Internet Engineering Task Force (IETF)
Request for Comments: 8450
Category: Standards Track
ISSN: 2070-1721

J. Weaver
BBC

October 2018

RTP Payload Format for VC-2 High Quality (HQ) Profile

Abstract

This memo describes an RTP payload format for the High Quality (HQ) profile of Society of Motion Picture and Television Engineers Standard ST 2042-1, known as VC-2. This document describes the transport of HQ Profile VC-2 in RTP packets and has applications for low-complexity, high-bandwidth streaming of both lossless and lossy compressed video.

The HQ profile of VC-2 is intended for low-latency video compression (with latency potentially on the order of lines of video) at high data rates (with compression ratios on the order of 2:1 or 4:1).

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/rfc8450.

Table of Contents

1. Introduction

This memo specifies an RTP payload format for the video coding standard Society of Motion Picture and Television Engineers ST 2042-1:2017 [VC2], also known as VC-2

The VC-2 codec is a wavelet-based codec intended primarily for professional video use with high bit-rates and only low levels of compression. It has been designed to have a low level of complexity and potentially a very low latency through both encoder and decoder: with some choices of parameters, this latency may be as low as a few lines of video.

The low level of complexity in the VC-2 codec allows for this low-latency operation but also means that it lacks many of the more powerful compression techniques used in other codecs. As such, it is suitable for low compression ratios that produce coded data rates around half to a quarter of that of uncompressed video, at a similar visual quality.

The primary use for VC-2 is likely to be in professional video production environments.

2. Conventions, Definitions, and Acronyms

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 VC-2 specification defines a VC-2 Stream as being composed of one or more Sequences. Each Sequence is independently decodable, containing all of the needed parameters and metadata for configuring the decoder.

Each Sequence consists of a series of 13-octet Parse Info Headers and variable-length Data Units. The Sequence begins and ends with a Parse Info Header, and each Data Unit is preceded by a Parse Info Header. Data Units come in a variety of types, and the type of a Data Unit is signaled in the preceding Parse Info Header. The most important types are the Sequence Header, which contains configuration data needed by the decoder, and several types of Coded Picture, which contain the coded data for the pictures themselves. Each picture represents a frame in a progressively scanned video Sequence or a field in an interlaced video Sequence.

The first Data Unit in a Sequence as produced by an encoder is always a Sequence Header; however, Sequences can be joined in the middle, so it should not be assumed that the first Data Unit received will always be a Sequence Header.

The High Quality (HQ) profile for VC-2 restricts the types of Parse Info Headers that may appear in the Sequence (and hence also the types of Data Unit) to only:

- Sequence Headers (which are always followed by a Data Unit),
- High Quality Pictures (which are always followed by a Data Unit),
- High Quality Fragments (which are always followed by a Data Unit),
- Auxiliary Data (which are always followed by a Data Unit),
- Padding Data (which are always followed by a Data Unit), and
- End of Sequence (which are never followed by a Data Unit).

At the time of writing, there is no definition for the use of Auxiliary Data in VC-2, and Padding Data is required to be ignored by all receivers.

Each High Quality Picture Data Unit contains a set of parameters for the picture followed by a series of Coded Slices, each representing a rectangular region of the transformed picture. Slices within a picture may vary in coded length, but all represent the same shape and size of rectangle in the picture.

Each High Quality Fragment Data Unit contains either a set of parameters for a picture or a series of Coded Slices. Fragments carry the same data as pictures, but broken up into smaller units to facilitate transmission via packet-based protocols such as RTP.

This payload format only makes use of Fragments, not pictures.

4. Payload Format

In this specification, each RTP packet is used to carry data corresponding to a single Parse Info Header and its following Data Unit (if it has one). A single packet MAY NOT contain data from more than one Parse Info Header or Data Unit. A single Parse Info Header and Data Unit pair MUST NOT be split across more than one packet. The sole exception to this rule is that an Auxiliary Data Unit MAY be split between multiple packets, using the B and E flags to indicate start and end.

This specification only covers the transport of Sequence Headers (together with their accompanying Data Unit), High Quality Fragments (together with their accompanying Data Unit), Auxiliary Data (together with their accompanying Data Unit), and (optionally) End Sequence Headers and Padding Data (for which no Data Unit it carried).

High Quality Pictures can be transported by converting them into an equivalent set of High Quality Fragments. The size of Fragments should be chosen so as to fit within the MTU of the network in use.

For this reason, this document defines six types of RTP packets in a VC-2 media stream:

- a VC-2 Sequence Header (Figure 1) (see Section 11 of the VC-2 specification [VC2]),
- a Picture Fragment containing the VC-2 Transform Parameters for a Picture (Figure 2) (see Section 14 of the VC-2 specification [VC2]),
- a Picture Fragment containing VC-2 Coded Slices (Figure 3) for a picture (see Section 14 of the VC-2 specification [VC2]),
- the end of a VC-2 Sequence (Figure 4) (see Section 10.5.2 of the VC-2 specification [VC2]),
- the contents of an Auxiliary Data Unit (Figure 5) (see Section 10.4.4 of the VC-2 specification [VC2]), and
- an indication of the presence of a padding Data Unit (Figure 6) (see Section 10.4.5 of the VC-2 specification [VC2]).

These six packet types can be distinguished by the fact that they use different codes in the Parse Code ("PC") field, except for the two types of Picture Fragment that use the same value in PC but have different values in the "No. of Slices" field.

The options for PC codes are explained in more detail in Section 4.3.

