Workgroup:	Audio/Video Transport Core Maintenance
Internet-Draft:	draft-ietf-avtcore-rtp-j2k-scl-00
Published:	7 November 2023
Intended Status:	Standards Track
Expires:	10 May 2024
Authors:	P.-A. Lemieux, Ed.	D. S. Taubman
---	---	---
	Sandflow Consulting LLC	*University of New South Wales*

RTP Payload Format for sub-codestream latency JPEG 2000 streaming

Abstract

This RTP payload format defines the streaming of a video signal encoded as a sequence of JPEG 2000 codestreams. The format allows sub-codestream latency, such that the first RTP packet for a given codestream can be emitted before the entire codestream is available.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This RTP payload format specifies the streaming of a video signal encoded as a sequence of JPEG 2000 codestreams.

In addition to supporting a variety of frame scanning techniques (progressive, interlaced and progressive segmented frame) and image characteristics, the payload format includes the following features specifically designed for streaming applications:

- the payload format allows sub-codestream latency such that the first RTP packet of a given codestream to be emitted before the entire codestream is available. Specifically, the payload format does not rely on the JPEG 2000 PLM and PLT marker segments for recovery after RTP Packet loss since these markers can only be written after the codestream is complete and are thus incompatible with sub-codestream latency. Instead, the payload format includes payload header fields (ORDH, ORDB, POS and PID) that indicates whether the RTP packet contains a resynchronization (resync) point and how a recipient can restart codestream processing from that resync point. This contrasts with [RFC5371], which also specifies an RTP payload format for JPEG 2000, but relies on codestream structures that cannot be emitted until the entire codestream is available.

- as in [RFC4175], the payload header contains an extension (ESEQ) to the standard 16-bit RTP sequence number, enabling the payload format to accommodate high data rates without ambiguity. This is necessary as the standard sequence number will roll over very quickly for high data rates likely to be encountered in this application. For example, the standard sequence number will roll over in 0.5 seconds with a 1-Gbps video stream with RTP Packet sizes of at least 1000 octets, which can be a problem for detecting loss and out-of-order packets particularly in instances where the round-trip time is greater than the roll over period (0.5 seconds in this example).
- the payload header optionally contains a temporal offset (PTSTAMP) relative to the first RTP Packet with the same value of RTP timestamp field (Section 5.2). The higher resolution of PTSTAMP compared to the timestamp allows receivers to recover the sender's clock more rapidly.

Finally, the payload format also makes use of the unique scalability features of JPEG 2000 to allow a network agent or recipient to discard resolutions and/or quality layers merely by inspecting payload headers (QUAL and RES fields), without having to parse the underlying codestream.

2. Requirements Language

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 following summarizes the structure of the JPEG 2000 codestream, which is specified in detail at [jpeg2000-1].

NOTE: as described at Section 6, a JPEG 2000 codestream allows capabilities defined in any part of the JPEG 2000 family of standards, including those specified in [jpeg2000-2] and [jpeg2000-15].

JPEG 2000 represents an image as one or more components, e.g., R, G and B, each uniformly sampled on a common rectangular reference grid. An image can be further divided into contiguous rectangular tiles that are each independently coded and decoded.

NOTE: This payload format allows the transmission of multi-tile images as multiple single-tile images per stream, as specified at Section 6.2.

JPEG 2000 codes each image as a standalone codestream. Each codestream consists of (i) marker segments, which contain coding parameters and metadata, and (ii) coded data.

The codestream starts with an SOC marker segment and ends with an EOC marker segment. The main header of the codestream consists of marker segments between the SOC and first SOT marker segment and contains information that applies to the codestream in its entirety. It is generally impossible to decode a codestream without its main header.

The rest of the codestream consists of additional marker segments (tile-part headers) interleaved with coded image data.

The coded image data ultimately consists of code-blocks, each containing coded samples belonging to a rectangular (spatial) region within one resolution level of one component. Code-blocks are further collected into precincts, which, accordingly, represents code-blocks belonging to a spatial region within one resolution level of one component.

The image coded data can be arranged into several progression orders, which dictates which aspect of the image appears first in the codestream (in terms of byte offset). The progression orders are parameterized according to:

Position (P) The first lines of the image come before the last lines of the image.

Component (C) The first component of the image come before the last component of the image.

Resolution Layer (R) The information needed to reconstruct the lower spatial resolutions of the image come before the information needed to reconstruct the higher spatial resolutions of the image.

Quality Layer (L) The information needed to reconstruct the most-significant bits of each sample come before the information needed to reconstruct the least-significant bit of each sample.

For example, in the PRCL progression order, the information needed to reconstruct the first lines of the image come before that needed to reconstruct the last lines of the image and, within a collection of lines, the information needed to reconstruct the lower spatial resolutions of the image come before the information needed to reconstruct the higher spatial resolutions. This progression order is particular useful for sub-frame latency operations.

