not contain a fragmented frame or that contain the first fragment of a frame MUST start at an EBDU boundary if Advanced profile is used. In this case, for Simple and Main profiles, the AU payload MUST start at the beginning of a frame.

If Advanced profile is used, AU payloads that contain a fragment of a frame other than the first fragment SHOULD start at an EBDU boundary, such as at the start of a slice.

However, slices are only defined for Advanced profile, and are not always used. Blocks and macroblocks are not BDUs (have no Start Code) and are not byte-aligned. Therefore, it may not always be possible to continue a fragmented frame at an EBDU boundary. One can determine if an AU payload starts at an EBDU boundary by inspecting the first three bytes of the AU payload. The AU payload starts at an EBDU boundary if the first three bytes are identical to the Start Code Prefix (i.e., 0x00, 0x00, 0x01).

In the case of Simple and Main profiles, since the blocks and macroblocks are not byte-aligned, the fragmentation boundary may be chosen arbitrarily.

If an RTP packet contains an AU with the last fragment of a frame, additional AUs SHOULD NOT be included in the RTP packet.

If the PTS Delta field in the AU header is present, each fragment of a frame MUST have the same presentation time. If the DTS Delta field in the AU header is present, each fragment of a frame MUST have the same decode time.

#### 4.3. Time Stamp Considerations

VC-1 video frames MUST be transmitted in the coded order. A coded order implies that no frames are dependent on subsequent frames, as discussed in [section 3.4](#). When a video frame consists of a single picture, the presentation time of the frame is identical to the presentation time of the picture. When the VC-1 interlace coding mode is used, frames may contain two pictures, one for each field. In that case, the presentation time of a frame is the presentation time of the field that is displayed first.

The RTP timestamp field MUST be set to the presentation time of the video frame contained in the first AU in the RTP packet. The presentation time can be used as the timestamp field in the RTP header because it differs from the sampling instant of the frame only by an arbitrary constant offset.

If the video frame in an AU has a presentation time that differs from the RTP timestamp field, then the presentation time **MUST** be specified using the PTS Delta field in the AU header. Since the RTP timestamp field must be identical to the presentation time of the first video frame, this can only happen if an RTP packet contains multiple AUs. The syntax of the PTS Delta field is defined in [section 5.2](#).

The decode time of a VC-1 frame is always monotonically increasing when the video frames are transmitted in the coded order. If neither B- nor BI-pictures are present in the coded bit stream, then the decode time of a frame **SHALL** be equal to the presentation time of the frame. A BI-picture is a special kind of B-picture, and in the remainder of this section the terms B-picture and B-frame also apply to BI-pictures and BI-frames, respectively.

If B-pictures may be present in the coded bit stream, then the decode times of frames are determined as follows:

- B-frames:  
The decode time **SHALL** be equal to the presentation time of the B-frame.
- First non-B frame in the coded order:  
The decode time **SHALL** be at least one frame period less than the decode time of the next frame in the coded order. A frame period is defined as the inverse of the frame rate used in the coded bit stream (e.g., 100 milliseconds if the frame rate is 10 frames per seconds.) For bit streams with a variable frame rate, the maximum frame rate **SHALL** determine the frame period. If the maximum frame rate is not specified, the maximum frame rate allowed by the profile and level **SHALL** be used.
- Non-B frames (other than the first frame in the coded order):  
The decode time **SHALL** be equal to the presentation time of the previous non-B frame in the coded order.

As an example, consider Figure 1 in [section 3.4](#). To determine the decode time of the first frame, I0, one must first determine the decode time of the next frame, P1. Because P1 is a non-B frame, its decode time is equal to the presentation time of I0, which is 3 time units. Thus, the decode time of I0 must be at least one frame period less than 3. In this example, the frame period is 1, because one frame is displayed every time unit. Consequently, the decode time of I0 is chosen as 2 time units. The decode time of the third frame in the coded order, P4, is 4, because it must be equal to the presentation time of the previous non-B frame in the coded order, P1. On the other hand, the decode time of B-frame B2 is 5 time units, which is identical to its presentation time.

If the decode time of a video frame differs from its presentation time, then the decode time MUST be specified using the DTS Delta field in the AU header. The syntax of the DTS Delta field is defined in [section 5.2](#).

Receivers are not required to use the DTS Delta field. However, possible uses include buffer management and pacing of frames prior to decoding. If RTP packets are lost, it is possible to use the DTS Delta field to determine if the sequence of lost RTP packets contained reference frames or only B-frames. This can be done by comparing the decode and presentation times of the first frame received after the lost sequence against the presentation time of the last reference frame received prior to the lost sequence.

Knowing if the stream will contain B-pictures may help the receiver allocate resources more efficiently and can reduce delay, as an absence of B-pictures in the stream implies that no reordering of frames will be needed between the decoding process and the display of the decoded frames. This may be important for interactive applications.

The receiver SHALL assume that the coded bit stream may contain B-pictures in the following cases:

- Advanced profile:  
If the value of the "bpic" media type parameter defined in [section 6.1](#) is 1, or if the "bpic" parameter is not specified.
- Main profile:  
If the MAXBFRAMES field in STRUCT\_C decoder initialization parameter has a non-zero value. STRUCT\_C is conveyed in the "config" media type parameter, which is defined in [section 6.1](#).