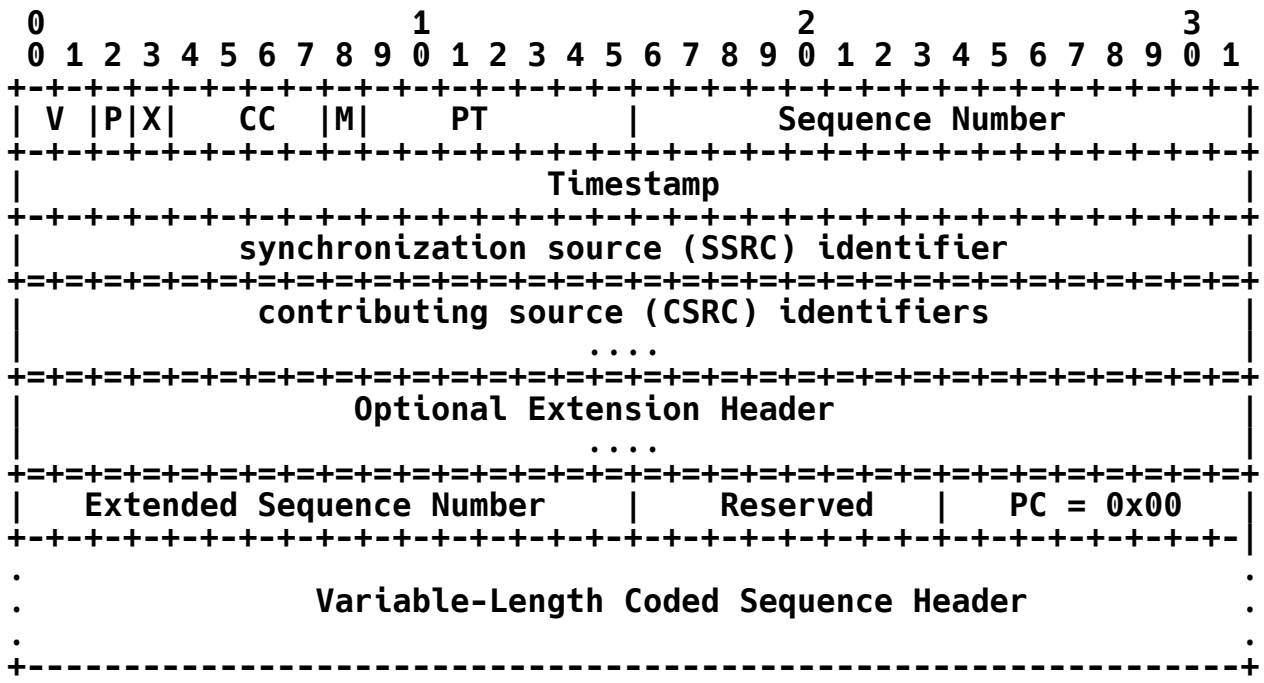

Figure 1: RTP Payload Format for Sequence Header

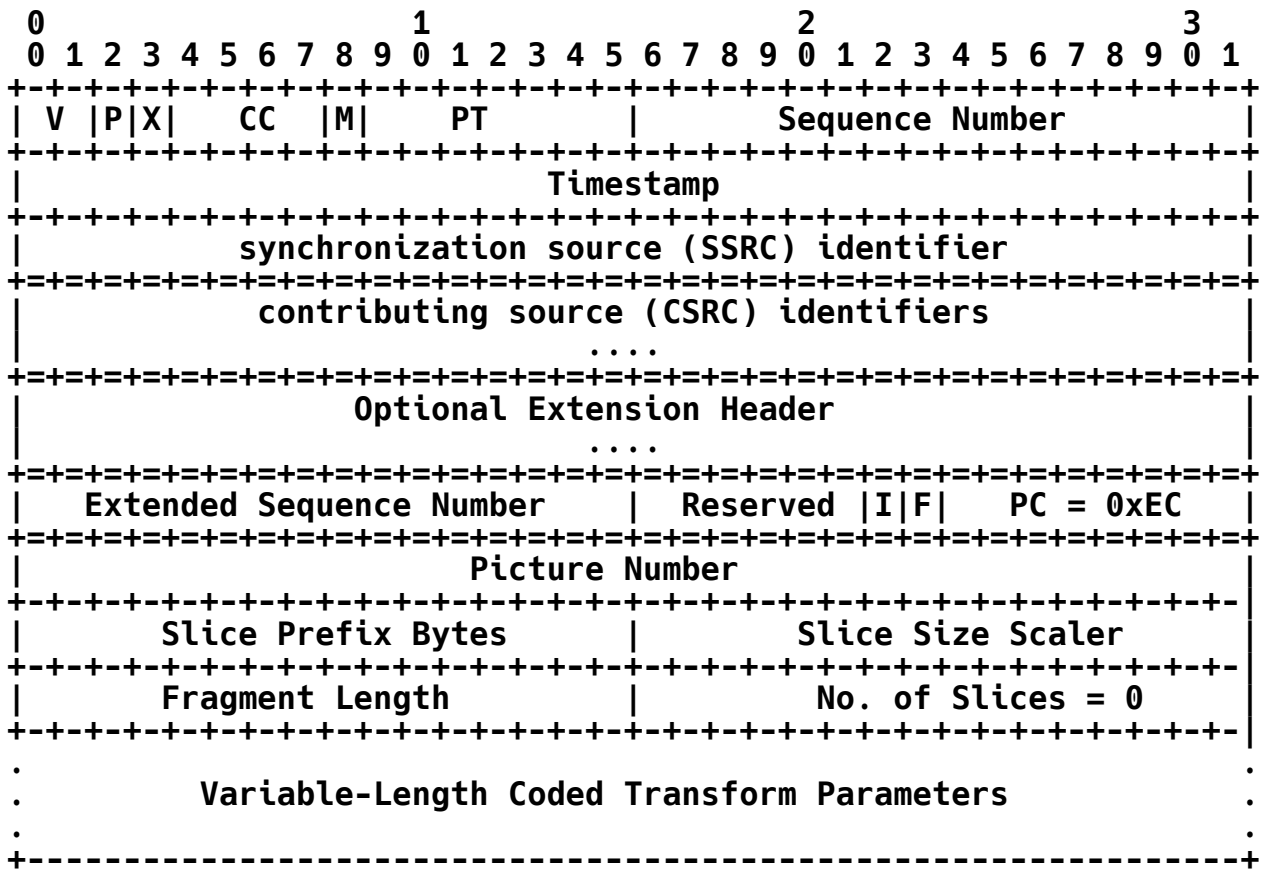

Figure 2: RTP Payload Format for Transform Parameters Fragment

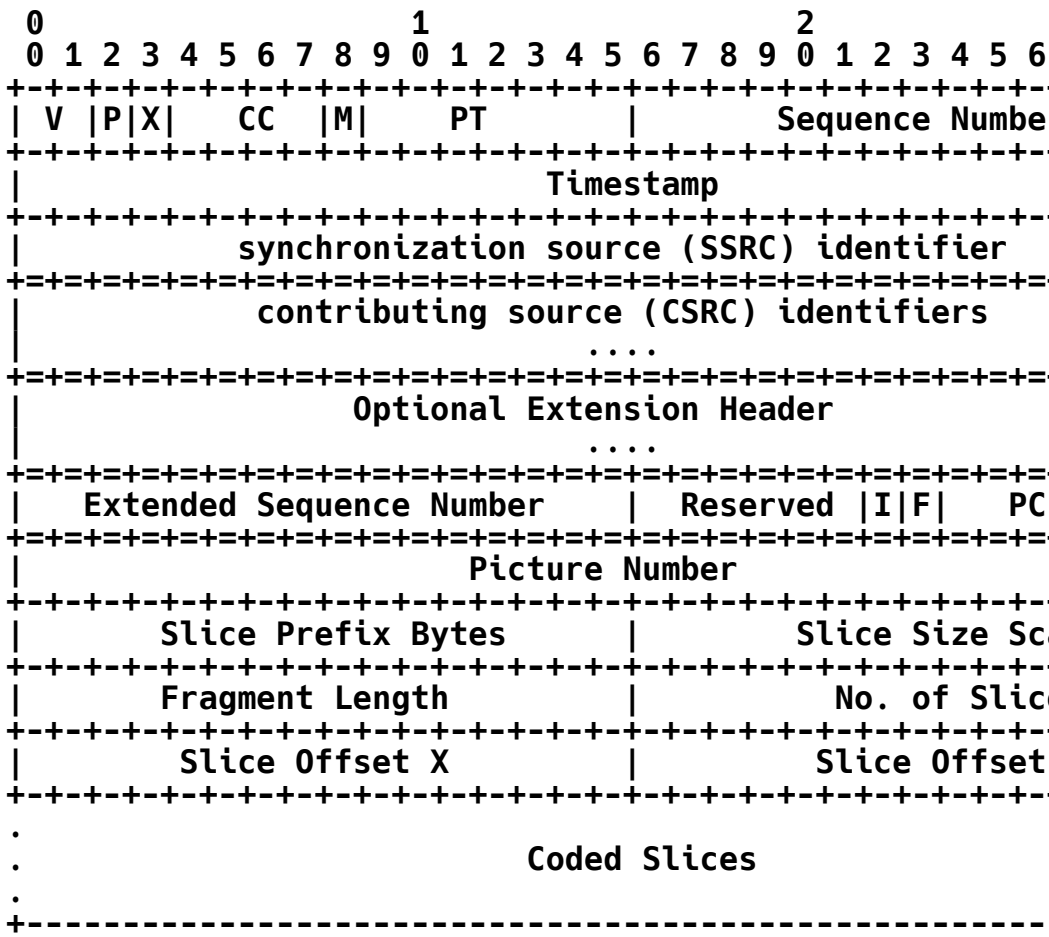

Figure 3: RTP Payload Format for Fragment Containing Slices

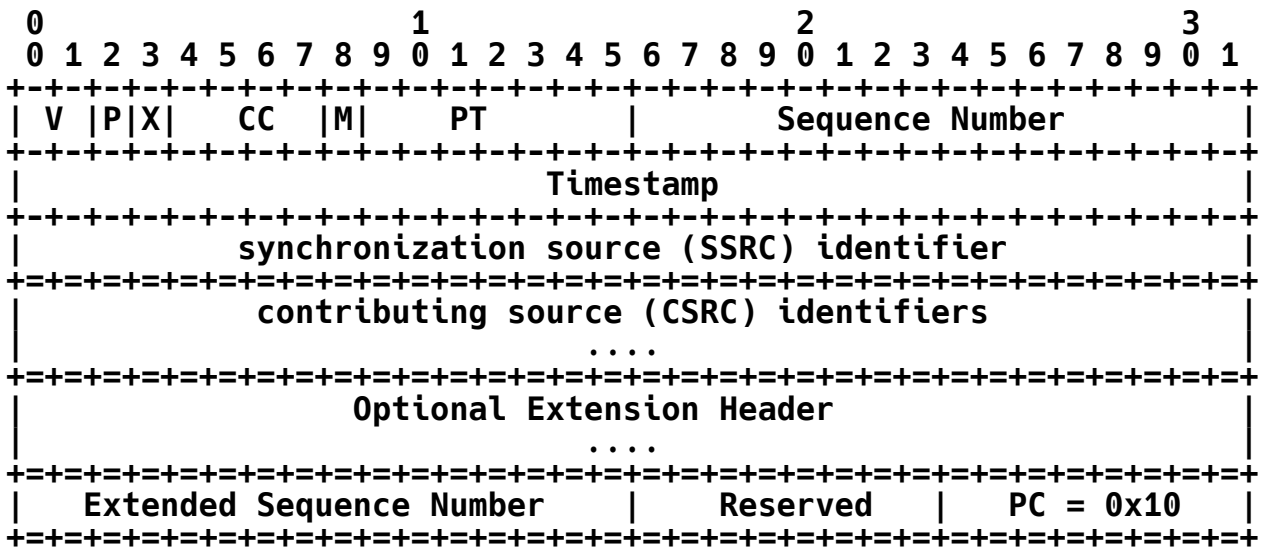

Figure 4: RTP Payload Format for End of Sequence

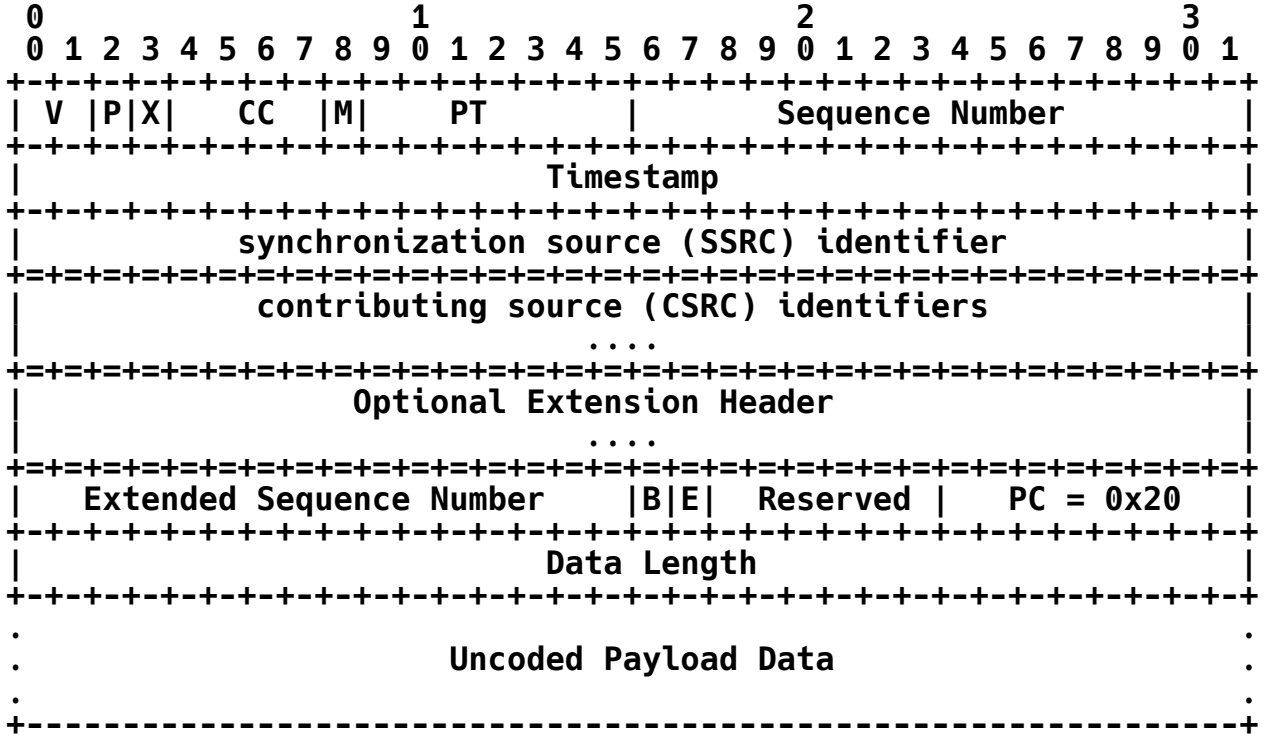

Figure 5: RTP Payload Format for Auxiliary Data

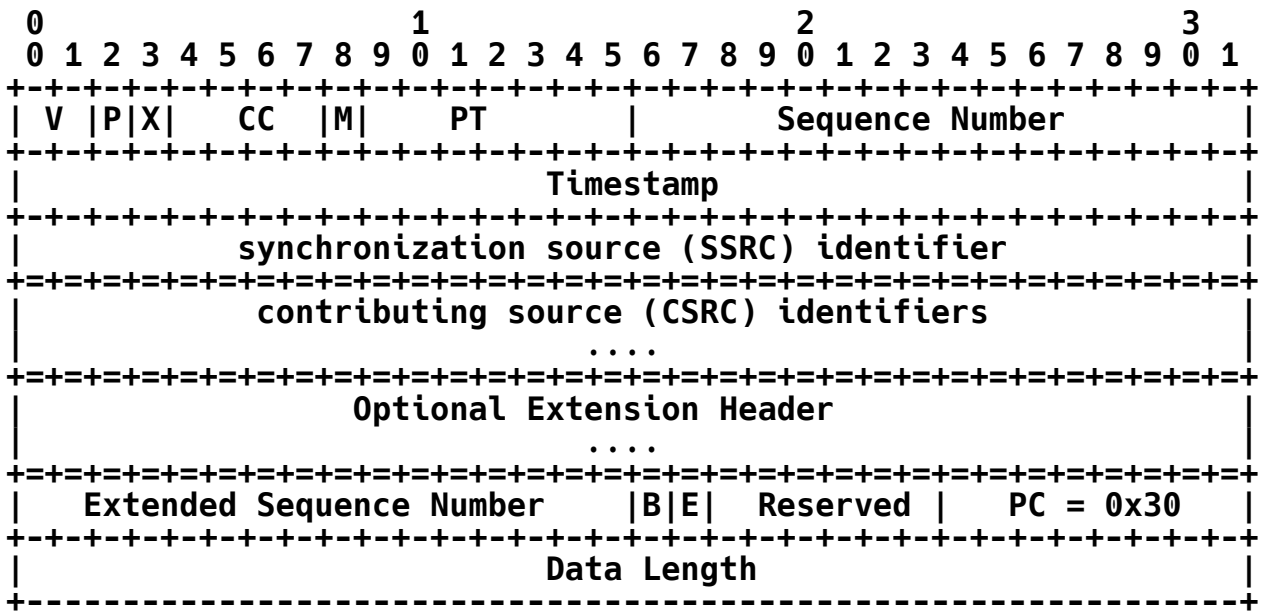

Figure 6: RTP Payload Format for Padding Data

All fields in the headers longer than a single bit are interpreted as unsigned integers in network byte order.

4.1. RTP Header Usage

The fields of the RTP header have the following additional notes on their usage:

Marker Bit (M): 1 bit. The marker bit MUST be set on any packet that contains the final slice in a coded picture and MUST NOT be set otherwise.

Payload Type (PT): 7 bits. A dynamically allocated payload type field that designates the payload as VC-2-coded video.

Sequence Number: 16 bits. Because the data rate of VC-2-coded Streams can often be very high, in the order of gigabits rather than megabits per second, the standard 16-bit RTP sequence number can cycle very quickly. For this reason, the sequence number is extended to 32 bits, and this field MUST hold the low-order 16 bits of this value.