4. Video signal description

This RTP payload format supports three distinct video frame scanning techniques:

- Progressive frame
- Interlaced frame, where each frame consists of two fields. Field 1 occurs temporarily before Field 2. The height in lines of each field is half the height of the image.
- Progressive segmented frame (PsF), where each frame consists of two segments. Segment 1 contains the odd lines (1, 3, 5, 7,...) of a frame and Segment 2 contains the even lines (2, 4, 6, 8,...) of the same frame, where lines from the top of the frame to the bottom of the frame are numbered sequentially starting at 1.

All frames are scanned left to right, top to bottom.

5. Payload Format

5.1. General

```
         <----------- Codestream (image) --------->
         |                              |
         < Extended Header >                       |
         |             |                |
         +-----+-----+-----+-----------//--+-----+-----+---------
         | SOC | ... | SOD | ............. | EOC |  P  | SOC  ...
         +-----+-----+-----+-----------//--+-----+-----+---------
         |                                  |
         |                                  |
         |                                  |
         +-------------------+------+-//--+-----------+---------
 Packets |        Main       | Body | ... |    Body   | Main ...
         +-------------------+------+-//--+-----------+---------
```

Figure 1: Packetization of a sequence of JPEG 2000 codestreams (not to scale). P are arbitrary padding bytes.

Each RTP packet, as specified at [RFC3550], is either a Main Packet or a Body Packet.

A Main Packet consists of the following ordered sequence of structures concatenated without gaps:

- the RTP Fixed Header;
- a Main Packet Payload Header, as specified at Section 5.3; and
- the payload, which consists of a JPEG 2000 codestream fragment.

A Body Packet consists of the following ordered sequence of structures concatenated without gaps:

- the RTP Fixed Header;
- a Body Packet Payload Header, as specified at Section 5.4; and
- the payload, which consists of a JPEG 2000 codestream fragment.

When concatenated, the sequence of JPEG 2000 codestream fragments emitted by the sender MUST be a sequence of JPEG 2000 codestreams where two successive JPEG 2000 codestreams MAY be separated by one or more arbitrary padding bytes.

The JPEG 2000 codestreams MUST conform to Section 6.

The padding bytes MUST be ignored by the recipient.

NOTE: Padding bytes can be used to achieve constant bit rate transmission.

A JPEG 2000 codestream consists of the bytes between, and including, the SOC and EOC markers, as defined in [jpeg2000-1].

A JPEG 2000 codestream fragment does not necessarily contain complete JPEG 2000 packets, as defined in [jpeg2000-1].

A JPEG 2000 codestream Extended Header consists of the bytes between, and including, the SOC marker and the first SOD marker.

The payload of a Body Packet MUST NOT contain any bytes of the JPEG 2000 codestream Extended Header.

The payload of a Main Packet MUST contain at least one byte of the JPEG 2000 codestream Extended Header and MAY contain bytes other than those of the JPEG 2000 codestream Extended Header.

A payload MUST NOT contain bytes from more than one JPEG 2000 codestream.

5.2. RTP Fixed Header Usage

The following RTP header fields have a specific meaning in the context of this payload format:

marker

- 1 The payload contains an EOC marker.
- 0 Otherwise

timestamp

The `timestamp` is the presentation time of the image to which the payload belongs.

The `timestamp` clock rate is 90 kHz.

The timestamp of successive progressive frames MUST advance at regular increments based on the instantaneous video frame rate.

The `timestamp` of Field 1 of successive interlaced frames MUST advance at regular increments based on the instantaneous video frame rate, and the `Timestamp` of Field 2 MUST be offset from the `timestamp` of Field 1 by one half of the instantaneous frame period.

The `timestamp` of both segments of a progressive segmented frame MUST be equal.

`timestamp` of all RTP packets of a given image MUST be equal.

sequence number

The low-order bits of the RTP sequence number. The higher order bits of the RTP sequence number are contained in the ESEQ field.

5.3. Main Packet Payload Header

Figure 1 specifies the structure of the payload header. Fields are interpreted as unsigned binary integers in network order.

```
MH .............................. (2)
TP .............................. (3)
ORDH ............................ (3)
P ............................... (1)
XTRAC ........................... (3)
PTSTAMP ........................ (12)
ESEQ ............................ (8)
R ............................... (1)
S ............................... (1)
C ............................... (1)
RSVD ............................ (4)
RANGE ........................... (1)
PRIMS ........................... (8)
TRANS ........................... (8)
MAT ............................. (8)
XTRAB .................. (XTRAC * 32)
```

Figure 2: Structure of the Main Packet Payload Header

MH

0
: The RTP Packet is a Body Packet.

1
: The RTP Packet is a Main Packet and the codestream has more than one Main Packet. The next RTP Packet is a Main Packet.

2
: The RTP Packet is a Main Packet and the codestream has more than one Main Packet. The next RTP Packet is a Body Packet.

3
: The RTP Packet is a Main Packet and the codestream has exactly one Main Packet.