Simple profile does not use B-pictures.

#### 4.4. Random Access Points

The entry-point header contains information that is needed by the decoder to decode the frames in that entry-point segment. This means that in the event of lost RTP packets, the decoder may be unable to decode frames until the next entry-point header is received.

The first frame after an entry-point header is a random access point into the coded bit stream. Simple and Main profiles do not have entry-point headers, so for those profiles, each I-picture is a random access point.

To allow the RTP receiver to detect that an RTP packet that was lost contained a random access point, this RTP payload format defines a field called "RA Count". This field is present in every AU, and its value is incremented (modulo 256) for every random access point. For additional details, see the definition of "RA Count" in [section 5.2](#).

To make it easy to determine if an AU contains a random access point, this RTP payload format also defines a bit called the "RA" flag in the AU Control field. This bit is set to 1 only on those AU's that contain a random access point. The RA bit is defined in [section 5.3](#).

#### 4.5. Removal of HRD Parameters

The sequence layer header of Advanced profile may include up to 31 leaky bucket parameter sets for the Hypothetical Reference Decoder (HRD). Each leaky bucket parameter set specifies a possible peak transmission bit rate (HRD\_RATE) and a decoder buffer capacity (HRD\_BUFFER). See [section 3.3](#) for additional discussion about the HRD.

If the actual peak transmission rate is known by the RTP sender, the RTP sender MAY remove all leaky bucket parameter sets except for the one corresponding to the actual peak transmission rate.

For each leaky bucket parameter set in the sequence layer header, there is also a parameter in the entry-point header that specifies the initial fullness (HRD\_FULL) of the leaky bucket.

If the RTP sender has removed any leaky bucket parameter sets from the sequence layer header, then for any removed leaky bucket parameter set, it MUST also remove the corresponding HRD\_FULL parameter in the entry-point header.

Removing leaky bucket parameter sets, as described above, may significantly reduce the size of the sequence layer headers and the entry-point headers.

#### 4.6. Repeating the Sequence Layer Header

To improve robustness against loss of RTP packets, it is RECOMMENDED that if the sequence layer header changes, it should be repeated frequently in the bit stream. In this case, it is RECOMMENDED that the number of leaky bucket parameters in the sequence layer header and the entry-point headers be reduced to one, as described in [section 4.5](#). This will help reduce the overhead caused by repeating the sequence layer header.

Any data in the VC-1 bit stream, including repeated copies of the sequence header itself, must be accounted for when computing the leaky bucket parameter for the HRD. See [section 3.3](#) for a discussion about the HRD.

If the value of TFCNTRFLAG in the sequence layer header is 1, each picture header contains a frame counter field (TFCNTR). Each time the sequence layer header is inserted in the bit stream, the value of this counter MUST be reset.

To allow the RTP receiver to detect that an RTP packet that was lost contained a new sequence layer header, the AU Control field defines a bit called the "SL" flag. This bit is toggled when a sequence layer header is transmitted, but only if that header is different from the most recently transmitted sequence layer header. The SL bit is defined in [section 5.3](#).

#### 4.7. Signaling of Media Type Parameters

When this RTP payload format is used with SDP, the decoder initialization parameters described in [section 3.3](#) MUST be signaled in SDP using the media type parameters specified in [section 6.1](#). [Section 6.2](#) specifies how to map the media type parameters to SDP [5], [section 6.3](#) defines rules specific to the SDP Offer/Answer model, and [section 6.4](#) defines rules for when SDP is used in a declarative style.

When Simple or Main profiles are used, it is not possible to change the decoder initialization parameters through the coded bit stream. Any changes to the decoder initialization parameters would have to be done through out-of-band means, e.g., by a SIP [14] re-invite or similar means that convey an updated session description.

When Advanced profile is used, the decoder initialization parameters MAY be changed by inserting a new sequence layer header or an entry-point header in the coded bit stream.

The sequence layer header specifies the VC-1 level, the maximum size of the coded frames and optionally also the maximum frame rate. The media type parameters "level", "width", "height", and "framerate" specify upper limits for these parameters. Thus, the sequence layer header MAY specify values that are lower than the values of the media type parameters "level", "width", "height", or "framerate", but the sequence layer header MUST NOT exceed the values of any of these media type parameters.

#### 4.8. The "mode=1" Media Type Parameter

In certain applications using Advanced profile, the sequence layer header never changes. This MAY be signaled with the media type parameter "mode=1". (The "mode" parameter is defined in [section 6.1.](#)) The "mode=1" parameter serves as a "hint" to the RTP receiver that all sequence layer headers in the bit stream will be identical. If "mode=1" is signaled and a sequence layer header is present in the coded bit stream, then it MUST be identical to the sequence layer header specified by the "config" media type parameter.

Since the sequence layer header never changes in "mode=1", the RTP sender MAY remove it from the bit stream. Note, however, that if the value of TFCNTRFLAG in the sequence layer header is 1, each picture header contains a frame counter field (TFCNTR). This field is reset each time the sequence layer header occurs in the bit stream. If the RTP sender chooses to remove the sequence layer header, then it MUST ensure that the resulting bit stream is still compliant with the VC-1 specification (e.g., by adjusting the TFCNTR field, if necessary.)