Timestamp: 32 bits. If the packet contains Transform Parameters or Coded Slice data for a coded picture, then the timestamp corresponds to the sampling instant of the coded picture. A 90kHz clock SHOULD be used. A single RTP packet MUST NOT contain coded data for more than one coded picture, so there is no ambiguity here.

A Sequence Header packet SHOULD have the same timestamp as the picture that will follow it in the Stream. An End of Sequence packet SHOULD have the same timestamp as the previous picture that appeared in the Stream.

The remaining RTP header fields are used as specified in RTP [RFC3550].

4.2. Payload Header

The fields of the extended headers are defined as follows:

Extended Sequence Number: 16 bits. MUST contain the high-order 16 bits of the 32-bit packet sequence number. This is needed since the high data rates of VC-2 Sequences mean that it is highly likely that the 16-bit sequence number will roll over too frequently to be of use for Stream synchronization.

B: 1 bit. MUST be set to 1 if the packet contains the first byte of an Auxiliary Data Unit and otherwise MUST be 0. If the recommendations in Section 4.4 ("Stream Constraints") are followed, then every Auxiliary Data Unit will be small enough to fit in a single packet, and so this bit (where present) will always be 1.

E: 1 bit. MUST be set to 1 if the packet contains the final byte of an Auxiliary Data Unit and otherwise MUST be 0. If the recommendations in Section 4.4 ("Stream Constraints") are followed, then every Auxiliary Data Unit will be small enough to fit in a single packet, and so this bit (where present) will always be 1.

I: 1 bit. MUST be set to 1 if the packet contains coded picture parameters or slice data from a field in an interlaced frame. MUST be set to 0 if the packet contains data from any part of a progressive frame.

F: 1 bit. MUST be set to 1 if the packet contains coded picture parameters or slice data from the second field of an interlaced frame. MUST be set to 0 if the packet contains data from the first field of an interlaced frame or any part of a progressive frame.

Parse Code (PC): 8 bits. Contains a Parse Code that MUST be the value indicated for the type of data in the packet.

Data Length: 32 bits. For an auxiliary Data Unit, this contains the number of bytes of data contained in the payload section of this packet. If the recommendations in Section 4.4 ("Stream Constraints") are followed, then no Auxiliary Data Unit will be large enough to cause a packet to exceed the MTU of the network.

Picture Number: 32 bits. MUST contain the Picture Number for the coded picture this packet contains data for, as described in Section 12.1 of the VC-2 specification [VC2].

The sender MUST send at least one transform-parameters packet for each coded picture and MAY include more than one as long as they contain identical data. The sender MUST NOT send a packet from a new picture until all the coded data from the current picture has been sent.

If the receiver receives Coded Slices packets for a picture but does not receive a Transform Parameters packet for that picture, then this is an indication of either packet loss, joining a Stream mid-picture, or a non-compliant transmitter. In this case, the receiver MAY assume that the parameters are unchanged since the last picture, or it MAY discard the picture. Choosing between these two options is left up to the implementation as it will be dependent on intended use. The former may result in malformed pictures, while the latter will result in dropped frames.

