

INTERNET-DRAFT  
[draft-edwards-avt-rtp-jpeg2000-00.txt](#)

Eric Edwards  
Satoshi Futemima  
Eisaburo Itakura  
Takahiro Fukuhara  
Sony Corporation  
November 14, 2001  
Expires: May 13 2002

## **RTP Payload Format for JPEG 2000 Video Streams**

Status of this memo

This document is an Internet-Draft and is in subject to all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference materials or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/1id-abstracts.html>

The list of Internet-Drafts Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

### **Abstract**

This document describes a payload format for transporting JPEG 2000 video streams using RTP (Real-time Transport Protocol). JPEG 2000 video streams are formed as a continuous series of JPEG 2000 still images which is next-generation still image coding. The JPEG 2000 payload format described in this document has three features: (1) Improvement of robustness to packet loss by fragmenting JPEG 2000 packet units intelligently, (2) Persistency of main header to minimize loss effect, (3) Priority information field for scalable delivery from the same codestream. These will allow the scalability and robustness of JPEG 2000 to be maximized in streaming applications.

### **1. Introduction**

This document specifies payload formats for JPEG 2000 video streams over the Real-time Transport Protocol (RTP).

JPEG-2000 is the international standardization system for next-generation still image encoding and its basic encoding technology is described in [[1](#)].

Edwards, et al.

[Page 1]

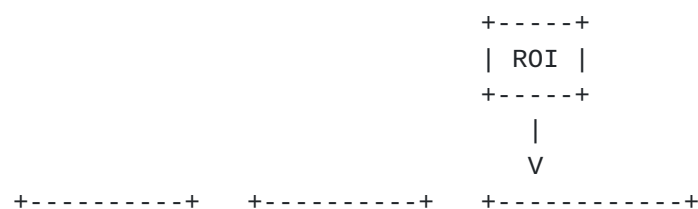
In JPEG 2000 part 3, Motion JPEG 2000 is defined[2]. However, this defines only the file format but not the transmission format for streaming on the Internet. For this reason, it is necessary to define the RTP format for JPEG 2000 video streams.

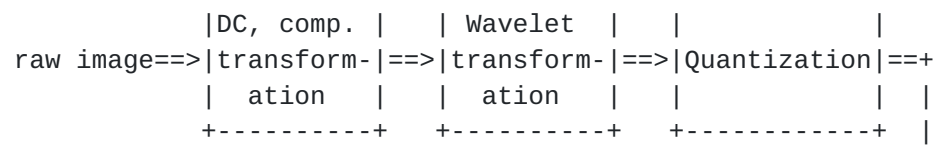
JPEG 2000 supports many features over the current JPEG standard[3][4][5].

- o Higher compression efficiency than JPEG with less visual loss especially at bit rates less than 0.25bpp for grayscale images.
- o A single codestream that offers both lossy and superior lossless compression.
- o Transmission over noisy environments. The JPEG 2000 codestream can be built with markers to boost its error resilience and recovery. The JPEG 2000 codestream is very robust to bit errors as it has been designed to avoid catastrophic decoding failure due to bit errors.
- o Progressive transmission by pixel accuracy and resolution: Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution is essential for many applications. This feature allows the reconstruction of images with different resolutions and pixel accuracy, as needed or desired, for different target devices. The image architecture provides for the efficient delivery of image data in many applications such as client/server applications.
- o Random codestream access and processing. There are parts of an image which maybe more important than others. Specific regions of the codestream can be defined to be less distorted than other areas. Access to any specific area of an image is handled efficiently without the need to completely decompress the codestream. Simple image transforms (rotating, translation, filtering) can be done with compressed codestream.

First, the JPEG 2000 algorithm is briefly explained below.

Fig. 1 shows a block diagram of JPEG 2000 encoder.





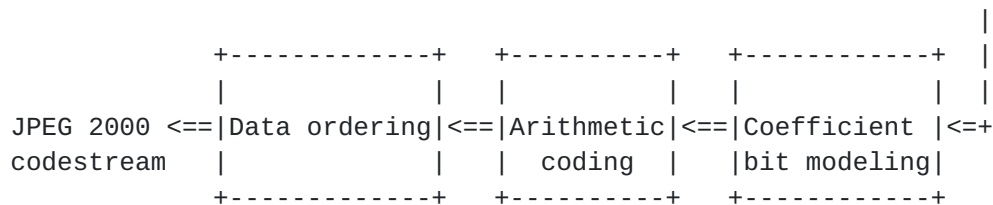


Fig. 1: Block diagram of the JPEG2000 encoder

First, the image will go through component separation, if it is a color image. Split into RGB, YUV, or various other colorspace. It can also further be sectioned into tiles within the image for processing.

Each color component or tile is transformed into the wavelet coefficients. The component or tile is sampled into various levels usually subsampled vertically and horizontally from high frequencies (which contains all the sharp details) to the low frequencies (which contains all the flat areas). These wavelet coefficients are categorized into different frequencies called subbands. Subband HH has the high frequency information, then HL and LH are the contains the middle frequencies, and the lowest frequencies and most important coefficients are in the LL subband.