#### 4.9. The "mode=3" Media Type Parameter

In certain applications using Advanced profile, both the sequence layer header and the entry-point header never change. This MAY be signaled with the media type parameter "mode=3". The same rules apply to "mode=3" as for "mode=1", described in [section 4.8.](#) Additionally, if "mode=3" is signaled, then the RTP sender MAY "compress" the coded bit stream by not including sequence layer headers and entry-point headers in the RTP packets.

The RTP receiver MUST "decompress" the coded bit stream by re-inserting the entry-point headers prior to delivering the coded bit stream to the VC-1 decoder. The sequence layer header does not need to be decompressed by the receiver, as it never changes.

If "mode=3" is signaled and the RTP receiver receives a complete AU or the first fragment of an AU, and the RA bit is set to 1 but the AU does not begin with an entry-point header, then this indicates that the entry-point header has been "compressed". In that case, the RTP receiver MUST insert an entry-point header at the beginning of the AU. When inserting the entry-point header, the RTP receiver MUST use the one that was specified by the "config" media type parameter.



## 5. RTP Payload Format Syntax

### 5.1. RTP Header Usage

The format of the RTP header is specified in [RFC 3550](#) [3] and is reprinted in Figure 3 for convenience.

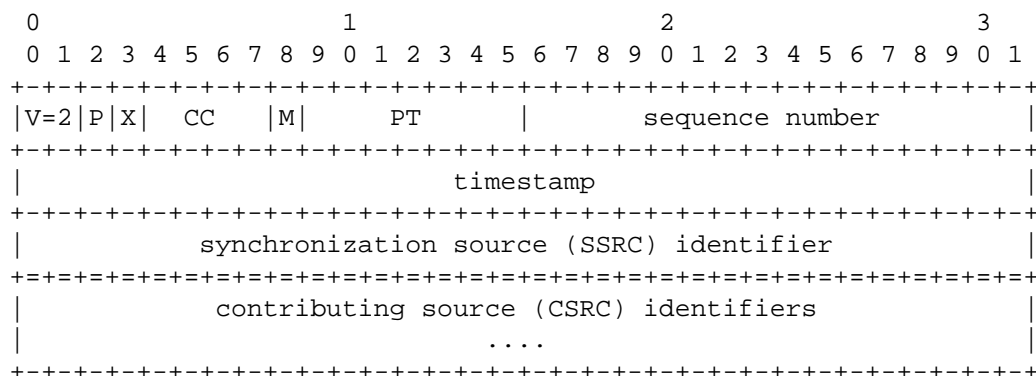


Figure 3. RTP header according to [RFC 3550](#)

The fields of the fixed RTP header have their usual meaning, which is defined in [RFC 3550](#) and by the RTP profile in use, with the following additional notes:

Marker bit (M): 1 bit

This bit is set to 1 if the RTP packet contains an Access Unit containing a complete VC-1 frame or the last fragment of a VC-1 frame.

Payload type (PT): 7 bits

This document does not assign an RTP payload type for this RTP payload format. The assignment of a payload type has to be performed either through the RTP profile used or in a dynamic way.

Sequence Number: 16 bits

The RTP receiver can use the sequence number field to recover the coded order of the VC-1 frames. A typical VC-1 decoder will require the VC-1 frames to be delivered in coded order. When VC-1 frames have been fragmented across RTP packets, the RTP receiver can use the sequence number field to ensure that no fragment is missing.

Timestamp: 32 bits

The RTP timestamp is set to the presentation time of the VC-1 frame in the first Access Unit. A clock rate of 90 kHz MUST be used.

## 5.2. AU Header Syntax

The Access Unit header consists of a one-byte AU Control field, the RA Count field, and 3 optional fields. All fields MUST be written in network byte order. The structure of the AU header is illustrated in Figure 4.

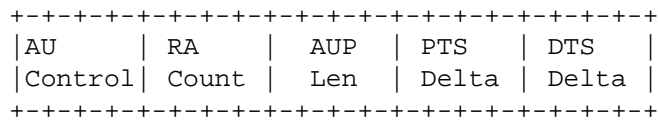


Figure 4. Structure of AU header

AU Control: 8 bits

The usage of the AU Control field is defined in [section 5.3](#).

RA Count: 8 bits

Random Access Point Counter. This field is a binary modulo 256 counter. The value of this field MUST be incremented by 1 each time an AU is transmitted where the RA bit in the AU Control field is set to 1. The initial value of this field is undefined and MAY be chosen randomly.

AUP Len: 16 bits

Access Unit Payload Length. Specifies the size, in bytes, of the payload of the Access Unit. The field does not include the size of the AU header itself. The field MUST be included in each AU header in an RTP packet, except for the last AU header in the packet. If this field is not included, the payload of the Access Unit SHALL be assumed to extend to the end of the RTP payload.

PTS Delta: 32 bits

Presentation time delta. Specifies the presentation time of the frame as a 2's complement offset (delta) from the timestamp field in the RTP header of this RTP packet. The PTS Delta field MUST use the same clock rate as the timestamp field in the RTP header.