Slice Prefix Bytes: 16 bits. MUST contain the Slice Prefix Bytes value for the coded picture this packet contains data for, as described in Section 12.3.4 of the VC-2 specification [VC2].

In the VC-2 specification, this value is not restricted to 16 bits, but the constraints on Streams specified in this document (Section 4.4) do require this.

Slice Size Scaler: 16 bits. MUST contain the Slice Size Scaler value for the coded picture this packet contains data for, as described in Section 12.3.4 of the VC-2 specification [VC2].

In the VC-2 specification, this value is not restricted to 16 bits, but the constraints on Streams specified in this document (Section 4.4) do require this.

Fragment Length: 16 bits. MUST contain the number of bytes of data contained in the coded payload section of this packet.

No. of Slices: 16 bits. MUST contain the number of Coded Slices contained in this packet, which MUST be 0 for a packet containing Transform Parameters. In a packet containing Coded Slices, this number MUST be the number of whole slices contained in the packet, and the packet MUST NOT contain any partial slices.

Slice Offset X: 16 bits. MUST contain the X coordinate of the first slice in this packet, in slices, starting from the top left corner of the picture.

Slice Offset Y: 16 bits. MUST contain the Y coordinate of the first slice in this packet, in slices, starting from the top left corner of the picture.

4.3. The Choice of Parse Codes (Informative)

The "PC" field in the packets is used to carry the Parse Code, which identifies the type of content in the packet. This code matches the value of the Parse Code used to identify each Data Unit in a VC-2 Stream, as defined in the VC-2 specification, and each packet contains the entire Data Unit.

Figure 7 lists all of the Parse Codes currently allowed in a VC-2 Sequence. The final column indicates whether the code in question can be present in a Stream transmitted using this specification.

PC (hex)	Binary	Description	Valid
0x00	0000 0000	Sequence Header	Yes
0x10	0001 0000	End of Sequence	Yes
0x20	0010 0000	Auxiliary Data	Yes
0x30	0011 0000	Padding Data	Yes
0xC8	1100 1000	LD Picture	No
0xE8	1110 1000	HQ Picture	No
0xEC	1110 1100	HQ Picture Fragment	Yes

Figure 7: Parse Codes and Meanings

4.4. Stream Constraints

A Sequence needs to conform to certain constraints in order to be transmissible with this specification.

- The Sequence MUST NOT contain Parse Info Headers with a Parse Code other than 0x00 (Sequence Header), 0x10 (End of Sequence), 0x20 (Auxiliary Data), 0x30 (Padding Data), or 0xEC (High Quality Picture Fragment). Some other Streams MAY be convertible to meet this restriction (see below).

- Every High Quality Picture Fragment MUST be no longer than 65535 bytes. This can usually be ensured by splitting large Fragments into several smaller Fragments, except in the case where an individual slice is too large, in which case see the notes below on conversion.

- Informative note: this requirement ensures that every High Quality Picture Fragment will always contain no more than 65535 slices.

- Every High Quality Picture Fragment SHOULD be small enough that the RTP packet carrying it will fit within the network MTU size. This can usually be ensured by splitting large Fragments into several smaller Fragments, except in the case where an individual slice is too large, in which case see the notes below on conversion.

- Every High Quality Picture Fragment MUST be encoded using values for Slice Prefix Bytes and Slice Size Scaler no greater than 65535.

If a Sequence intended for transmission does not conform to these restrictions, then it MAY be possible to simply convert it into a form that does by splitting pictures and/or large Fragments into suitably sized Fragments. This can be done provided that the following (weaker) constraints are met:

- The Sequence does not contain Parse Info Headers with a Parse Code other than 0x00 (Sequence Header), 0x10 (End of Sequence), 0x20 (Auxiliary Data), 0x30 (Padding Data), 0xE8 (High Quality Picture), or 0xEC (High Quality Picture Fragment).

- None of the High Quality Pictures or High Quality Picture Fragments contain slices that are individually longer than 65535 bytes. Note: When this is the case, the values of Slice Prefix Bytes and Slice Size Scaler will necessarily also be smaller than 65535.

- None of the High Quality Pictures or High Quality Picture Fragments contain slices that are individually so large that an RTP packet carrying a Fragment containing that single slice will fit within the network MTU size.

It is not possible to send a Stream that does not meet the above requirements via this mechanism unless the Stream is re-encoded by a VC-2 Encoder so as to meet them.

In addition, every Auxiliary Data Unit SHOULD be small enough that a single RTP packet carrying it will fit within the network MTU size. Since there is currently no specification for the format of Auxiliary Data in VC-2, the mechanism for ensuring this with an encoder implementation that includes Auxiliary Data Units will be dependent upon the implementation's use for them.

When encoding VC-2 video intended to be transported via RTP, a VC-2 profile and level that ensures these requirements are met SHOULD be used.

4.5. Payload Data

For the Sequence Header packet type (PC = 0x00), the payload data MUST be the coded Sequence Header exactly as it appears in the VC-2 Sequence.

For the Transform Parameters packet type (PC = 0xEC and No. of Slices = 0), the payload data MUST be the variable-length coded Transform Parameters. This MUST NOT include the Fragment header (since all data in the picture header is already included in the packet header).

For the Auxiliary Data packet type (PC = 0x20), the payload data MUST be a portion of the auxiliary data bytes contained in the Auxiliary Data Unit being transmitted. The B flag MUST be set on the packet that contains the first byte, the E flag MUST be set on the packet that contains the last byte, the bytes MUST be included in order, and the packets MUST have contiguous sequence numbers.