TP

Indicates the scanning structure of the image to which the payload belongs.

0
: Progressive frame.

1
: Field 1 of an interlaced frame, where the first line of the field is the first line of the frame.

2
: Field 2 of an interlaced frame, where the first line of the field is the second line of the frame.

3
: Field 1 of an interlaced frame, where the first line of the field is the second line of the frame.

4
: Field 2 of an interlaced frame, where the first line of the field is the first line of the frame.

5
: Segment 1 of a progressive segmented frame, where the first line of the image is the first line of the frame.

6
: Segment 2 of a progressive segmented frame, where the first line of the image is the second line of the frame.

7
: Extension value. See Section 8.6 and Section 7.8.

ORDH
: Specifies the progression order used by the codestream and whether resync points are signaled.

0
: Resync points are not necessarily signaled. The progression order can vary over the codestream.

1
: The progression order is LRCP for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

2
: The progression order is RLCP for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

3
: The progression order is RPCL for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

4
: The progression order is PCRL for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

5
: The progression order is CPRL for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

6
: The progression order is PRCL for the entire codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

7
: The progression order can vary over the codestream. The first resync point is specified in every Body Packet that contains one or more resync points.

`ORDH` MUST be 0 is the codestream consists of more than one tile.

NOTE: Only `ORDH` = 4 and `ORDH` = 6 allow sub-codestream latency streaming.

NOTE: Progression order PRCL is defined in [jpeg2000-2]. The other progression orders are specified in [jpeg2000-1].

P

0

PTSTAMP is not used.

1

PTSTAMP is used.

XTRAC

Length, in multiples of 4 bytes, of the XTRAB field.

PTSTAMP

PTSTAMP = (timestamp + TOFF) mod 4096, if P = 1 in the Main Packet of this codestream.

TOFF is the transmission time of this RTP Packet, in the timebase of the timestamp clock and relative to the first packet with the same timestamp value.

TOFF = 0 in the first RTP Packet with the same timestamp value.

PTSTAMP = 0, if P = 0 in the Main Packet of this codestream.

ESEQ

The high order bits of the extended sequence number.

R

Determines whether Main Packet and codestream header information can be reused across codestreams.

1

All Main Packets in this stream, as identified by its SSRC value:

- MUST have identical Main Packet Payload Headers, with the exception of their TP, MH, ESEQ and PTSTAMP fields;
- MUST contain the same codestream main header information, with the exception of the SOT and COM marker segments, and any pointer marker segments; and
- MUST NOT contain bytes other than Extended Header bytes.

0

Otherwise

S

0

Component colorimetry is not specified, and left to the session or the application.

PRIMS, TRANS and MAT and RANGE MUST be zero.

1

Component colorimetry is specified by the PRIMS, TRANS and MAT and RANGE fields.

The codestream components MUST conform to one of the combinations at Table 1.

Combination name	Component index			
	0	**1**	**2**	**3**
Y	Y			
YA	Y	A		
RGB	R	G	B	
RGBA	R	G	B	A
YCbCr	Y	C_B	C_R	
YCbCrA	Y	C_B	C_R	A
The channel A is an opacity channel. The minimum sample value (0) indicates a completely transparent sample, and the maximum sample value (as determined by the bit depth of the codestream component) indicates a completely opaque sample. The opacity channel MUST map to a component with unsigned samples.				

Table 1: Mapping of codestream components to color channels

C
: 0
: Code-block caching is not in use.

: 1
: Code-block caching is in use.

: R MUST be equal to 1.

RSVD
: Reserved value. See Section 8.5 and Section 7.7.

RANGE
: Value of the VideoFullRangeFlag specified in [rec-itu-t-h273]

PRIMS
: One of the ColourPrimaries values specified in [rec-itu-t-h273]

TRANS
: One of the TransferCharacteristics values specified in [rec-itu-t-h273]

MAT
: One of the MatrixCoefficients values specified in [rec-itu-t-h273]

XTRAB
: Allows the contents of the Main Packet Payload Header to be extended in the future. See Section 8.4 and Section 7.6.

5.4. Body Packet Payload Header

Figure 3 specifies the structure of the Body Packet Payload Header. Fields are interpreted as unsigned binary integers in network order.

```
MH ............................... (2)
TP ............................... (3)
RES .............................. (3)
ORDB ............................. (1)
QUAL ............................. (3)
PTSTAMP ......................... (12)
ESEQ ............................. (8)
POS ............................. (12)
PID ............................. (20)
```

Figure 3: Structure of the Body Packet Payload Header

RES

0
: The payload can contribute to all resolution layers.

Otherwise
: The payload contains at least one byte of one JPEG 2000 packet belonging to resolution level (N_L + RES - 7) but does not contain any byte of any JPEG 2000 packet belonging to lower resolution levels. N_L is the number of decomposition levels of the codestream.

ORDB

0
: No resync point is specified for the payload.

1
: The payload contains a resync point.