Quantization is performed on the coefficients within each subband. The wavelet coefficient is divided by the quantization step size and the result is truncated. This can happen iteratively to produce an accurate target bitrate.

After quantization, code-blocks are formed from within the precincts within the tiles. Precincts are a finer separation than tiles and code-blocks are the smallest separation of the image data. Entropy coding is performed within each code-block and arithmetically encoded by bitplane. There are 3 passes for the code-block: significance propagation pass, magnitude refinement pass, and cleanup pass.

After the coefficients of all code-blocks have been coded into a short bitstream, a header is added turning it into a packet. The header has all the information needed to decompress the packet into code-blocks. A group of packets is called layers.

For additional features in transmitting, a re-ordering of the formed packets is necessary. The standard has four ways to transmit and decode a compressed image by: resolution, quality, location, or component. As there are many markers builtin to the codestream of JPEG 2000, a parser can go through the bitstream and get the proper order of packets to transmit and decode.

This is only to serve as an introduction to JPEG 2000 to aid in understanding the rest of this document. Further details of the encoder can be found in various texts on JPEG 2000.

To decompress a JPEG 2000 codestream, one would follow the reverse order of the encoding order, minus the quantization step. It is outside the scope of this document to describe in detail this procedure. Please refer to various JPEG 2000 texts for details.

## **2. JPEG 2000 video features**

As described above, JPEG 2000 has the following features.

- o Higher compression efficiency than existing JPEG and yet less SNR deterioration  
(improved compression efficiency over JPEG with dramatic improvements at low bitrates)
- o Random codestream access and processing
- o Both lossless and compression and lossy compression can be performed by the same algorithm.
- o Optional spatial resolution and SNR progressive can be easily taken out from a single codestream.  
(NOTE)SNR means Signal to Noise Ratio. This is the factor to define the quality.
- o Parts of an image can have more bits for more detail. (ROI (Region of Interest) function)
- o Various levels of error resilience functionality.

JPEG 2000 video streams are formed as a continuous series of JPEG 2000 still images, so the above features of JPEG 2000 can be used effectively. JPEG 2000 video stream has the following merits.

- o SNR is improved at a low bit rate. The formation can be used as a video stream format at a low band.
- o This is a Full Intra format in which each frame is independently compressed has a low encoding and decoding delay. This is suitable for interactive video communication. Even if a packet loss occurs in any part of the frame, error is not propagated to subsequent frames. Moreover, each frame can be handled independently this facilitates video editing.
- o JPEG 2000 has flexible and accurate rate control. This is suitable for traffic control and congestion control at the Internet transmission.
- o JPEG 2000 can provide within its own codestream error

resilience markers to aid in codestream recovery. An encoder can insert a resynchronization marker at the beginning of a JPEG 2000 packet and a segmentation symbol at the end of the bit plane to aid in recovery within a frame.



### **3. Requirements for RTP payload format of JPEG 2000 video streams**

To provide a payload format that makes the most of the merits of JPEG 2000 video stream, described in the previous section, the following must be taken into consideration.

- Provisions for packet loss

On the Internet, 5% packet loss is common and this percentage may sometimes come to 20% or more. To split JPEG 2000 video streams into RTP packets, efficient packetization of the codestream is required to minimize the effects of disabled decoding due to missing code-blocks over error prone environments. If the main header is lost in transmission, the decoding ability is lost. Accordingly, a system to compensate for the loss of the main header as much as possible is required.

- A packetizing scheme that permits making the most of the JPEG 2000 functionality.

A packetizing scheme so that an image can be progressively transmitted and reconstructed progressively by the receiver using JPEG 2000 functionality. Maximizing performance over various network conditions and various computing power of receiving platforms.

### **4. Proposal for an RTP payload format for JPEG 2000 video streams**

#### **4.1 RTP fixed header usage**

For each RTP packet, the RTP fixed header is followed by the JPEG 2000 payload header, which is followed by JPEG 2000 codestream. The RTP header fields that have a meaning specific to the JPEG 2000 video are described as follows:

**Payload type (PT):** The payload type is dynamically assigned by means outside the scope of this document. A payload type in the dynamic range SHALL be chosen by means of an out of band signaling protocol (e.g., RTSP, SIP, etc).

**Marker bit (M):** The marker bit of the RTP fixed header is set to 1 on the last RTP packet of a video frame, and otherwise, must be 0. When transmission is performed by multiple RTP sessions, the bit is set in the last packet of the frame in each session.

**Timestamp:** The RTP timestamp is in units of 90 KHz. The same timestamp must appear in each fragment of a given frame. The

initial value of the timestamp is random (unpredictable) to make known-plaintext attacks on encryption more difficult, even if the source itself does not encrypt, because the packets may flow through a translator that does.