This field SHOULD NOT be included in the first AU header in the RTP packet, because the RTP timestamp field specifies the presentation time of the frame in the first AU. If this field

is not included, the presentation time of the frame SHALL be assumed to be specified by the timestamp field in the RTP header.

#### DTS Delta: 32 bits

Decode time delta. Specifies the decode time of the frame as a 2's complement offset (delta) between the presentation time and the decode time. Note that if the presentation time is larger than the decode time, this results in a value for the DTS Delta field that is greater than zero. The DTS Delta field MUST use the same clock rate as the timestamp field in the RTP header. If this field is not included, the decode time of the frame SHALL be assumed to be identical to the presentation time of the frame.

### 5.3. AU Control Field Syntax

The structure of the 8-bit AU Control field is shown in Figure 5.

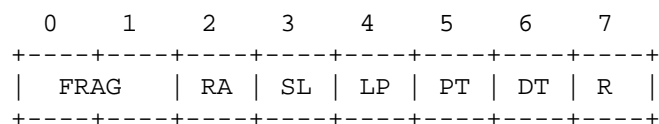


Figure 5. Syntax of AU Control field.

#### FRAG: 2 bits

Fragmentation Information. This field indicates if the AU payload contains a complete frame or a fragment of a frame. It MUST be set as follows:

- 0: The AU payload contains a fragment of a frame other than the first or last fragment.
- 1: The AU payload contains the first fragment of a frame.
- 2: The AU payload contains the last fragment of a frame.
- 3: The AU payload contains a complete frame (not fragmented.)

#### RA: 1 bit

Random Access Point indicator. This bit MUST be set to 1 if the AU contains a frame that is a random access point. In the case of Simple and Main profiles, any I-picture is a random access point.

In the case of Advanced profile, the first frame after an entry-point header is a random access point.

If entry-point headers are not transmitted at every random access point, this MUST be indicated using the media type parameter "mode=3".

SL: 1 bit

Sequence Layer Counter. This bit MUST be toggled, i.e., changed from 0 to 1 or from 1 to 0, if the AU contains a sequence layer header and if it is different from the most recently transmitted sequence layer header. Otherwise, the value of this bit must be identical to the value of the SL bit in the previous AU.

The initial value of this bit is undefined and MAY be chosen randomly.

The bit MUST be 0 for Simple and Main profile bit streams or if the sequence layer header never changes.

LP: 1 bit

Length Present. This bit MUST be set to 1 if the AU header includes the AUP Len field.

PT: 1 bit

PTS Delta Present. This bit MUST be set to 1 if the AU header includes the PTS Delta field.

DT: 1 bit

DTS Delta Present. This bit MUST be set to 1 if the AU header includes the DTS Delta field.

R: 1 bit

Reserved. This bit MUST be set to 0 and MUST be ignored by receivers.

## 6. RTP Payload Format Parameters

### 6.1. Media type Registration

This registration uses the template defined in [RFC 4288](#) [7] and follows [RFC 3555](#) [8].

Type name: video

Subtype name: vc1

Required parameters:

profile:

The value is an integer identifying the VC-1 profile. The following values are defined:

- 0: Simple profile
- 1: Main profile
- 3: Advanced profile

If the profile parameter is used to indicate properties of a coded bit stream, it indicates the VC-1 profile that a decoder has to support when it decodes the bit stream.

If the profile parameter is used for capability exchange or in a session setup procedure, it indicates the VC-1 profile that the codec supports.

level:

The value is an integer that specifies the level of the VC-1 profile.

For Advanced profile, valid values are 0 through 4, which correspond to levels L0 through L4, respectively. For Simple and Main profiles, the following values are defined:

- 1: Low Level
- 2: Medium Level
- 3: High Level (only valid for Main profile)

If the level parameter is used to indicate properties of a coded bit stream, it indicates the highest level of the VC-1 profile that a decoder has to support when it decodes the bit stream. Note that support for a level implies support for all numerically lower levels of the given profile.

If the level parameter is used for capability exchange or in a session setup procedure, it indicates the highest level of the VC-1 profile that the codec supports. See [section 6.3 of RFC 4425](#) for specific rules for how this parameter is used with the SDP Offer/Answer model.

Optional parameters:

config:

The value is a base16 [6] (hexadecimal) representation of an octet string that expresses the decoder initialization parameters. Decoder initialization parameters are mapped onto the base16 octet string in an MSB-first basis. The first bit of the decoder initialization parameters MUST be located at the MSB of the first octet. If the decoder initialization parameters are not multiples of 8 bits, up to 7 zero-valued padding bits MUST be added in the last octet to achieve octet alignment.

For Simple and Main profiles, the decoder initialization parameters are STRUCT\_C, as defined in Annex J of SMPTE 421M [1].

For Advanced profile, the decoder initialization parameters are a sequence layer header directly followed by an entry-point header. The two headers MUST be in EBDU format, meaning that they must include their Start Codes and must use the encapsulation method defined in Annex E of SMPTE 421M [1].

width:

The value is an integer greater than zero, specifying the maximum horizontal size of the coded frames, in luma samples (pixels in the luma picture).