`ORDB` MUST be 0 is the codestream consists of more than one tile.

QUAL

0
: The payload can contribute to all quality layers.

Otherwise
: The payload contributes only to quality layer index `QUAL` or above.

POS

Byte offset from the start of the payload to the first byte of the resync point belonging to the precinct identified by PID.

`POS` MUST be 0 if `ORDB` = 0.

PID

Unique identifier of the precinct of the resync point.

`PID = c + s * num_components`

where:

- *c* is the index (starting from 0) of the image component to which the precinct belongs;
- *s* is a sequence number which identifies the precinct within its tile-component; and
- *num_components* is the number of components of the codestream.

If PID is present, the payload MUST NOT contain codestream bytes from more than one precinct.

PID MUST be 0 if ORDB = 0.

NOTE: PID is identical to precinct identifier I specified in [jpeg2000-9].

6. JPEG 2000 codestream requirements

6.1. General

The JPEG 2000 codestream MAY include capabilities beyond those specified at [jpeg2000-1], including those specified in [jpeg2000-2] and [jpeg2000-15].

NOTE: The Rsiz parameter and CAP marker segments of each JPEG 2000 codestream contain detailed information on the capabilities necessary to decode the codestream.

NOTE: The caps media type parameter defined in Section 9.2 allows applications to signal required device capabilities.

NOTE: The block coder specified at [jpeg2000-15] improves throughput and reduces latency compared to the original arithmetic block coder defined in [jpeg2000-1].

For interlaced or progressive segmented frames, the height specified in the JPEG 2000 main header MUST be the height in lines of the field or the segment, respectively.

If any decomposition level involves only horizontal decomposition then no decomposition level MUST involve only vertical decomposition; and conversely, if any decomposition level involves only vertical decomposition then no decomposition level MUST involve only horizontal decomposition.

6.2. Transmitting multi-tile images as multiple single-tile images

A sequence of multi-tile images can be transmitted by splitting it into multiple sequences of single-tile images, where:

- each sequence of single-tile images corresponds to a unique tile of the multi-tile image;
- each sequence of single-tile images is transmitted in a separate RTP stream;
- the coordinates of each single-tile image are expressed using the coordinate system of the multi-tile image; and
- the bitstreams of each of each single-tile image are identical to the corresponding bitstreams in the multi-tile image.

Such sequences of single-tile images are identified using the tile media type parameter specified at Section 9.2.

7. Sender requirements

7.1. Main Packet

Only Main Packets MAY contain bytes of the JPEG 2000 codestream Extended Header.

The sender MUST either emit a single Main Packet with MH = 3, or one or more Main Packets with MH = 1 followed by a single Main Packet with MH = 2.

The Main Packet Payload Headers fields MUST be identical in all Main Packet of a given codestream, with the exception of:

- MH;
- ESEQ; and
- PTSTAMP.

7.2. RTP Packet filtering

A network agent MAY strip out RTP Packet from a codestream that are of no interest to a particular client, e.g., based on a resolution or a spatial region of interest. Such a network agent SHOULD include a CSRC identifier to identify the SSRC field of the original source from which content was stripped.

7.3. Resync point

A resync point is the first byte of JPEG 2000 packet header data for a precinct and for which PID < 2^{24}.

NOTE: Resync points cannot be specified if the codestream consists of more than one tile (ORDB and ORDH are both equal to zero). To transmit codestreams that consist of more than one tile and benefit from resync points, the technique specified at Section 6.2 can be used.

NOTE: A resync point can be used by a receiver to process a codestream even if earlier packets in the codestream have been corrupted, lost or deliberately discarded by a network agent. As a corollary, resync points can be used by a network agent to discard packets that are not relevant to a given rendering resolution or region of interest. Resync points play a role similar to pointer marker segments, albeit tailored for high bandwidth low latency streaming applications.

7.4. PTSTAMP field

A sender SHOULD set P = 1, but only if it can generate PTSTAMP accurately.

PTSTAMP can be derived from the same clock that is used to produce the 32-bit timestamp field in the RTP fixed header. Specifically, a sender maintains, at least conceptually, a 32-bit counter that is incremented by a 90kHz clock. The counter is sampled at the point when each RTP Packet is or SHOULD be at least notionally transmitted and the 12 LSBs of the sample are stored in the PTSTAMP field.

If PTSTAMP = 1, then the transmission time TOFF (as defined at Section 5.3) for two consecutive RTP packets with identical timestamp fields MUST NOT differ by more than 4095.

7.5. RES field

A sender SHOULD set RES > 0 whenever possible.

NOTE: While a sender can always safely set RES = 0, this makes it more difficult to discard packets based on resolution, as described at Section 8.3.

7.6. Extra information

The sender MUST set the value of XTRAC to 0.

Future edition of this specification can permit other values.

7.7. Reserved values

The sender MUST set reserved values to 0.

Future edition of this specification can specify other values such that these values can be ignored by receivers that conform to this specification.

