Stream: Internet Engineering Task Force (IETF)
RFC: 9828
Category: Standards Track
Published: August 2025
ISSN: 2070-1721
Authors: P.-A. Lemieux, Ed. *Sandflow Consulting LLC* D. Taubman *University of New South Wales*

RFC 9828
RTP Payload Format for JPEG 2000 Streaming with Sub-Codestream Latency

Abstract

This document defines the RTP payload format for 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 image can be emitted before the entire image is available to or encoded by the sender.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The Real-time Transport Protocol (RTP), which is specified in [RFC3550], provides end-to-end network transport functions for transmitting real-time data but does not define the characteristics of the data itself (the payload), which varies across applications and is defined in companion RTP payload format documents.

This RTP payload format specifies the streaming of a video signal encoded as a sequence of JPEG 2000 codestreams (see Section 3 for a primer on the structure of JPEG 2000 codestreams). JPEG 2000 is a flexible image codec that supports resolution and quality scalability, lossy-to-lossless coding, non-iterative optimal rate control, and high dynamic range, multi-channel, and subsampled images. These features have made it a mainstay in high-performance applications, including medical, geospatial, archival, cinema, studio post-production, and TV production.

In addition to supporting a variety of frame-scanning techniques (progressive, interlaced, and progressive segmented frame) and image characteristics, the payload format supports real-time image transmission (live streaming), where image content is encoded, transmitted, and decoded continuously as it is being produced, with minimal latency. Target applications include real-time TV production over IP [OV2110-0], remote presence, surveillance, etc. Specifically:

- The payload format allows sub-codestream latency such that the first RTP packet of a given codestream can be emitted before the entire codestream is available. Specifically, the payload format does not rely on the JPEG 2000 PLM (Packet length, main header) and PLT (Packet length, tile-part header) 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 indicate 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 format defines an extended sequence number, which extends the standard (16-bit) sequence number of the RTP Fixed Header by storing additional (high-order) bits in the payload header (ESEQ field). This enables the payload format to accommodate high data rates without ambiguity, since 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 RTP packets, particularly in instances where the round-trip time is greater than the rollover 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.

In addition to support for sub-codestream latency and high-precision sender clock recovery, the payload format improves on [RFC5371] by supporting:

- code-block caching for screen content (see Section 7.9);
- progressive segmented frame (PsF) video support (see Appendix B); and
- explicit colorspace signaling (see Section 5.3).

Finally, the payload format also makes use of the unique scalability features of JPEG 2000 to allow an intermediate system or recipient to discard resolution levels and/or quality layers merely by inspecting RTP packet headers (QUAL and RES fields), without having to parse the underlying codestream (see Section 7.2).

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.

In case of conflict between the contents of a figure and the prose, the prose takes precedence.

3. Media Format Description

The following summarizes the structure of the JPEG 2000 codestream, which is specified in detail in [JPEG2000-1].

NOTE: As described in 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, each uniformly sampled on a common rectangular reference grid. For example, an image can consist of the customary Y (luma), C_b (blue-difference chroma), and C_r (red-difference chroma) components, with the C_b and C_r being subsampled by a factor of two compared to the Y component.

An image can be further divided into contiguous rectangular tiles that are each independently coded and decoded.

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

For convenience to the reader, the following lists both the abbreviated and full names of marker segments that are mentioned in this specification (several other marker segments are defined by JPEG 2000 and can be present in a codestream):

CAP:	Extended Capabilities
COC:	Coding style component
COD:	Coding style default
COM:	Comment
EOC:	End of codestream
PLM:	Packet length, main header
PLT:	Packet length, tile-part header
SOC:	Start of codestream
SOD:	Start of data
SOT:	Start of tile-part

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.

At the heart of JPEG 2000 coding is the wavelet transform, which decomposes the image into successive resolution levels, with each level related to the next one by a spatial factor of two, i.e., each successive resolution level has half the horizontal and half the vertical resolution of the previous one.

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, represent code-blocks belonging to a spatial region within one resolution level of one component.

The coded image 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 comes before the last component of the image.

Resolution Level (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 particularly useful for subframe latency operations.