For Simple and Main profiles, the value SHALL be identical to the actual horizontal size of the coded frames.

For Advanced profile, the value SHALL be greater than, or equal to, the largest horizontal size of the coded frames.

If this parameter is not specified, it defaults to the maximum horizontal size allowed by the specified profile and level.

height:

The value is an integer greater than zero, specifying the maximum vertical size of the coded frames, in luma samples (pixels in a progressively coded luma picture).

For Simple and Main profiles, the value SHALL be identical to the actual vertical size of the coded frames.

For Advanced profile, the value SHALL be greater than, or equal to, the largest vertical size of the coded frames.

If this parameter is not specified, it defaults to the maximum vertical size allowed by the specified profile and level.

bitrate:

The value is an integer greater than zero, specifying the peak transmission rate of the coded bit stream in bits per second. The number does not include the overhead caused by RTP encapsulation, i.e., it does not include the AU headers, or any of the RTP, UDP, or IP headers.

If this parameter is not specified, it defaults to the maximum bit rate allowed by the specified profile and level. See the values for "RMax" in Annex D of SMPTE 421M [1].

buffer:

The value is an integer specifying the leaky bucket size, B, in milliseconds, required to contain a stream transmitted at the transmission rate specified by the bitrate parameter. This parameter is defined in the hypothetical reference decoder model for VC-1, in Annex C of SMPTE 421M [1].

Note that this parameter relates to the codec bit stream only, and does not account for any buffering time that may be required to compensate for jitter in the network.

If this parameter is not specified, it defaults to the maximum buffer size allowed by the specified profile and level. See the values for "BMax" and "RMax" in Annex D of SMPTE 421M [1].

framerate:

The value is an integer greater than zero, specifying the maximum number of frames per second in the coded bit stream, multiplied by 1000 and rounded to the nearest integer value. For example, 30000/1001 (approximately 29.97) frames per second is represented as 29970.

This parameter can be used to control resource allocation at the receiver. For example, a receiver may choose to perform additional post-processing on decoded frames only if the frame rate is expected to be low. The parameter MUST NOT be used for pacing of the rendering process, since the actual frame rate may differ from the specified value.

If the parameter is not specified, it defaults to the maximum frame rate allowed by the specified profile and level.

**bpic:**

This parameter signals that B- and BI-pictures may be present when Advanced profile is used. If this parameter is present, and B- or BI-pictures may be present in the coded bit stream, this parameter **MUST** be equal to 1.

A value of 0 indicates that B- and BI-pictures **SHALL NOT** be present in the coded bit stream, even if the sequence layer header changes. Inclusion of this parameter with a value of 0 is **RECOMMENDED**, if neither B- nor BI-pictures are included in the coded bit stream.

This parameter **MUST NOT** be used with Simple and Main profiles. For Main profile, the presence of B- and BI-pictures is indicated by the **MAXBFRAMES** field in **STRUCT\_C** decoder initialization parameter.

For Advanced profile, if this parameter is not specified, a value of 1 **SHALL** be assumed.

**mode:**

The value is an integer specifying the use of the sequence layer header and the entry-point header. This parameter is only defined for Advanced profile. The following values are defined:

- 0: Both the sequence layer header and the entry-point header may change, and changed headers will be included in the RTP packets.
- 1: The sequence layer header specified in the config parameter never changes. The rules in section 4.8 of [RFC 4425](#) **MUST** be followed.
- 3: The sequence layer header and the entry-point header specified in the config parameter never change. The rules in [section 4.9 of RFC 4425](#) **MUST** be followed.

If the mode parameter is not specified, a value of 0 **SHALL** be assumed. The mode parameter **SHOULD** be specified if modes 1 or 3 apply to the VC-1 bit stream.

**max-width, max-height, max-bitrate, max-buffer, max-framerate:**

These parameters are defined for use in a capability exchange procedure. The parameters do not signal properties of the coded bit stream, but rather upper limits or



preferred values for the "width", "height", "bitrate", "buffer", and "framerate" parameters. Section 6.3 of [RFC 4425](#) provides specific rules for how these parameters are used with the SDP Offer/Answer model.

Receivers that signal support for a given profile and level MUST support the maximum values for these parameters for that profile and level. For example, a receiver that indicates support for Main profile, Low level, must support a width of 352 luma samples and a height of 288 luma samples, even if this requires scaling the image to fit the resolution of a smaller display device.

A receiver MAY use any of the max-width, max-height, max-bitrate, max-buffer, and max-framerate parameters to indicate preferred capabilities. For example, a receiver may choose to specify values for max-width and max-height that match the resolution of its display device, since a bit stream encoded using those parameters would not need to be rescaled.

If any of the max-width, max-height, max-bitrate, max-buffer, and max-framerate parameters signal a capability that is less than the required capabilities of the signaled profile and level, then the parameter SHALL be interpreted as a preferred value for that capability.

Any of the parameters MAY also be used to signal capabilities that exceed the required capabilities of the signaled profile and level. In that case, the parameter SHALL be interpreted as the maximum value that can be supported for that capability.

When more than one parameter from the set (max-width, max-height, max-bitrate, max-buffer, and max-framerate) is present, all signaled capabilities MUST be supported simultaneously.