## 4.2 RTP Payload header format

The RTP payload header format for JPEG 2000 video stream is as follows:

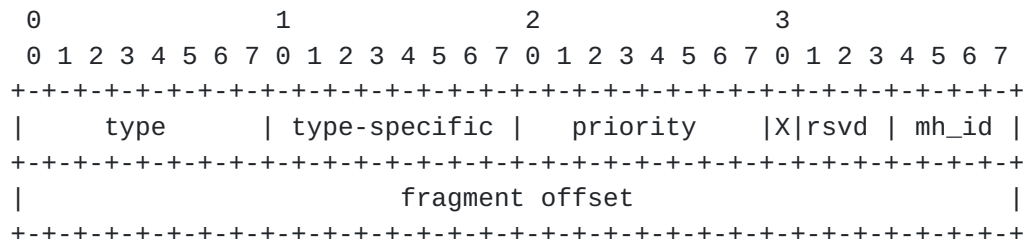


Fig. 2: RTP payload header format for JPEG 2000

type : 8 bits

The type field shows which part of JPEG 2000 codestream is included. The details of the type are described later.

type-specific : 8 bits

Interpretation depends on the value of the type field.  
 This field is defined for future usage.  
 This bit must be set to 0 when not used.  
 i.e. Tile specific priority number (general idea)

priority : 8 bits

The priority field shows the importance of the JPEG 2000 packet included in the given RTP packet. Typically, the higher priority is set at the packet which contains the JPEG 2000 packets of the lower layers and the lower subbands.

X : 1 bit

extension bit. This bit must be set to 1 when JPEG 2000 optional payload header follows the JPEG 2000 payload header, and otherwise set to 0.  
 The details of the optional payload header is described later.

rsvd : 3 bits

These bits are reserved for future use and must be set to 0.

mh\_id : 4 bit

identification of the main header of JPEG 2000. The same mh\_id

is used as long as the coding parameters described in the main header remain unchanged.

fragment offset : 32 bits

Edwards, et al.

[Page 6]

Because JPEG 2000 frames are typically larger than underlying network's maximum transfer units (MTU), frames may often be fragmented into several packets. The fragment offset is the data offset in bytes of the current packet from the first byte in the JPEG 2000 codestream. This field helps the receiver to reassemble JPEG 2000 codestream.

To perform scalable video delivery by using multiple RTP sessions, the offset value from the first byte of the same frame is set for fragment offset. Accordingly, in scalable video delivery using multiple RTP sessions, maybe the fragment offset will not be started with 0 in some RTP sessions even if the packet is the first one of the frame.

## 5. Fragmentation of JPEG 2000 codestream and Type Field

Fig. 2 shows the construction of the JPEG 2000 codestream. The JPEG 2000 codestream consists of a main header beginning with the SOC marker, one or more tiles (only one tile for no tile division), and the EOC marker to indicate the end of the codestream. Each tile consists of a tile-part header starts with the SOT marker and ending with the SOD marker, and a bit stream (a series of JPEG 2000 packets) of the bit stream.