4. Video Signal Description

This RTP payload format supports three distinct techniques for scanning video frames:

- Progressive frame
- Interlaced frame, where each frame consists of two fields. Field 1 occurs earlier in time than 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 | .. | SOT | .. | SOD | ............. | EOC |  P  | SOC  ...
+-----+-//-+-----+-//-+-----+--------//-----+-----+-----+---------
|          |                                            |
<----------> Main header                                |
|                                                       |
+------------------------------+------+--//-+-----------+---------
|             Main             | Body | ... |    Body   | Main ...
+------------------------------+------+--//-+-----------+---------
|                              |
<--------- RTP Packet --------->
```

Figure 1: Packetization of a Sequence of JPEG 2000 Codestreams (Not to Scale)

In Figure 1, P denotes (optional) padding bytes. See Section 3 for expansions of SOC, SOD, SOT, and EOC.

Each RTP packet, as specified in [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 in 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 in 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 padding bytes (see Figure 1).

The sender **MUST** set the value of each padding byte to zero.

The receiver **MUST** ignore the values of the padding bytes.

The JPEG 2000 codestreams **MUST** conform to Section 6.

NOTE 1: Padding bytes can be used to achieve constant bit rate transmission.

A JPEG 2000 codestream fragment, and thus an RTP packet, 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.

NOTE 2: The concept of the JPEG 2000 codestream Extended Header is specific to this document and is distinct from the JPEG 2000 codestream main header, which is defined in [JPEG2000-1]. The codestream main header consists of the bytes between, and including, the SOC marker and the first SOT marker. The codestream main header is a subset of the codestream Extended Header (see Figure 1).

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.

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

`sequence number`
The low-order bits of the extended sequence number.

The high-order bits of the extended sequence number are contained in the `ESEQ` field, which is specified in Section 5.3.

The extended sequence number is calculated as follows:

```
<extended sequence number> = <ESEQ field> * 65536 + <sequence
number field of the RTP Fixed Header>
```

5.3. Main Packet Payload Header

Figure 2 specifies the structure of the payload header. Fields are interpreted as unsigned binary integers in network order.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|MH | TP  |ORDH |P|XTRAC|        PTSTAMP        |     ESEQ      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|S|C| RSVD  |*|    PRIMS      |    TRANS      |      MAT      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              XTRAB                            |
|                               ...                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

* RANGE
```

Figure 2: Structure of the Main Packet Payload Header

MH (Codestream Main Header Presence): 2 bits

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 (Image Type): 3 bits

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 Sections 8.6 and 7.8.

ORDH (Progression Order Flag, Main Packet): 3 bits

Specifies the progression order used by the codestream and whether resync points are signaled. See Section 3 for details about progression orders.

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 if the codestream consists of more than one tile.

NOTE 1: Only `ORDH` = 4 and `ORDH` = 6 allow sub-codestream latency streaming.

NOTE 2: Progression order PRCL is defined in [JPEG2000-2]. The other progression orders are specified in [JPEG2000-1].

P (Precision Timestamp Presence): 1 bit

0 `PTSTAMP` is not used.

1 `PTSTAMP` is used.

XTRAC (Extension Payload Length): 3 bits

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

PTSTAMP (Precision Timestamp): 12 bits

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 RTP 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.

NOTE 3: As described in Sections 7.4 and 8.1, `PTSTAMP` is intended to improve clock recovery at the receiver and only applies when the transmission time of two consecutive RTP packets with identical `timestamp` fields differ by no more than 45 ms (which is 4095/90,000). [RFC5450] addresses the general case when an RTP packet is transmitted at a time other than its nominal transmission time.

ESEQ (Extended Sequence Number High-Order Bits): 8 bits

The high-order bits of the extended sequence number.

NOTE 4: The low-order bits of the extended sequence number are stored in the `sequence number` field of the RTP Fixed Header.

Section 5.2 specifies the formula to compute the extended sequence number.

R (Codestream Main Header Reuse): 1 bit

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

1 All Main Packets in this stream, as identified by the value of the `SSRC` field in the RTP Fixed Header:

- **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 (Parameterized Colorspace Presence): 1 bit

0 Component colorimetry is not specified; it is left to the session or the application.

`PRIMS`, `TRANS`, `MAT`, and `RANGE` **MUST** be zero.

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

The codestream components **MUST** conform to one of the combinations in 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

Table 1: Mapping of Codestream Components to Color Channels

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.

C (Code-Block Caching Usage): 1 bit

0 Code-block caching is not in use.

1 Code-block caching is in use.

R **MUST** be equal to 1.

RSVD (Reserved): 4 bits

Unassigned value. See Sections 8.5 and 7.7.

RANGE (Video Full Range Usage): 1 bit

Value of the VideoFullRangeFlag specified in [REC-ITU-T-H273].

PRIMS (Color Primaries): 8 bits

One of the ColourPrimaries values specified in [REC-ITU-T-H273].

TRANS (Transfer Characteristics): 8 bits

One of the TransferCharacteristics values specified in [REC-ITU-T-H273].

MAT (Color Matrix Coefficients): 8 bits

One of the MatrixCoefficients values specified in [REC-ITU-T-H273].