A sender or receiver MUST NOT use these parameters to signal capabilities that meet the requirements of a higher level of the VC-1 profile than that specified in the "level" parameter, even if the sender or receiver can support all the properties of the higher level, except if specifying a higher level is not allowed due to other restrictions. As an example of such a restriction, in the SDP Offer/Answer model, the value of the level parameter that can be used in an Answer is limited by what was specified in the Offer.

**max-width:**

The value is an integer greater than zero, specifying a horizontal size for the coded frames, in luma samples (pixels in the luma picture). If the value is less than the maximum horizontal size allowed by the profile and level, then the value specifies the preferred horizontal size. Otherwise, it specifies the maximum horizontal size that is supported.

If this parameter is not specified, it defaults to the maximum horizontal size allowed by the specified profile and level.

**max-height:**

The value is an integer greater than zero, specifying a vertical size for the coded frames, in luma samples (pixels in a progressively coded luma picture). If the value is less than the maximum vertical size allowed by the profile and level, then the value specifies the preferred vertical size. Otherwise, it specifies the maximum vertical size that is supported.

If this parameter is not specified, it defaults to the maximum vertical size allowed by the specified profile and level.

**max-bitrate:**

The value is an integer greater than zero, specifying a peak transmission rate for the coded bit stream in bits per second. The number does not include the overhead caused by RTP encapsulation, i.e., it does not include the AU headers, or any of the RTP, UDP, or IP headers.

If the value is less than the maximum bit rate allowed by the profile and level, then the value specifies the preferred bit rate. Otherwise, it specifies the maximum bit rate that is supported.

If this parameter is not specified, it defaults to the maximum bit rate allowed by the specified profile and level. See the values for "RMax" in Annex D of SMPTE 421M [1].

**max-buffer:**

The value is an integer specifying a leaky bucket size, B, in milliseconds, required to contain a stream transmitted at the transmission rate specified by the max-bitrate

parameter. This parameter is defined in the hypothetical reference decoder model for VC-1, in Annex C of SMPTE 421M [1].

Note that this parameter relates to the codec bit stream only and does not account for any buffering time that may be required to compensate for jitter in the network.

If the value is less than the maximum leaky bucket size allowed by the max-bitrate parameter and the profile and level, then the value specifies the preferred leaky bucket size. Otherwise, it specifies the maximum leaky bucket size that is supported for the bit rate specified by the max-bitrate parameter.

If this parameter is not specified, it defaults to the maximum buffer size allowed by the specified profile and level. See the values for "BMax" and "RMax" in Annex D of SMPTE 421M [1].

max-framerate:

The value is an integer greater than zero, specifying a number of frames per second for the coded bit stream. The value is the frame rate multiplied by 1000 and rounded to the nearest integer value. For example, 30000/1001 (approximately 29.97) frames per second is represented as 29970.

If the value is less than the maximum frame rate allowed by the profile and level, then the value specifies the preferred frame rate. Otherwise, it specifies the maximum frame rate that is supported.

If the parameter is not specified, it defaults to the maximum frame rate allowed by the specified profile and level.

Encoding considerations:

This media type is framed and contains binary data.

Security considerations:

See [Section 7 of RFC 4425](#).

Interoperability considerations:

None.

Published specification:

[RFC 4425](#).

Applications that use this media type:

Multimedia streaming and conferencing tools.

Additional Information:

None.

Person & email address to contact for further information:

Anders Klemets <anderskl@microsoft.com>

IETF AVT working group.

Intended Usage:

COMMON

Restrictions on usage:

This media type depends on RTP framing; therefore, it is only defined for transfer via RTP [3].

Authors:

Anders Klemets

Change controller:

IETF Audio/Video Transport Working Group delegated from the IESG.

## 6.2. Mapping of media type parameters to SDP

The information carried in the media type specification has a specific mapping to fields in the Session Description Protocol (SDP) [4]. If SDP is used to specify sessions using this payload format, the mapping is done as follows:

- o The media name in the "m=" line of SDP MUST be video (the type name).
- o The encoding name in the "a=rtpmap" line of SDP MUST be vc1 (the subtype name).
- o The clock rate in the "a=rtpmap" line MUST be 90000.
- o The REQUIRED parameters "profile" and "level" MUST be included in the "a=fmtp" line of SDP.

These parameters are expressed in the form of a semicolon separated list of parameter=value pairs.

- o The OPTIONAL parameters "config", "width", "height", "bitrate", "buffer", "framerate", "bpic", "mode", "max-width", "max-height", "max-bitrate", "max-buffer", and "max-framerate", when present, MUST be included in the "a=fmtp" line of SDP.

These parameters are expressed in the form of a semicolon separated list of parameter=value pairs:

```
a=fmtp:<dynamic payload type> <parameter
name>=<value>[,<value>][; <parameter name>=<value>]
```

- o Any unknown parameters to the device that uses the SDP MUST be ignored. For example, parameters defined in later specifications MAY be copied into the SDP and MUST be ignored by receivers that do not understand them.