For the Picture Fragment packet type (PC = 0xEC and No. of Slices > 0), the payload data MUST be a specified number of Coded Slices in the same order that they appear in the VC-2 Stream. Which slices appear in the packet is identified using the Slice Offset X and Slice Offset Y fields in the payload header.

For the End of Sequence packet type (PC = 0x10), there is no payload data.

4.5.1. Reassembling the Data

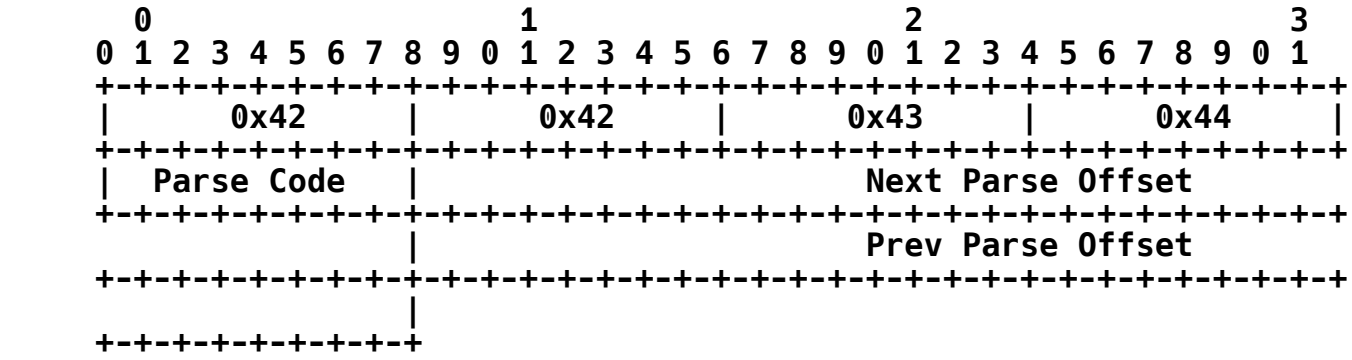

Figure 8: VC-2 Parse Info Header

To reassemble the data in the RTP packets into a valid VC-2 Sequence:

- The receiver SHOULD take the data from each packet with a Parse Code of 0x00 and prepend a valid VC-2 Parse Info Header (Figure 8) with the same Parse Code (0x00). The resulting Sequence Header Parse Info Header and Data Unit MUST be included in the output stream before any coded pictures that followed the packet being processed in the RTP stream, unless an identical Sequence Header has already been included, and they MAY be repeated (with appropriate modifications to the next and previous header offsets) at any point that results in a valid VC-2 Stream.

- The receiver SHOULD take the data from each packet with a Parse Code of 0xEC and No. of Slices set to 0 (which together indicate that this packet contains the Transform Parameters for a coded picture) and prepend with the same Parse Code a valid VC-2 Parse Info Header (Figure 8) followed by the picture number, Fragment data length, and slice count (0).

- The receiver SHOULD take the data from each packet with a Parse Code of 0xEC and No. of Slices not set to 0 (which together indicate that this packet contains Coded Slices) and prepend with the same Parse Code a valid VC-2 Parse Info Header (Figure 8) followed by the picture number, Fragment data length, slice count, x offset and y offset taken from the packet header.

- A receiver MAY combine all Fragment Data Units (with Parse Code 0xEC) and the same picture number into a single picture Data Unit with Parse Code 0xE8. If the Stream is required to comply with major versions 1 or 2 of the VC-2 specification, then this MUST be done.

- The receiver SHOULD take the data from each packet with a Parse Code of 0x20 and the B bit set and prepend a valid VC-2 Parse Info Header (Figure 8) with the Parse Code 0x20, and then take each subsequent packet with Parse Code 0x20 without the B bit set and append its payload to the growing Data Unit. When all packets for a particular Data Unit have been received, it SHOULD be included in the output stream. The final packet for a Data Unit will have the E bit set.

- Once a Data Unit has been assembled, whether a Sequence Header, Coded Picture Fragment, Coded Picture, or Auxiliary Data Unit, the next parse offset and previous parse offset values in its Parse Info Header SHOULD be filled with the offset between the start of the header and the start of the next or previous header.

- An End of Sequence Parse Info Header MAY be inserted when a packet with Parse Code set to 0x10 is encountered, or at any other time that is allowed in a valid VC-2 Stream. After an End of Sequence Parse Info Header is included in the output stream, either the Stream must end, or it MUST be followed by a Sequence Header indicating the start of a new Sequence. The next parse offset of the End of Sequence header MUST be set to 0, and the previous parse offset SHOULD be filled with the offset from the start of the previous Parse Info Header in the Stream.

- A Padding Data Parse Info Header MAY be inserted when a packet with Parse Code set to 0x30 and the B bit set is encountered, or at any other time that is allowed in a valid VC-2 Stream. The

length of the accompanying Data Unit MAY have any value, and its contents MUST be set to a series of zero bytes. The next parse offset and previous parse offset values in its Parse Info Header SHOULD be filled with the offset between the start of the header and the start of the next or previous header.