XTRAB (Extension Payload): variable

Allows the contents of the Main Packet Payload Header to be extended in the future. The size of the field is determined by Section 5.3. See also Sections 8.4 and 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.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |MH | TP  |RES  |*|QUAL |       PTSTAMP         |     ESEQ      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |         POS           |                  PID                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 * ORDB
```

Figure 3: Structure of the Body Packet Payload Header

MH: See Section 5.3.

TP: See Section 5.3.

RES (Resolution Levels): 3 bits

0 The payload can contribute to all resolution levels.

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 (Progression Order Flag, Body Packet): 1 bit

0 No resync point is specified for the payload.

1 The payload contains a resync point.

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

QUAL (Quality Layers): 3 bits

0 The payload can contribute to all quality layers.

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

PTSTAMP: See Section 5.3.

ESEQ: See Section 5.3.

POS (Resync Point Offset): 12 bits

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 (Precinct Identifier): 20 bits

Unique identifier of the precinct of the resync point.

```
PID = c + s * num_components
```

where:

c: Index (starting from 0) of the image component to which the precinct belongs.

s: Sequence number that identifies the precinct within its tile-component.

num_components: 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

A JPEG 2000 codestream consists of the bytes between, and including, the SOC and EOC markers, as defined in [JPEG2000-1].

The JPEG 2000 codestream **MAY** include capabilities beyond those specified in [JPEG2000-1], including those specified in [JPEG2000-2] and [JPEG2000-15].

NOTE 1: The `Rsiz` parameter and `CAP` marker segments of each JPEG 2000 codestream contain detailed information on the capabilities necessary to decode the codestream.

NOTE 2: The `caps` media type parameter defined in Section 9.2 allows applications to signal required device capabilities.

NOTE 3: The block coder specified in [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; conversely, if any decomposition level involves only vertical decomposition, then no decomposition level **MUST** involve only horizontal decomposition.

7. Sender

7.1. Main Packet

A Main Packet **MAY** contain zero or more bytes of the JPEG 2000 codestream Extended Header.

The sender **MUST** emit either 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 Packets of a given codestream, with the exception of:

- `MH`;
- `ESEQ`; and
- `PTSTAMP`.

7.2. RTP Packet Filtering

An intermediate system **MAY** strip out RTP packets from a codestream that are of no interest to a particular client, e.g., based on a resolution or a spatial region of interest.

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 1: Resync points cannot be specified if the codestream consists of more than one tile (`ORDB` and `ORDH` are both equal to zero).

NOTE 2: A resync point can be used by a receiver to process a codestream even if earlier RTP packets in the codestream have been corrupted, lost, or deliberately discarded by an intermediate system. As a corollary, resync points can be used by an intermediate system to discard RTP 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 90 kHz clock. The counter is sampled at the point in time when each RTP packet is transmitted and the 12 least significant bits of the sample are stored in the `PTSTAMP` field.

If `P` = 1, then the transmission time `TOFF` (as defined in 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 RTP packets based on resolution, as described in Section 8.3.

7.6. Extra Information

The sender **MUST** set the value of `XTRAC` to 0.

Future updates to this RTP payload format can permit other values.

7.7. Unassigned Values

The sender **MUST** set unassigned values to 0.

Unassigned values are available for assignment by future updates to this RTP payload format. As specified in Section 8.5, these future assigned values are ignored by receivers that conform to this specification. In contrast to extension values (see Section 7.8), this mechanism allows updates to this RTP payload format that remain compatible with 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 only applies 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, a receiver maintains a simple cache of previously received code-blocks, which it uses to replace empty code-blocks (see Section 8.7).

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

The sender cannot, however, determine with certainty the state of the receiver's cache; for example, 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
- it contains an HT cleanup segment and the first two bytes of the Magsgn byte-stream are between `0xFF80` and `0xFF8F` (if the code-block conforms to [JPEG2000-15]).

NOTE: The last condition allows the encoder to insert padding bytes to achieve a constant bit rate even when a code-block does not contribute code-bytes, as suggested in Annex F.4 of [JPEG2000-15].

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 RTP 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 transform (DWT) stages.

Decomposition Level	Resolution Level	Sub-bands	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 RTP packets where `RES` >= 6 since those RTP 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	Sub-bands	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 DWT stages consist of only horizontal transforms, as specified in 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 Tables 2 and 3.

8.4. Extra Information

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

Future updates to this RTP payload format can specify XTRAB contents such that this content can be ignored by receivers that conform to this specification.

8.5. Unassigned Values

The receiver **MUST** ignore unassigned values (see additional discussion in Section 7.7).

8.6. Extension Values

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

8.7. Code-Block Caching

This section only applies if `C` = 1.

When `C` = 1, 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 (see Section 7.9).