7.8. Extension values

A sender MUST NOT use an extension value.

7.9. Code-block caching

This section applies only if C = 1.

A sender can improve bandwidth efficiency by only occasionally transmitting code-blocks corresponding to static portions of the video and otherwise transmitting empty code-blocks. When C = 1, and as described at Section 8.7, a receiver maintains a simple cache of previously received code-blocks, which it uses to replace empty code-blocks.

A sender alone determines which and when code-blocks are replaced with empty code-blocks.

The sender cannot however determine with certainty the state of the receiver's cache: some code-blocks might have been lost in transit, the sender doesn't know exactly when the receiver started processing the stream, etc.

A code-block is *empty* if:

- it does not contribute code-bytes as specified in the parent JPEG 2000 packet header; or
- if the code-block conforms to [jpeg2000-15], contains an HT cleanup segment and the first two bytes of the Magsgn byte-stream are between 0xFF80 and 0xFF8F.

NOTE: the last condition allows the encoder to insert padding bytes to achieve a constant bit rate even when code-block does not contribute code-bytes, as suggested at [jpeg2000-15], F.4.

8. Receiver

8.1. PTSTAMP

Receivers can use PTSTAMP values to accelerate sender clock recovery since PTSTAMP typically updates more regularly than timestamp.

8.2. QUAL

A receiver can discard packets where QUAL > N if it is interested in reconstructing an image that only incorporates quality layers N and below.

8.3. RES

The JPEG 2000 coding process decomposes an image using a sequence of discrete wavelet transforms (DWT) stages.

Decomposition level	Resolution level	Subbands	Keep all Body Packets with RES equal to or less than this value...	... to decode an image with at most these dimensions
1	5	HL1,LH1,HH1	7	W x H
2	4	HL2,LH2,HH2	6	(W/2) x (H/2)
3	3	HL3,LH3,HH3	5	(W/4) x (H/4)
4	2	HL4,LH4,HH4	4	(W/8) x (H/8)
5	1	HL5,LH5,HH5	3	(W/16) x (H/16)
5	0	LL5	2	(W/32) x (H/32)

Table 2: Optional discarding of Body Packets based on the value of the RES field when decoding a reduced resolution image, in the case where N_L = 5 and all DWT stages consist of both horizontal and vertical transforms. The image has nominal width and height of W x H.

Table 2 illustrates the case where each DWT stage consists of both horizontal and vertical transforms, which is the only mode supported in [jpeg2000-1]. The first stage transforms the image into (i) the image at half-resolution (LL1 sub-bands) and (ii) residual high-frequency data (HH1, LH1, HL1 sub-bands). The second stage transforms the image at half-resolution (LL1 sub-bands) into the image at quarter resolution (LL2 sub-bands) and residual high-frequency data (HH2, LH2, HL2 sub-bands). This process is repeated N_L times, where N_L is the number of decomposition levels as defined in the COD and COC marker segments of the codestream.

The decoding process reconstructs the image by reversing the coding process, starting with the lowest resolution image stored in the codestream (LL_{N_L}).

As a result, it is possible to reconstruct a lower resolution of the image by stopping the decoding process at a selected stage. For example, in order to reconstruct the image at quarter resolution (LL2), only sub-bands with index greater than 2, e.g., HL3, LH3, HH3, HL4, LH4, HH4, etc., are necessary. In other words, a receiver that wishes to reconstruct an image at quarter resolution could discard all packets where RES >= 6 since those packets can only contribute to HL1, LH1, HH1, HL2, LH2 and HH2 sub-bands.

In the case where all DWT stages consist of both horizontal and vertical transforms, the maximum decodable resolution is reduced by a factor of $2^{7 - N}$ if all Body Packets where RES > N are discarded.

Decomposition level	Resolution level	Subbands	Keep all Body Packets with RES equal to or less than this value...	... to decode an image with at most these dimensions
1	5	HL1,LH1,HH1	7	W x H
2	4	HL2,LH2,HH2	6	(W/2) x (H/2)
3	3	HX3	5	(W/4) x (H/2)
4	2	HX4	4	(W/8) x (H/2)
5	1	HX5	3	(W/16) x (H/2)
5	0	LX5	2	(W/32) x (H/2)

Table 3: Optional discarding of Body Packets based on the value of the RES field when decoding a reduced resolution image, in the case where N_L = 5 and some DWT stages consist of only horizontal transforms. The image has nominal width and height of W x H.

Table 3 illustrates the case where some of DWT stage consist of only horizontal transforms, as specified at Annex F of [jpeg2000-2].

A receiver can therefore discard all Body Packets where RES is greater than some threshold value if it is interested in decoding an image with its resolution reduced by a factor determined by the threshold value, as illustrated in Table 2 and Table 3.

8.4. Extra information

The receiver MUST accept values XTRAC other than 0 and MUST ignore the value of XTRAB, whose length is given by XTRAC.