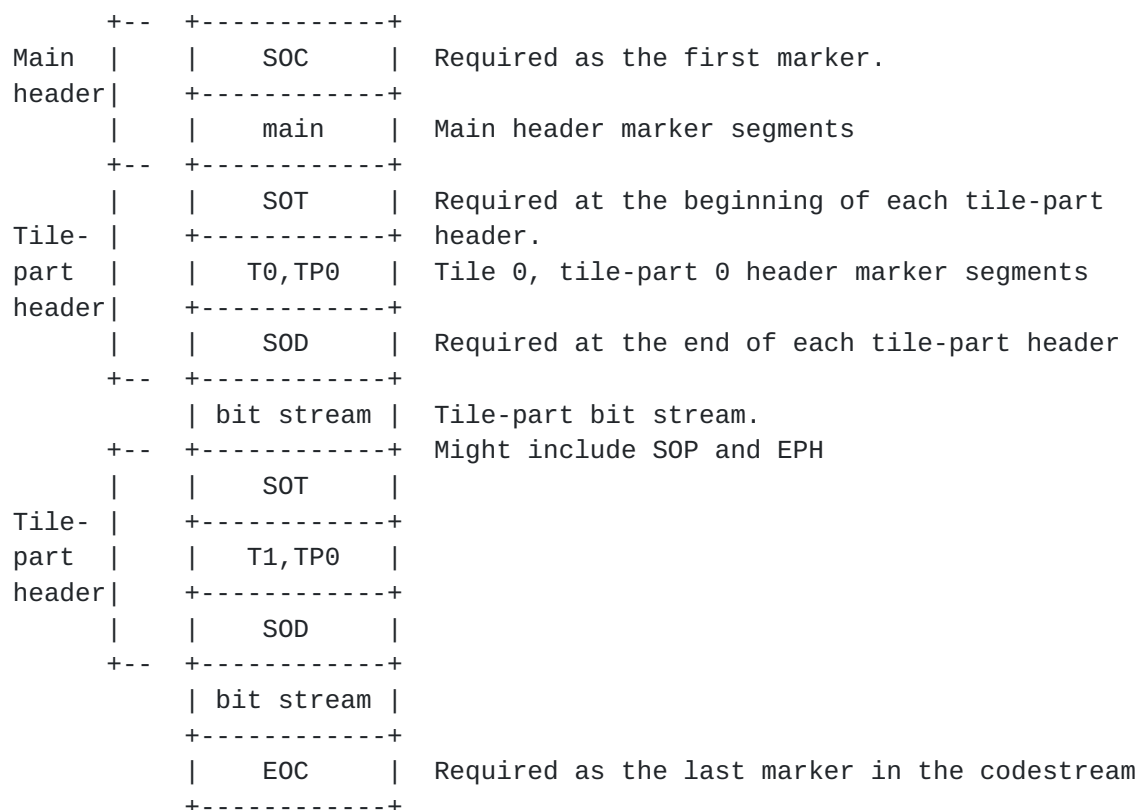


Fig. 3: Construction of the JPEG 2000 codestream

JPEG 2000 video streams are typically larger than underlying

Edwards, et al.

[Page 7]

network's maximum transfer units (MTU), video sequence may often be fragmented into several IP packets at the network layer. the JPEG 2000 video streams are fragmented into RTP packets according to the following basic rule.

The JPEG 2000 construction consists of a main header, tile-part headers, and JPEG 2000 packets. When we packetize the JPEG 2000 codestream, these construction units from the codestream should be maintained. Each RTP packet should consist of a main header, tile-part header, or JPEG 2000 packet.

If the sender understands JPEG 2000 codestream and can read the JPEG 2000 packets from the codestream. (i.e. the sender is intelligent) JPEG 2000 packets should be packed into RTP payload packets in the following way:

1. If the JPEG 2000 packets are smaller than the MTU size, the sender should put as many whole JPEG 2000 packets into a single RTP packet. That is, the JPEG 2000 payload data should begin with one of the SOC marker, SOT marker, or SOP marker (if it exists).
2. If the JPEG 2000 packets are larger than the MTU size, the sender should segment the JPEG 2000 packets at the largest possible MTU size but without JPEG 2000 packets overlapping.

If the server does not understand JPEG 2000 codestream (i.e. the sender is not intelligent,) it should pack JPEG 2000 codestream in the largest possible MTU data size for the RTP packet. JPEG 2000 codestream will be segmented along arbitrary lengths by the sender into RTP packets.

Regardless of the sender's capabilities, the receiver must be able to handle RTP packets of any size.

If we do not fragment at the sender, any packets larger than the MTU size, will be fragmented into multiple smaller IP packets than the MTU size by the IP layer. If one fragmented IP packet is lost during transmission, it is recognized as a loss of the whole RTP packet because the receiving host cannot reassemble the RTP packet. The segmentation of the JPEG 2000 codestream into RTP packets, should fit within the RTP payload size.

In the following, all the possible packetization cases are described with diagrams. For each case, the type field value shown in Fig. 2 is also indicated.

### **5.1 Separation at arbitrary lengths**

In this case, a JPEG 2000 codestream is split into several fragments at arbitrary byte-position.  
The type value of the RTP packet is set to 0.



```

+---+---+---+-----+
|RTP|PL |SOC| jpeg 2000 codestream |           type = 0
|hdr|hdr|   | fragment (1)         |
+---+---+---+-----+
+---+---+---+-----+
|RTP|PL |   | jpeg 2000 codestream |           type = 0
|hdr|hdr|   | fragment (2)         |
+---+---+---+-----+
...
+---+---+---+-----+
|RTP|PL | jpeg 2000 codestream |EOC|           type = 0
|hdr|hdr| fragment (N)         |   |
+---+---+---+-----+
*PL hdr = payload header

```

Such RTP packetization scheme is not recommended from the standpoint of error resilience. It is desirable to use it only in some limited environments shown below.

- The sender finds it difficult to distinguish the main header, tile header, and JPEG 2000 packets from one another. There is no SOP marker in the JPEG 2000 codestream. The sender is not intelligent.
- The network environment is error free.
- If the JPEG 2000 error resilience markers (TLM, PLM, PLT, PPM, and PPT markers) are present in the codestream. Error resilience will be handled outside of RTP. Its description is not within the scope of this document. Using these markers may improve the error resilience.

## **5.2 General JPEG 2000 RTP packet types**

- (1) JPEG 2000 main header(SOC marker) must come first of the RTP payload (just after the RTP payload header). The type value of the RTP packets which contain the whole main header (not fragmented) is 4,

(1-a) The RTP packet only contains the complete main header.

```

+---+---+-----+
|RTP|PL |Main |           type = 4
|hdr|hdr|header|
+---+---+-----+

```

(1-b) The main header and the first tile-part header are packed into one RTP packet.

```
+---+---+-----+-----+
|RTP|PL |Main  |Tile-part|
|hdr|hdr|header|header  |
+---+---+-----+-----+
```

type = 4

(1-c) The main header, the first tile-part header and JPEG 2000 packet(s) are packed into one RTP packet.

```
+---+---+-----+-----+-----+-----+
|RTP|PL |Main  |Tile-part|jpeg 2000| ... |jpeg 2000|      type = 4
|hdr|hdr|header|header  |packet   |      |packet   |
+---+---+-----+-----+-----+-----+
```

(1-d) The main header is split into the several RTP packets.