When decoding a codestream, the following procedures apply 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.
- 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 in Section 9.2. This media type has been registered in accordance with [RFC4855] using the template in [RFC6838].

9.2. Definition

Type name: video

Subtype name: jpeg2000-scl

Required parameters: N/A

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 4, and the RTP header and payload **MUST** conform to 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 to 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 parameter 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 as follows using ABNF [RFC5234]:

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

The application that defines the `absolute-URI` **MUST** ensure that it does not contain any `";"` character and **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.

`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: JPEG 2000 is a flexible image format. As a result, the size of the memory structures required to process JPEG 2000 images can vary greatly depending on the characteristics of the image and the encoding parameters. For example, the JPEG 2000 syntax allows image height and width up to 2^{32} - 1 pixels, which is also captured in the syntax of the `height` and `width` parameters of this media type. Therefore, implementations **SHOULD** take care when processing input that influences the size of memory structures and **SHOULD** fail gracefully when resource constraints are exceeded.

See also Section 13.

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: RFC 9828

Applications that use this media type: video streaming and communication

Fragment identifier considerations: N/A

Additional information: N/A

Person & 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 in [RFC3550]. Transport within other framing protocols is not defined at the time.

Author: Pierre-Anthony Lemieux <pal@sandflow.com>

Change controller: IETF

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. Congestion Control

Since this RTP payload format can be used in a variety of applications across many different contexts, there is no single congestion control mechanism that will work for all. Below is a non-exhaustive list of example mechanisms that can be used:

- a sender can offer several alternative streams for a given video signal, each with a different data rate corresponding to a different compression ratio;
- a sender can modulate the data rate within a stream by modulating the resolution, frame rate, or compression ratio of the underlying video signal; or
- an intermediate system can lower the data rate of a stream by discarding resolution levels and/or quality layers, as specified in Section 7.2.

As suggested in Section 10 of [RFC3550], RTCP feedback can be used in the data rate adaptation process.

12. IANA Considerations

IANA has registered the media type specified in Section 9.

13. 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], and [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 lies with 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 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. In addition, the RTP payload format does not contain any active content.

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

14. References

14.1. Normative References

[JPEG2000-1] ITU-T, "Information technology - JPEG 2000 image coding system: Core coding system", ITU-T Recommendation T.800, July 2024, <https://www.itu.int/rec/T-REC-T.800/>.

[JPEG2000-15] ITU-T, "Information technology - JPEG 2000 image coding system: High-throughput JPEG 2000", ITU-T Recommendation T.814, June 2019, <https://www.itu.int/rec/T-REC-T.814/>.

[JPEG2000-2] ITU-T, "Information Technology - JPEG 2000 image coding system - Extensions", ITU-T Recommendation T.801, August 2023, <https://www.itu.int/rec/T-REC-T.801>.

[JPEG2000-9] ITU-T, "Information technology - JPEG 2000 image coding system: Interactivity tools, APIs and protocols", ITU-T Recommendation T.808, December 2022, <https://www.itu.int/rec/T-REC-T.808>.

[REC-ITU-T-H273] ITU-T, "Coding-independent code points for video signal type identification", ITU-T Recommendation H.273, July 2024, <https://www.itu.int/rec/T-REC-H.273>.

[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>.

[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>.

[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>.

[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>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

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

14.2. Informative References

[OV2110-0] SMPTE, "Professional Media Over Managed IP Networks, Roadmap for the 2110 Document Suite", 4 December 2018, <https://pub.smpte.org/doc/ov2110-0/20181204-pub/>.

[RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <https://www.rfc-editor.org/info/rfc3551>.

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

[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>.

[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>.

[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>.

[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>.

[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>.

[RFC5450] Singer, D. and H. Desineni, "Transmission Time Offsets in RTP Streams", RFC 5450, DOI 10.17487/RFC5450, March 2009, <https://www.rfc-editor.org/info/rfc5450>.

[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>.

[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>.

Appendix A. Pixel Formats

Table 4 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
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 4: Defined Pixel Formats

Each pixel format is characterized by the following:

`NAME`: Identifies the pixel format

`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 subsampled.

`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.

`TRANS`: Identifies the transfer characteristics allowed by the pixel format, as defined in [REC-ITU-T-H273].

`PRIMS`: Identifies the color primaries allowed by the pixel format, as defined in [REC-ITU-T-H273].

`MAT`: Identifies the matrix coefficients allowed by the pixel format, as defined in [REC-ITU-T-H273].

`VFR`: Allows values of the VideoFullRangeFlag defined in [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.

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