Future edition of this specification can specify XTRAB contents such that this content can be ignored by receivers that conform to this specification.

8.5. Reserved values

The receiver MUST ignore the value of reserved values.

8.6. Extension values

The receiver MUST discard an RTP packet that contains any extension value.

8.7. Code-block caching

This section applies only if C = 1.

When C = 1, and as specified in Section 7.9, the sender can improve bandwidth efficiency by only occasionally transmitting code-blocks corresponding to static portions of the video and otherwise transmitting empty code-blocks, as defined at Section 7.9.

When decoding a codestream, and for each code-block in the codestream:

- if the code-block in the codestream is empty, the receiver MUST replace it with a matching code-block from the cache, if one exists; or
- if the code-block in the codestream is not empty, the receiver MUST replace any matching code-block from the cache with the code-block in the codestream.

Two code-blocks are *matching* if the following characteristics are identical for both: spatial coordinates, resolution level, component, sub-band and value of the TP field of the parent RTP packet.

9. Media Type

9.1. General

This RTP payload format is identified using the media type defined at Section 9.2, which is registered in accordance with [RFC4855] and using the template of [RFC6838].

9.2. Definition

Type name
    video

Subtype name
    jpeg2000-scl

Required parameters
    None

Optional parameters
    pixel
        Specifies the pixel format used by the video sequence.

        The parameter MUST be a URI-reference as specified in [RFC3986].

If the parameter is a relative-ref as specified in [RFC3986], then it MUST be equal to one of the pixel formats specified in Table 5 and the RTP header and payload MUST conform with the characteristics of that pixel format.

If the parameter is not a relative-ref, the specification of the pixel format is left to the application that defined the URI.

If the parameter is not specified, the pixel format is unspecified.

sample
: Specifies the format of the samples in each component of the codestream.

The parameter MUST be a URI-reference as specified in [RFC3986].

If the parameter is a relative-ref as specified in [RFC3986], then it MUST be equal to one of the formats specified in Appendix C and the stream MUST conform with the characteristics of that format.

If the parameter is not a relative-ref, the specification of the sample format is left to the application that defined the URI.

If the parameter is not specified, the sample format is unspecified.

width
: Maximum width in pixels of each image. Integer between 0 and 4,294,967,295.

The parameter MUST be a sequence of 1 or more digits.

If the parameter is not specified, the maximum width is unspecified.

height
: Maximum height in pixels of each image. Integer between 0 and 4,294,967,295.

The parameter MUST be a sequence of 1 or more digits.

If the parameter is not specified, the maximum height is unspecified.

signal
: Specifies the sequence of image types.

The parameter MUST be a URI-reference as specified in [RFC3986].

If the parameter is a relative-ref as specified in [RFC3986], then it MUST be equal to one of the signal formats specified in Appendix B and the image sequence MUST conform to that signal format.

If the parameter is not a relative-ref, the specification of the pixel format is left to the application that defined the URI.

If the parameter is not specified, the stream consists of an arbitrary sequence of image types.

caps
: The parameters contains a list of sets of constraints to which the stream conforms, with each set of constraints identified using an absolute-URI defined by an application.

The parameter MUST conform to the uri-list syntax expressed using ABNF ([RFC5234]):

```
uri-list = absolute-URI *(";" absolute-URI)
```

Each absolute-URI MUST NOT contain any ";" character.

The application that defines the absolute-URI MUST associate it with a set of constraints to which the stream conforms. Such constraints can, for example, include the maximum height and width of images.

If the parameter is not specified, constraints, beyond those specified in this document, are unspecified.

tile

The parameter MUST conform to the tile syntax expressed using ABNF ([RFC5234]):

```
tile       = tile-index "&" image-siz
tile-index  = %x31-39 *%x30-39
image-siz   = 1*HEXDIG
```

If the tile parameter is present, each image MUST correspond to one tile of a multi-tile image, as defined in Section 6.2.

tile-index is the index of the tile in the multi-tile image.

image-siz contains the SIZ marker segment parameters of the multi-tile image, encoded as a case insensitive hexadecimal string.

The SIZ parameters of each single-tile image MUST conform to the following:

- Xsiz MUST be equal to the smaller of (i) the coordinate of the right edge of tile index tile-index in the multi-tile image and (ii) Xsiz of the multi-tile image.
- Ysiz MUST be equal to the smaller of (i) the coordinate of the bottom edge of tile index tile-index in the multi-tile image and (ii) Ysiz of the multi-tile image.
- XOsiz MUST be equal to the larger of (i) the coordinate of the left edge of tile index tile-index in the multi-tile image and (ii) XOsiz of the multi-tile image.
- YOsiz MUST be equal to the larger of (i) the coordinate of the top edge of tile index tile-index in the multi-tile image and (ii) YOsiz of the multi-tile image.
- XTOsiz MUST be equal to the coordinate of the left edge of tile index tile-index in the multi-tile image.
- YTOsiz MUST be equal to the coordinate of the top edge of tile index tile-index in the multi-tile image.
- All other parameters MUST be equal to that in the multi-tile image.