If the main header is larger than one RTP packet, then it may be split into several RTP packets.

In this case, the RTP packets must contain only a piece of the main header. The type value of the RTP packets which contain the first piece of the main header is type 5, and the last piece is type 7 and the middle pieces are all type 6.

```
+---+---+-----+
|RTP|PL |Main Header(1)|      type = 5
|hdr|hdr|              |
+---+---+-----+
+---+---+-----+
|RTP|PL |Main Header(2)|      type = 6
|hdr|hdr|              |
+---+---+-----+
+---+---+-----+
|RTP|PL |Main Header(3)|      type = 6
|hdr|hdr|              |
+---+---+-----+
...
+---+---+-----+
|RTP|PL |Main Header(N)|      type = 7
|hdr|hdr|              |
+---+---+-----+
```

(Note) When the main header is split into multiple RTP packets, the first tile-part header must not be included in the RTP packet containing the last fragment.

```
+---+---+-----+
|RTP|PL |Main Header(N)|Tile-part|      This packetization is
|hdr|hdr|              |header  |      not allowed.
+---+---+-----+
```

- (2) Tile-part headers (SOT marker) must come first of the RTP payload (just after the RTP payload header), except for the first tile-part header just after the main header.

The first tile-part header may either be packed with the main header, or be separated to another RTP packet.  
The type value of the RTP packet which begins with the tile-part header is 8.

(2-a) The RTP packet only contains the complete tile-part header.

```
+---+---+-----+
|RTP|PL |Tile-part |           type = 8
|hdr|hdr|Header   |
+---+---+-----+
```

(2-b) The tile-part header and JPEG 2000 packet(s) are packed into one RTP packet.

```
+---+---+-----+-----+-----+
|RTP|PL |Tile-part |jpeg 2000| ... |jpeg 2000|   type = 8
|hdr|hdr|Header   |packet  |   |   |
+---+---+-----+-----+-----+
```

(2-c) The tile-part header is split into the several RTP packets.

If the tile-part header is larger than one RTP packet, it may be split into several RTP packets. In this case, the RTP packets contain only a piece of the tile-part header.

The RTP packets which contain the first piece of the tile-part header is type 9, and the last piece is type 11, and the middle pieces are all type 10.

```
+---+---+-----+
|RTP|PL |Tile-part header |           type = 9
|hdr|hdr|fragment(1)     |
+---+---+-----+
+---+---+-----+
|RTP|PL |Tile-part header |           type = 10
|hdr|hdr|fragment(2)     |
+---+---+-----+
+---+---+-----+
|RTP|PL |Tile-part header |           type = 10
|hdr|hdr|fragment(3)     |
+---+---+-----+
...
+---+---+-----+
|RTP|PL |Tile-part header |           type = 11
|hdr|hdr|fragment(N)     |
+---+---+-----+
```

(Note) When the tile-part header is split into multiple RTP packets, the JPEG 2000 packet must not be included in the RTP packet containing the last fragment.

```
+---+---+-----+
|RTP|PL |Tile-part header |jpeg 2000| This packetization is
```

```
|hdr|hdr|fragment(N)      |packet  |  not allowed.  
+---+---+-----+-----+
```

- (3) The JPEG 2000 packet must be packed by itself, except for JPEG 2000 packets just after the tile-part header. Also several JPEG 2000 packets may be packed into the one RTP packet. If SOP(Start of Packet) marker is used for error resilience, SOP marker shall be placed at the beginning of the RTP payload. (When the SOP marker is used, it is placed at the beginning of the RTP packet.) The type value of the RTP packet, which contains only jpeg 2000 packet(s) is 12.

(3-a) More than one jpeg 2000 packets are packed into one RTP packet.

```
+---+---+-----+-----+-----+
|RTP|PT |jpeg 2000| ... |jpeg 2000|      type = 12
|hdr|hdr|packet  |      |packet  |
+---+---+-----+-----+-----+
```

(3-b) The jpeg 2000 packet is split into the several RTP packets

If the JPEG 2000 packet is larger than one RTP packet, then it may be split into two or more RTP packets. In this case, the RTP packets contain only a piece of the jpeg 2000 packet.

The RTP packet with the first piece of JPEG 2000 packet is type 13, and the last piece is type 15, and the middle pieces are all type 14.

```
+---+---+-----+-----+-----+
|RTP|PT |jpeg 2000 packet  |      type = 13
|hdr|hdr|fragment(1)      |
+---+---+-----+-----+-----+
+---+---+-----+-----+-----+
|RTP|PT |jpeg 2000 packet  |      type = 14
|hdr|hdr|fragment(2)      |
+---+---+-----+-----+-----+
+---+---+-----+-----+-----+
|RTP|PT |jpeg 2000 packet  |      type = 14
|hdr|hdr|fragment(3)      |
+---+---+-----+-----+-----+
...
+---+---+-----+-----+-----+
|RTP|PT |jpeg 2000 packet  |      type = 15
|hdr|hdr|fragment(N)      |
+---+---+-----+-----+-----+
```

(Note) When the JPEG 2000 packet is split into multiple RTP packets, another JPEG 2000 packet must not be included in the RTP packet containing the last fragment.

```
+---+---+-----+-----+-----+
```

RTP	PT	jpeg 2000 packet	jpeg 2000	This packetization is
hdr	hdr	fragment(N)	packet	not allowed.