5. Forward Error Correction (FEC) Considerations

VC-2 provides no underlying protection against data loss, so it may be useful to employ Forward Error Correction to the Stream. A mechanism for doing this to a generic RTP stream is specified in RFC 5109 [RFC5109]. If making use of this mechanism to provide multilevel protection, then the packets SHOULD be assigned to layers based upon their packet type, with the packet types being, in order of importance:

1. Sequence Headers
2. Fragments containing Transform Parameters
3. Fragments containing Coded Slices
4. Auxiliary Data and end of Sequence
5. Padding

It is RECOMMENDED that if multilevel protection is to be used, then one layer will protect all Sequence Header packets, and a second will protect Sequence Headers and all Fragments. If desired, a third layer MAY protect Auxiliary Data and End of Sequence packets. Padding data SHOULD NOT be protected by FEC.

6. Congestion Control Considerations

Congestion control for RTP SHALL be used in accordance with RFC 3550 [RFC3550] and any applicable RTP profile -- e.g., RFC 3551 [RFC3551]. An additional requirement if best-effort service is being used is: users of this payload format MUST monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Circuit Breakers [RFC8083] are an update to RTP [RFC3550] that defines criteria for when one is required to stop sending RTP Packet Streams, and applications implementing this standard MUST comply with it. RFC 8085 [RFC8085] provides additional information on the best practices for applying congestion control to UDP streams.

In particular, it should be noted that the expected data rate for RTP sessions that use this profile is likely to be in the range of gigabits per second. If used on a closed network that has been correctly provisioned for the expected data rates, this might not pose a problem, but there is always the risk of data getting out onto the open internet.

7. Payload Format Parameters

This RTP payload format is identified using the 'video/vc2' media type, which is registered in accordance with RFC 4855 [RFC4855], using the template of RFC 6838 [RFC6838].

7.1. Media Type Definition

Type name:

video

Subtype name:

vc2

Required parameters:

rate: The RTP timestamp clock rate. Applications using this payload format SHOULD use a value of 90000.

profile: The VC-2 profile in use. The only value currently allowed is "HQ".

Optional parameters:

version: the VC-2 specification version in use. The only currently allowed value is "3" since all Sequences transported using this mechanism will contain HQ Picture Fragment Data Units, which the VC-2 specification [VC2] defines as requiring version 3.

level: The VC-2 level in use. Any integer may be used.

Encoding considerations:

This media type is framed and binary; see Section 4.8 in RFC 6838 [RFC6838].

Security considerations:

Please see the security considerations in RFC 8450.

Interoperability considerations: N/A

Published specification:

RFC 8450

Applications that use this media type:

Video Communication.

Fragment identifier considerations: N/A

Additional information: N/A

Person & email address to contact for further information:

James P. Weaver <james.barrett@bbc.co.uk>

Intended usage:

COMMON

Restrictions on usage:

This media type depends on RTP framing and hence is only defined for transfer via RTP [RFC3550]. Transport within other framing protocols is not defined at this time.

Author:

James P. Weaver <james.barrett@bbc.co.uk>

Change controller:

IETF PAYLOAD Working Group delegated from the IESG.

Provisional registration? (standards tree only):

No

7.2. Mapping to the Session Description Protocol (SDP)

The mapping of the above-defined payload format media type and its parameters SHALL be done according to Section 3 of RFC 4855 [RFC4855].

- The type name ("video") goes in SDP "m=" as the media name.
- The subtype name ("vc2") goes in SDP "a=rtpmap" as the encoding name, followed by a slash ("/") and the rate parameter.
- The required parameter profile and the optional parameters version and level, when present, are included in the "a=fmtp" attribute line of SDP as a semicolon-separated list of parameter=value pairs.

Version and level SHALL be specified in decimal when present.

For example, a sample SDP mapping for VC-2 could be as follows:

```
m=video 30000 RTP/AVP 112
a=rtpmap:112 vc2/90000
a=fmtp:112 profile=HQ;version=3;level=0
```

In this example, a dynamic payload type 112 is used for vc-2 data. The 90 kHz RTP timestamp rate is specified in the "a=rtpmap" line after the subtype. In the "a=fmtp:" line, profile HQ, version 3, and level 0 (unknown or non-standard level) are specified.

7.3. Offer/Answer Considerations

All parameters are declarative.

8. IANA Considerations

IANA has registered 'video/vc2' as specified in Section 7.1. The media type has been added to the IANA registry for "RTP Payload Format Media Types" <https://www.iana.org/assignments/rtp-parameters>.

9. 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 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 and important considerations in "Options for Securing RTP Sessions" [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this section discusses the security-impacting properties of the payload format itself.

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

To avoid buffer overruns when processing these packets, the receiver MUST consider both the reported Fragment length and the actual received size of a packet containing slice data.

In some cases, the transmitter may need to decode variable-length coded headers in order to extract some data from the VC-2 bitstream before assembling packets. This process is potentially subject to buffer overruns if not implemented carefully.

10. References

10.1. 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>.

[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>.

[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>.

[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>.

[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>.

[RFC8083] Perkins, C. and V. Singh, "Multimedia Congestion Control: Circuit Breakers for Unicast RTP Sessions", RFC 8083, DOI 10.17487/RFC8083, March 2017, <https://www.rfc-editor.org/info/rfc8083>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[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>.

[VC2] SMPTE, "SMPTE Standard - VC-2 Video Compression", ST 2042-1:2017, DOI 10.5594/SMPTE.ST2042-1.2017, June 2017, <https://ieeexplore.ieee.org/servlet/opac?punumber=7967894>.

10.2. Informative References

[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>.

[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>.

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

[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>.

[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>.

Author's Address

James P. Weaver
BBC

Email: james.barrett@bbc.co.uk