Figure 4 illustrates an example where a multi-tile image that consists of two tiles is transmitted as two single-tile images (images 1 and 2). Figure 5 and Table 4 describe the tile and SIZ parameter values, respectively.

```
(a) Multi-tile image

<----------Xsiz---------->
<----> XTOSiz
+-----------------------+
|    <---Xtsiz---->      |
|    xxxxxxxxxxxxxxxxxxxxxxxxxxxx
|    x T0         x T1   |     x
|    x     #######x#######     x
|    x     #......x......#     x
|    x     #......x......#     x
+----xxxxxxxxxxxxxxxxxxxxxxxxxxxx
<---XOsiz-->

(b) Single-tile image 1

<-------Xsiz------>     +--+
<----> XTOSiz           |  | Reference grid
+-----------------+     +--+
|    <---Xtsiz---->
|    xxxxxxxxxxxxxx     xxxx
|    x T0         x     x  x Tile area
|    x     #######x     xxxx
|    x     #......x
|    x     #......x     ####
+----xxxxxxxxxxxxxx     #..# Image area
<---XOsiz-->            ####

(a) Single-tile image 2

<----------Xsiz---------->
<-----XTOSiz------>
+-----------------------+
|                  |
|          xxxxxxxxxxxxx
|          x T0   |    x
|          x#######    x
|          x......#    x
|          x......#    x
+----------------xxxxxxxxxxxxx
<-------XOsiz-----X---Xtsiz--->
```

Figure 4: Ssiz parameters for a multi-tile image and two corresponding single-tile images 1 and 2.

```
NOTE: '\' line wrapping per RFC 8792

tile=0&00000190000000c8000000c800000000000000\
c8000000c800000064000000000001080000
```

Figure 5: Example tile parameter for a multi-tile image that consists of two tiles.

SIZ parameter	Two-tile image	Image where tile-index = 1	Image where tile-index = 2
XSiz	400	300	400
YSiz	200	200	200
XTSiz	200	200	200
YTSiz	200	200	200
XOSiz	200	200	300
YOSiz	0	0	0
XTOSiz	100	100	300
YTOSiz	0	0	0

Table 4: Selected SIZ parameters for a two-tile image and two corresponding single-time images.

cache
: The value of the parameter MUST be either `false` or `true`.

  If the parameter is `true`, the field `C` MAY be 0 or 1; otherwise the field `C` MUST be 0.

  If the parameter is not specified, then the parameter is equal to `false`.

Encoding considerations
: This media type is framed and binary, see Section 4.8 of [RFC6838].

Security considerations
: See Section 12.

Interoperability considerations
: The RTP stream is a sequence of JPEG 2000 images. An implementation that conforms to the family of JPEG 2000 standards can decode and attempt to display each image.

Published specification
: This document

Applications that use this media type
: video streaming and communication

Person and email address to contact for further information
: Pierre-Anthony Lemieux <pal@sandflow.com>

Intended usage
: COMMON

Restrictions on Usage
: This media type depends on RTP framing, and hence is only defined for use with RTP as specified at [RFC3550]. Transport within other framing protocols is not defined at the time.

Author
: Pierre-Anthony Lemieux

Change controller
: IETF Audio/Video Transport Core Maintenance Working Group delegated from the IESG.

10. Mapping to the Session Description Protocol (SDP)

The mapping of the payload format media type and its parameters to SDP, as specified in [RFC8866] MUST be done according to Section 3 of [RFC4855].

11. IANA Considerations

This memo requests that IANA registers the content type specified at Section 9. The media type is also requested to be added to the IANA registry for RTP Payload Format MIME types.

12. Security considerations

RTP packets using the payload format specified in this document are subject to the security considerations discussed in [RFC3550] , and in any applicable RTP profile such as [RFC3551], [RFC4585], [RFC3711], [RFC5124]. However, as [RFC7202] discusses, it is not an RTP payload format's responsibility to discuss or mandate what solutions are used to meet the basic security goals like confidentiality, integrity, and source authenticity for RTP in general. This responsibility lays on anyone using RTP in an application. They can find guidance on available security mechanisms and important considerations in [RFC7201]. Applications SHOULD use one or more appropriate strong security mechanisms. The rest of this Security Considerations 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 RTP 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.

Security considerations related to the JPEG 2000 codestream contained in the payload are discussed at Section 3 of [RFC3745].

13. References

13.1. Normative References

[jpeg2000-1] ITU-T, "Recommendation ITU-T T.800, JPEG 2000 image coding system: Core coding system", June 2019.

[jpeg2000-2] ITU-T, "Recommendation ITU-T T.801, JPEG 2000 image coding system: Extensions", June 2021.

[jpeg2000-15] ITU-T, "Recommendation ITU-T T.814, JPEG 2000 image coding system: High-throughput JPEG 2000", June 2019.