+---+---+-----+-----+



## **6. Scalable Delivery and Priority field**

JPEG 2000 codestream has rich functionality built into it so decoders can easily handle scalable delivery or progressive transmission. Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution is essential for many applications. This feature allows the reconstruction of images with different resolutions and pixel accuracy, as needed or desired, for different target devices. The largest image source devices can provide a codestream that is easily processed for the smallest image display device.

The JPEG 2000 packets contain all compressed image data from a specific layer, a specific component, a specific resolution level, and a specific precinct. The order in which these packets are found in the codestream is called the "progression order". The ordering of the packets can progress along four axes: layer, component, resolution level and precinct.

Providing priority field to show importance of data contained in a given RTP packet makes the most of JPEG 2000 progressive/scalable functions.

In resolution progression order, the higher decomposition level is more important. The priority field of the RTP packet that contains the higher decomposition level is set to the higher priority. When transmitted in spatial resolution order, LL0 components data is set to the highest priority.

### **6.1 Priority mapping table**

For the progression order, the priority value to be given to each JPEG 2000 packet is defined by the priority mapping table. The higher the importance, the smaller the priority value. The priority mapping table can define the priority values for spatial resolution, layer, color component, or precinct level. This priority table is sent from the sender to a receiver through another protocol (RTSP, SIP, etc.) outside of RTP. To change the priority mapping table, a new priority mapping table must be sent from the sender to the receiver as needed.

If there is no priority mapping table, the priority value of the RTP packet must be set to '0xff'.

For example, the priority table can be sent to the receiver from the sender but the receiver will determine its own level of priority RTP packets to receive using the priority table as a guideline.

The priority value of 1 has the highest priority in the priority

mapping table. As the priority value increases, the priority becomes lower. If transmission is performed without attaching any priority mapping table, 0xff (255) must be set in the priority field.

For RTP packets that only consist of a whole or fragmented main or tile header and containing no JPEG 2000 packets, priority 0 must be set by the sender if a priority mapping table is used. (If a priority mapping table is not used, the priority value must be 0xff for the same RTP packets.)

The sender may transmit each priority using separate multiple RTP sessions defined by the priority value. For example, different priority may be allocated to other multicast groups. The sender may also transmit all priority valued RTP packets using a single RTP session.

When multiple JPEG 2000 packets are included in a single RTP packet, the higher priority value of JPEG 2000 packets is set for the whole RTP packet by the sender.

In the following, an example of priority mapping table is shown. The component based priority should be used when there is a higher priority component like Y in YUV components.

#### **6.1.1 Layer based priority**

This is an example of priority mapping table in the progression order in which SNR is improved progressively. The JPEG 2000 packet of layer 0 and resolution 0 has the highest priority. The JPEG 2000 packets with layer 0 and resolution 1 or more are next in priority. As the layer number increases, the priority becomes lower.

L	R	C	P		priority
-----+-----					
0	0	-	-		1
0	>0	-	-		2
1	-	-	-		3
....					....

#### **6.1.2 Resolution level based priority**

This is an example of priority mapping table in the progression order in which the spatial resolution is increased. The JPEG 2000 packet with layer 0 and resolution 0 has the highest priority and the JPEG 2000 packets with later 1 or more and resolution 0 are next in priority. As the resolution level increases, the priority becomes lower.

L	R	C	P		priority
-----+-----					
0	0	-	-		1

>0	0	-	-		2
-	1	-	-		3
....					....

### 6.1.3 Component based priority

The priority mapping table for component progression is used only when there is priority order among components. This example is for YUV components. The JPEG 2000 packet with layer 0, resolution 0, and component 0 has the highest priority. The JPEG packets with layer 1 or more, resolution 0, and component 0 are next in priority. The JPEG 2000 packets with resolution 0 and component 0 are the third in priority. As the resolution increases, the priority becomes lower.

L	R	C	P	priority
0	0	0	-	1
>0	0	0	-	2
	>0	0	-	3
-	-	1	-	4
....				....

## 6.2 Sender's Actions

Priority is given in accordance with the priority mapping table. The priority field is only a hint for the receiver but never forces the receiver to use any specific processing method. If the priority mapping table is not used, '0xff' must be set.

## 6.3 Receiver's Action

Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution is essential for many applications. This feature allows the reconstruction of images with different resolutions and pixel accuracy, as needed or desired, for different target devices. The image architecture provides for the efficient delivery of image data in many applications such as client/server applications. The receiver should decode packets above a certain priority to obtain maximum performance depending on the receiver's platform.