### 6.3. Usage with the SDP Offer/Answer Model

When VC-1 is offered over RTP using SDP in an Offer/Answer model [5] for negotiation for unicast usage, the following rules and limitations apply:

- o The "profile" parameter MUST be used symmetrically, i.e., the answerer MUST either maintain the parameter or remove the media format (payload type) completely if the offered VC-1 profile is not supported.
- o The "level" parameter specifies the highest level of the VC-1 profile supported by the codec.

The answerer MUST NOT specify a numerically higher level in the answer than that specified in the offer. The answerer MAY specify a level that is lower than that specified in the offer, i.e., the level parameter can be "downgraded".

If the offer specifies the sendrecv or sendonly direction attribute and the answer downgrades the level parameter, this may require a new offer to specify an updated "config" parameter. If the "config" parameter cannot be used with the level specified in the answer, then the offerer MUST initiate another Offer/Answer round or not use media format (payload type).

- o The parameters "config", "bpic", "width", "height", "framerate", "bitrate", "buffer", and "mode", describe the properties of the VC-1 bit stream that the offerer or answerer is sending for this media format configuration.

In the case of unicast usage and when the direction attribute in the offer or answer is `recvonly`, the interpretation of these parameters is undefined and they **MUST NOT** be used.

- o The parameters `"config"`, `"width"`, `"height"`, `"bitrate"`, and `"buffer"` **MUST** be specified when the direction attribute is `sendrecv` or `sendonly`.
- o The parameters `"max-width"`, `"max-height"`, `"max-framerate"`, `"max-bitrate"`, and `"max-buffer"` **MAY** be specified in an offer or an answer, and their interpretation is as follows:

When the direction attribute is `sendonly`, the parameters describe the limits of the VC-1 bit stream that the sender is capable of producing for the given profile and level, and for any lower level of the same profile.

When the direction attribute is `recvonly` or `sendrecv`, the parameters describe properties of the receiver implementation. If the value of a property is less than that allowed by the level of the VC-1 profile, then it **SHALL** be interpreted as a preferred value and the sender's VC-1 bit stream **SHOULD NOT** exceed it. If the value of a property is greater than what is allowed by the level of the VC-1 profile, then it **SHALL** be interpreted as the upper limit of the value that the receiver accepts for the given profile and level, and for any lower level of the same profile.

For example, if a `recvonly` or `sendrecv` offer specifies `"profile=0;level=1;max-bitrate=48000"`, then 48 kbps is merely a suggested bit rate, because all receiver implementations of Simple profile, Low level, are required to support bit rates of up to 96 kbps. Assuming that the offer is accepted, the answerer should specify `"bitrate=48000"` in the answer, but any value up to 96000 is allowed. But if the offer specifies `"max-bitrate=200000"`, this means that the receiver implementation supports a maximum of 200 kbps for the given profile and level (or lower level). In this case, the answerer is allowed to answer with a bitrate parameter of up to 200000.

- o If an offerer wishes to have non-symmetrical capabilities between sending and receiving, e.g., use different levels in each direction, then the offerer has to offer different RTP sessions. This can be done by specifying different media lines declared as `"recvonly"` and `"sendonly"`, respectively.

For streams being delivered over multicast, the following rules apply in addition:

- o The "level" parameter specifies the highest level of the VC-1 profile used by the participants in the multicast session. The value of this parameter MUST NOT be changed by the answerer. Thus, a payload type can be either accepted unaltered or removed.
- o The parameters "config", "bpic", "width", "height", "framerate", "bitrate", "buffer", and "mode", specify properties of the VC-1 bit stream that will be sent and/or received on the multicast session. The parameters MAY be specified, even if the direction attribute is recvonly.

The values of these parameters MUST NOT be changed by the answerer. Thus, a payload type can be either accepted unaltered or removed.

- o The values of the parameters "max-width", "max-height", "max-framerate", "max-bitrate", and "max-buffer" MUST be supported by the answerer for all streams declared as sendrecv or recvonly. Otherwise, one of the following actions MUST be performed: the media format is removed or the session is rejected.

#### 6.4. Usage in Declarative Session Descriptions

When VC-1 is offered over RTP using SDP in a declarative style, as in RTSP [12] or SAP [13], the following rules and limitations apply:

- o The parameters "profile" and "level" indicate only the properties of the coded bit stream. They do not imply a limit on capabilities supported by the sender.
- o The parameters "config", "width", "height", "bitrate", and "buffer" MUST be specified.
- o The parameters "max-width", "max-height", "max-framerate", "max-bitrate", and "max-buffer" MUST NOT be used.

An example of media representation in SDP is as follows (Simple profile, Medium level):

```
m=video 49170 RTP/AVP 98
a=rtpmap:98 vc1/90000
a=fmtp:98 profile=0;level=2;width=352;height=288;framerate=15000;
bitrate=384000;buffer=2000;config=4e291800
```

## 7. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [4], and in any appropriate RTP profile. This implies that confidentiality of the media streams is achieved by encryption; for example, through the application of SRTP [11].

A potential denial-of-service threat exists for data encodings using compression techniques that have non-uniform receiver-end computational load. The attacker can inject pathological RTP packets into the stream that are complex to decode and that cause the receiver to be overloaded. VC-1 is particularly vulnerable to such attacks, because it is possible for an attacker to generate RTP packets containing frames that affect the decoding process of many future frames. Therefore, the usage of data origin authentication and data integrity protection of at least the RTP packet is RECOMMENDED; for example, with SRTP [11].