[rec-itu-t-h273] ITU-T, "Recommendation ITU-T H.273, Coding-independent code points for video signal type identification", July 2021.

[jpeg2000-9] ITU-T, "JPEG 2000 image coding system: Interactivity tools, APIs and protocols", January 2005.

[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>.

[RFC8866] Begen, A., Kyzivat, P., Perkins, C., and M. Handley, "SDP: Session Description Protocol", RFC 8866, DOI 10.17487/RFC8866, January 2021, <https://www.rfc-editor.org/info/rfc8866>.

[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>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <https://www.rfc-editor.org/info/rfc5234>.

[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>.

[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>.

13.2. Informative References

[RFC5371] Futemma, S., Itakura, E., and A. Leung, "RTP Payload Format for JPEG 2000 Video Streams", RFC 5371, DOI 10.17487/RFC5371, October 2008, <https://www.rfc-editor.org/info/rfc5371>.

[RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, DOI 10.17487/RFC4175, September 2005, <https://www.rfc-editor.org/info/rfc4175>.

[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>.

[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>.

[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>.

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

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

[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>.

[RFC3745] Singer, D., Clark, R., and D. Lee, "MIME Type Registrations for JPEG 2000 (ISO/IEC 15444)", RFC 3745, DOI 10.17487/RFC3745, April 2004, <https://www.rfc-editor.org/info/rfc3745>.

Appendix A. Pixel formats

Table 5 defines pixel formats.

NAME	SAMP	COMPS	TRANS	PRIMS	MAT	VFR	Mapping in Table 1
rgb444sdr	4:4:4	RGB	1	1	0	0, 1	RGB
rgb444wcg	4:4:4	RGB	1	9	0	0, 1	RGB
rgb444pq	4:4:4	RGB	16	9	0	0, 1	RGB
rgb444hlg	4:4:4	RGB	18	9	0	0, 1	RGB
ycbcr420sdr	4:2:0	YCbCr	1	1	1	0	YCbCr

NAME	SAMP	COMPS	TRANS	PRIMS	MAT	VFR	Mapping in Table 1
ycbcr422sdr	4:2:2	YCbCr	1	1	1	0	YCbCr
ycbcr422wcg	4:2:2	YCbCr	1	9	9	0	YCbCr
ycbcr422pq	4:2:2	YCbCr	16	9	9	0	YCbCr
ycbcr422hlg	4:2:2	YCbCr	18	9	9	0	YCbCr

Table 5: Defined pixel formats

Each pixel format is characterized by the following:

NAME
: Identifies the pixel format

COMPS
: RGB Each codestream contains exactly three components, associated with the R, G and B color channels, in order.

  YCbCr Each codestream contains exactly three components, associated with the Y, C_b and C_r color channels, in order.

SAMP
: 4:2:0 The C_b and C_r color channels are subsampled horizontally and vertically by 1/2.

  4:2:2 The C_b and C_r color channels are subsampled horizontally by 1/2.

  4:4:4 No color channels are sub-sampled.

TRANS
: Identifies the transfer characteristics allowed by the pixel format, as defined at [rec-itu-t-h273]

PRIMS
: Identifies the color primaries allowed by the pixel format, as defined at [rec-itu-t-h273]

MAT
: Identifies the matrix coefficients allowed by the pixel format, as defined at [rec-itu-t-h273]

VFR
: Allows values of the VideoFullRangeFlag defined at [rec-itu-t-h273]

Appendix B. Signal formats

prog
: The stream MUST only consist of a sequence of progressive frames.

psf
: Progressive segmented frame (PsF) stream. The stream MUST only consist of an alternating sequence of first segment and second segment.

tff
: Interlaced stream. The stream MUST only consist of an alternating sequence of first field and second field, where the first line of the first field is the first line of the frame.

bff
: Interlaced stream. The stream MUST only consist of an alternating sequence of first field and second field, where the first line of the first field is the second line of the frame.

Appendix C. Sample formats

8
: All components consist of unsigned 8-bit integer samples.

10
: All components consist of unsigned 10-bit integer samples.

12
: All components consist of unsigned 12-bit integer samples.

16
: All components consist of unsigned 16-bit integer samples.

Appendix D. Summary of Changes (Informative)

D.1. Introduction

This Appendix summarizes substantive changes across revisions of this specification. This summary is informative and not intended to be exhaustive.

D.2. Changes from draft-ietf-avtcore-rtp-j2k-scl-00

- Allow multi-tile images in a single stream, in addition to allowing multi-tile images to be transmitted as multiple single-tile streams, as specified at Section 6.2.
- Fix incorrect TRANS values at Table 5.

Authors' Addresses

Pierre-Anthony Lemieux (editor)
Sandflow Consulting LLC
San Mateo, CA
United States of America
Email: pal@sandflow.com

David Scott Taubman
University of New South Wales
Sydney
Australia
Email: d.taubman@unsw.edu.au