The receiver can determine on its own (using or not using the mapping table and several other variables) the priority value level the RTP packets it should decode.

For example, when the CPU power is incompetent or the terminal has only a low-resolution display, decoding only RTP packets below a certain priority permits obtaining optimal performance.

If any high-priority RTP packet is not received when a packet loss

occurs, frame(s) can be skipped because visual loss may be remarkable even if decoding can be successfully performed.

When any uninterpretable or unexpected priority is received, the

Edwards, et al.

[Page 15]

receiver must interpret packets as no priority (i.e. priority=0xff.)

## **7. JPEG 2000 main header compensation**

The JPEG 2000 image main header describes various encode parameters and the decoder decodes by using the parameters described in the main header. If the RTP packet that contains the main header is lost, the corresponding JPEG 2000 codestream cannot be decoded. In an extremely rare case, if the main header has dropped and all the remainder JPEG 2000 packets has been received successfully, the receiver cannot decode the frame. Even when the main header is lost, it can be recovered to a certain level using the following method.

A recovery of the main header that has been lost is very simple. In the case of JPEG 2000 video, it is common that encode parameters will not greatly change in each frame. Even if the RTP packet including the main header of a frame has dropped, decoding processing can be performed by using the main header of the previous frame if this previous frame is already encoded by the same encode parameters.

The mh\_id field of the payload header is used to recognize whether the encoding parameters of the main header are the same as the encoding parameters of the previous frame. The same value is set in mh\_id of the RTP packet in the same frame. mh\_id and encode parameters are not associated with each other as 1:1 but they are used to recognize whether the encode parameters of the previous frame are the same or not.

The mh\_id field is saved from previous frames to be used to recover the current frame's main header, if lost. If the mh\_id of the current frame has the same value as the mh\_id value of the previous frame, the previous frame's main header can be used to decode the current frame, in case the main header lost.

### **7.1 Sender processing**

The sender transmits RTP packets with the same mh\_id value unless the encoder parameters are different from the previous frame. The encode parameters are the fixed information marker segment (SIZ marker) and functional marker segments (COD, COC, RGN, QCD, QCC, and POC) specified in JPEG 2000 Part 1 Annex. A. If the encode parameters have been changed, the sender transmits RTP packets by incrementing the mh\_id value by one. The initial mh\_id value is 1. When the mh\_id value exceeds 15, the value returns to 1 again.

If the mh\_id field is set to 0, the receiver must not save the main

header and must not compensate for lost headers using the above method.



## 7.2 Receiver processing

When the receiver has received the main header correctly, the RTP sequence number, the mh\_id and main header are saved except when the mh\_id value is 0. Only the last main header that was received correctly is saved. That is, if there has been a saved main header, the previous one is deleted and the new main header is saved.

When the main header could not be received, the receiver compares the current mh\_id value (this mh\_id can be known by receiving at least one RTP packet) with the saved mh\_id value. When the values are the same, decoding is performed by using the saved main header.

The main header of mh\_id = 0 is an indication from the sender to not compensate for lost headers or to save any headers. .

## 8. Optional Payload Header

When the extension bit of the JPEG 2000 payload header is 1, the payload header is followed by an optional payload header. The JPEG 2000 video stream payload comes after the optional payload header. The figure shows a general format of the optional payload header.

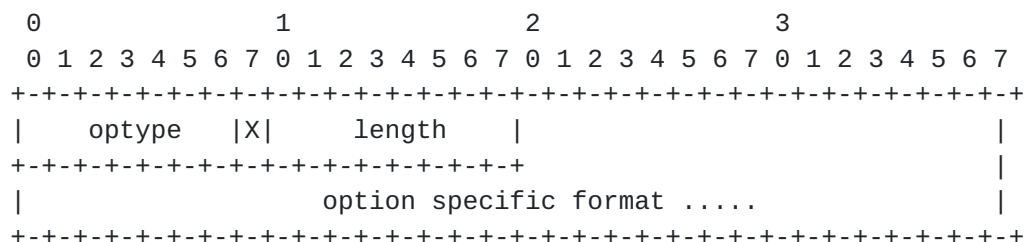


Fig. JPEG 2000 video stream optional payload header generic format

optype : 7 bits

optype shows the optional payload header type.

X : 1bit

more extension bit. This must be set to 1 if another optional payload header follows this optional payload header; otherwise it must be set to 0.

length : 8 bits

length of optional header in bytes.

The receiver performs processing for the optional header when the

extension bit of the JPEG 2000 payload header is 1.  
When having received an optype that cannot be interpreted, the receiver will skip the amount specified in the length field and not process the optional payload header..

When the more extension bit of the optional header is 1, another optional payload header will come immediately after this optional payload header.

### 8.1 Quantization Optional Header

As one of optional payload headers, the quantization optional header is defined. If only the QCD and/or QCC information has been changed, this optional payload header conveys the information. One optional payload header for QCD and another optional payload header for the QCC information. Both changes must not be conveyed in a single optional payload header.