Note that the appropriate mechanism to ensure confidentiality and integrity of RTP packets and their payloads is dependent on the application and on the transport and signaling protocols employed. Thus, although SRTP is given as an example above, other possible choices exist.

VC-1 bit streams can carry user-data, such as closed captioning information and content meta-data. The VC-1 specification does not define how to interpret user-data. Identifiers for user-data are required to be registered with SMPTE. It is conceivable for types of user-data to be defined to include programmatic content, such as scripts or commands that would be executed by the receiver. Depending on the type of user-data, it might be possible for a sender to generate user-data in a non-compliant manner to crash the receiver or make it temporarily unavailable. Senders that transport VC-1 bit streams SHOULD ensure that the user-data is compliant with the specification registered with SMPTE (see Annex F of [1].) Receivers SHOULD prevent malfunction in case of non-compliant user-data.

It is important to note that VC-1 streams can have very high bandwidth requirements (up to 135 Mbps for high-definition video). This causes a potential for denial-of-service if transmitted onto many Internet paths. Therefore, users of this payload format MUST comply with the congestion control requirements described in [section 8](#).



## 8. Congestion Control

Congestion control for RTP SHALL be used in accordance with [RFC 3550](#) [3], and with any applicable RTP profile; e.g., [RFC 3551](#) [15].

If best-effort service is being used, users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

The bit rate adaptation necessary for obeying the congestion control principle is easily achievable when real-time encoding is used. When pre-encoded content is being transmitted, bandwidth adaptation requires one or more of the following:

- The availability of more than one coded representation of the same content at different bit rates. The switching between the different representations can normally be performed in the same RTP session by switching streams at random access point boundaries.
- The existence of non-reference frames (e.g., B-frames) in the bit stream. Non-reference frames can be discarded by the transmitter prior to encapsulation in RTP.

Only when non-downgradable parameters (such as the VC-1 "profile" parameter) are required to be changed does it become necessary to terminate and re-start the media stream. This may be accomplished by using a different RTP payload type.

Regardless of the method used for bandwidth adaptation, the resulting bit stream MUST be compliant with the VC-1 specification [1]. For example, if non-reference frames are discarded, then the FRMCNT syntax element (Simple and Main profile frames only) and the optional TFCNTR syntax element (Advanced profile frames only) must increment as if no frames had been discarded. Because the TFCNTR syntax element counts the frames in the display order, which is different from the order in which they are transmitted (the coded order), it will require the transmitter to "look ahead" or buffer some number of frames.

As another example, when switching between different representations of the same content, it may be necessary to signal a discontinuity by modifying the FRMCNT field, or if Advanced profile is used, by setting the BROKEN\_LINK flag in the entry-point header to 1.

This payload format may also be used in networks that provide quality-of-service guarantees. If enhanced service is being used, receivers SHOULD monitor packet loss to ensure that the service that was requested is actually being delivered. If it is not, then they SHOULD assume that they are receiving best-effort service and behave accordingly.

## 9. IANA Considerations

IANA has registered the media type "video/vcl" and the associated RTP payload format in the Media Types registry and in the RTP Payload Format MIME types registry, as specified in [section 6.1](#).

## 10. References

### 10.1. Normative References

- [1] Society of Motion Picture and Television Engineers, "VC-1 Compressed Video Bitstream Format and Decoding Process", SMPTE 421M.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [3] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [4] Handley, M. and V. Jacobson, "SDP: Session Description Protocol", [RFC 2327](#), April 1998.
- [5] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [6] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", [RFC 3548](#), July 2003.
- [7] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [8] Casner, S. and P. Hoschka, "MIME Type Registration of RTP Payload Formats", [RFC 3555](#), July 2003.

## 10.2. Informative References

- [9] Srinivasan, S., Hsu, P., Holcomb, T., Mukerjee, K., Regunathan, S.L., Lin, B., Liang, J., Lee, M., and J. Ribas-Corbera, "Windows Media Video 9: overview and applications", Signal Processing: Image Communication, Volume 19, Issue 9, October 2004.
- [10] Ribas-Corbera, J., Chou, P.A., and S.L. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC", IEEE Transactions on Circuits and Systems for Video Technology, August 2003.
- [11] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", [RFC 3711](#), March 2004.
- [12] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", [RFC 2326](#), April 1998.
- [13] Handley, M., Perkins, C., and E. Whelan, "Session Announcement Protocol", [RFC 2974](#), October 2000.
- [14] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", [RFC 2543](#), March 1999.
- [15] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, [RFC 3551](#), July 2003.

## Acknowledgements

Thanks to Regis Crinon, Miska Hannuksela, Colin Perkins, Shankar Regunathan, Gary Sullivan, Stephan Wenger, and Magnus Westerlund for providing detailed feedback on this document.

## Author's Address

Anders Klemets  
Microsoft Corp.  
1 Microsoft Way  
Redmond, WA 98052  
USA

EMail: [Anders.Klemets@microsoft.com](mailto:Anders.Klemets@microsoft.com)

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).