If the receiver having received the quantization optional header but the main header of the current frame is lost; the receiver can replace the QCD and QCC information in the saved main header using the current QCD or QCC optional header only if the mh\_id value of the current frame and previous frame differ by 1. The receiver should interpret this optional payload header only when the mh\_id value changes.

This header is supposed to be used when an adjustment is made by quantization size in order to keep the amount of compressed JPEG 2000 image data at a constant level.

The quantization optional header format is shown in the figure below.

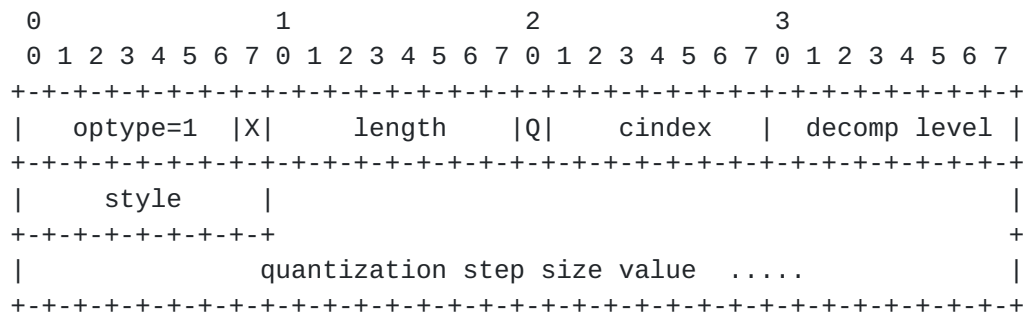


Fig. Quantization Optional Header format

Each field is explained below.

optype : 1 bit

The optype value of the quantization optional header is 1.

Q : 1 bit

This indicates whether the information is of QCD or of QCC. If the information is of QCD, 0 is set. If the information is of QCC, 1 is set.

cindex : 7 bits

When the information is of QCC, this represents a component number.

decomp level : 8 bits

This indicates the decomposition level of the corresponding frame.

style : 8 bits

This indicates the quantization style specified in the QCD and QCC marker segments. (Refer to JPEG 2000 Part I: Annex A Table A-28.)

quantization step size value : variable length

This is followed by the quantization stop size value specified by style. (Refer to JPEG 2000 Part I: Annex A Table A-29 and A-30.)

## **9. Security Consideration**

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specifications[3]. This implies that confidentiality of the media streams is achieved by encryption. Because the data compression used with this payload format is applied end-to-end, encryption may be performed on the compressed data so there is no conflict between the two operations.

## **10. Author's Address**

Eric Edwards  
Sony Corporation  
Media Processing Division  
Network & Software Technology Center of America  
3300 Zanker Road, MD: SJ2C4  
San Jose, CA 95134  
Phone: +1 408 955 6462  
Fax: +1 408 955 5724  
Email: Eric.Edwards@am.sony.com

Satoshi Futemma  
Sony Corporation  
6-7-35 Kitashinagawa Shinagawa-ku  
Tokyo 141-0001 JAPAN  
Phone: +81 3 5448 4373

Fax: +81 3 5448 4622

Email: [satosi-f@sm.sony.co.jp](mailto:satosi-f@sm.sony.co.jp)

Eisaburo Itakura

Edwards, et al.

[Page 19]

Sony Corporation  
6-7-35 Kitashinagawa Shinagawa-ku  
Tokyo 141-0001 JAPAN  
Phone: +81 3 5448 3096  
Fax: +81 3 5448 4622  
Email: itakura@sm.sony.co.jp

Takahiro Fukuhara  
Sony Corporation  
1-11-1 Osaki Shinagawa-ku  
Tokyo 141-0032 JAPAN  
Phone: +81 3 5435 3665  
Fax: +81 3 5435 3891  
Email: fukuhara@av.crl.sony.co.jp

## **11. References**

- [1] ISO/IEC JTC1/SC29/WG1: "JPEG 2000 Part I Final Draft International Standard", September 2000.
- [2] ISO/IEC JTC1/SC29/WG1: "Motion JPEG 2000 Committee Draft 1.0", <http://www.jpeg.org/public/cd15444-3.pdf>, December 2000.
- [3] A. N. Skodras, C. A. Christopoulos and T. Ebrahimi: "JPEG2000: The Upcoming Still Image Compression Standard", In Proc. of the 11th Portuguese Conference on Pattern Recognition, pp. 359-366, Porto, Portugal, May 2000.
- [4] ISO/IEC JTC1/SC29/WG1: "JPEG2000 requirements and profiles version 6.3", draft in progress, <http://www.jpeg.org/public/wg1n1803.pdf>, July 2000.
- [5] Diego Santa-Cruz, Touradj Ebrahimi, Joel Askelof, Mathias Larsson and Charilaos Christopoulos: "JPEG 2000 still image coding versus other standards", In Proc. of SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII, vol. 4115, pp. 446-454, July 2000.
- [6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson "RTP: A Transport Protocol for Real Time Applications", [RFC 1889](#), January 1996.

