Network Working Group
Request for Comments: 5371
Category: Standards Track

S. Futemma
E. Itakura
A. Leung
Sony

October 2008

RTP Payload Format for JPEG 2000 Video Streams

Status of This Memo

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

Abstract

This memo describes an RTP payload format for the ISO/IEC International Standard 15444-1 | ITU-T Rec. T.800, better known as JPEG 2000. JPEG 2000 features are considered in the design of this payload format. JPEG 2000 is a truly scalable compression technology allowing applications to encode once and decode many different ways. The JPEG 2000 video stream is formed by extending from a single image to a series of JPEG 2000 images.

Table of Contents

1. Introduction 3
1.1. Conventions Used in This Document 6
2. JPEG 2000 Video Features 6
3. Payload Design 6
4. Payload Format 7
4.1. RTP Fixed Header Usage 7
4.2. RTP Payload Header Format 8
5. RTP Packetization 10
6. Media Type Registration 11
7. SDP Parameters 14
7.1. SDP Mapping 14
7.2. Usage with the SDP Offer/Answer Model 15
7.2.1. Examples 16
7.2.2. Examples: Non-90kHz Timestamp 16
8. IANA Considerations 17
9. Security Considerations 17
10. Congestion Control 18
11. References 19
11.1. Normative References 19
11.2. Informative References 19
Appendix A. Informative Appendix 21
A.1. Recommended Practices 21
A.2. Sample Headers in Detail 22
A.2.1. Sample 1: Progressive Image with Single Tile, 3500 Bytes (i.e., thumbnail) 22
A.2.2. Sample 2: Image with 4 Tiles 24
A.2.3. Sample 3: Packing Multiple Tiles in Single Payload, Fragmented Header 25
A.2.4. Sample 4: Interlace Image, Single Tile 27

1. Introduction

This document specifies a payload format for JPEG 2000 video streams over the Real-time Transport Protocol (RTP). JPEG 2000 is an ISO/IEC International Standard and ITU-T Recommendation (ISO/IEC International Standard 15444-1 | ITU-T Rec. T.800) developed for next-generation, still-image compression. JPEG stands for the Joint Photographers Experts Group, an international group made of academia and industry to develop image compression standards. JPEG 2000 basic compression technology is defined in detail in ISO JPEG 2000 Part 1: Core Coding System [JPEG2000Pt_1], with motion defined in ISO JPEG 2000 Part 3: Motion JPEG 2000 [JPEG2000Pt_3].

Part 3 of the JPEG 2000 standard defines Motion JPEG 2000 [JPEG2000Pt_3]. However, Motion JPEG 2000 defines a file format, not a transmission format for the network. This document specifies a transmission format for the network for a series of JPEG 2000 images.

JPEG 2000 supports many powerful features [JPEG2000Pt_1] [JPEG2000Pt_3] that are not supported in the current JPEG standard, such as:

- Higher compression efficiency than JPEG with less visual distortion especially at extreme compression ratios.

- A single codestream that offers both lossy and lossless compression.

- Better error resiliency than JPEG.

- Progressive transmission by pixel accuracy (Signal-to-Noise Ratio (SNR) scalability) and resolution (resolution scalability).

- Random codestream access and processing.

The JPEG 2000 algorithm is briefly explained. Figure 1 shows a block diagram of the JPEG 2000 encoding method.

```
                                                  +-----+
                                                  | ROI |
                                                  +-----+
                                                     |
                                                     V
               +----------+   +----------+   +------------+
               |DC, comp. |   | Wavelet  |   |            |
Raw Image  ==> |transform-|==>|transform-|==>|Quantization|==+
               |  ation   |   |  ation   |   |            |  |
               +----------+   +----------+   +------------+  |
                                                             |
              +-----------+   +----------+   +------------+  |
              |           |   |          |   |            |  |
 JPEG 2000 <==| Data      |<==| Rate     |<==| EBCOT      |<=+
 codestream   | Ordering  |   | Control  |   |            |
              +-----------+   +----------+   +------------+
```

Figure 1: Block diagram of the JPEG 2000 encoder

The image is first transformed into wavelet coefficients. The image is sampled into various levels, vertically and horizontally, from high frequencies (which contain sharp details) to low frequencies (which contain smooth areas). Quantization is performed on the coefficients within each sub-band.

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.) EBCOT coding (Embedded Block Coding Optimized for Truncation) is performed within each code block and arithmetically encoded by bit plane. Rate control is performed to achieve the highest quality image for a specified rate.

As a result, for a given tile, data units called JPEG 2000 packets are generated, which contain data from a specific layer, specific component, specific resolution, or specific precinct, depending on the data ordering.

Finally, the JPEG 2000 packets are interleaved according to the progression along four axes: layer, resolution, component, and precinct. A JPEG 2000 header is added to become a fully compliant JPEG 2000 codestream.

To decompress a JPEG 2000 codestream, one would follow the reverse order of the encoding order, without the rate control.

It is outside the scope of this document to further describe in detail this procedure. Please refer to various JPEG 2000 related texts for further details [JPEG2000Pt_1].

Figure 2 shows a JPEG 2000 codestream in detail. A JPEG 2000 codestream is structured from the main header, beginning with the SOC (Start Of Codestream) marker, one or more tiles, and the EOC (End Of Codestream) marker to indicate the end of the codestream. Each tile consists of a tile-part header that starts with the SOT (Start of Tile) marker and ends with a SOD (Start of Data) marker, and bitstream (a series of JPEG 2000 packets).

```
        +--  +------------+
  Main  |    |    SOC     |  Required as the first marker
  header|    +------------+
        |    |    main    |  Main header marker segments
        +--  +------------+
        |    |    SOT     |  Required at the beginning of each
  Tile- |    +------------+    tile-part header
  part  |    |   T0,TP0   |  Tile 0, tile-part 0 header marker
  header|    +------------+    segments
        |    |    SOD     |  Required at the end of each tile-part
        +--  +------------+    header
             | bitstream  |  Tile-part bitstream
        +--  +------------+
        |    |    SOT     |
  Tile- |    +------------+
  part  |    |   T1,TP0   |
  header|    +------------+
        |    |    SOD     |
        +--  +------------+
             | bit stream |
             +------------+
                   .
                   .
                   .
             +------------+
             |    EOC     |  Required as the last marker in the
             +------------+  codestream
```

Figure 2: Basic construction of the JPEG 2000 codestream

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. JPEG 2000 Video Features

JPEG 2000 video streams are formed as a continuous series of JPEG 2000 still images. Previously described features of JPEG 2000 may be used effectively in streaming applications for a JPEG 2000 video. A JPEG 2000 video stream has the following qualities:

- At low bit rates, the SNR is improved dramatically over JPEG and Motion JPEG.

- This is a full intra-frame format -- each frame is independently compressed -- and therefore has a low encoding and decoding delay.

- JPEG 2000 has flexible and accurate rate control.

- This is suitable for traffic control and congestion control during network transmission.

- JPEG 2000 can provide its own codestream error resilience markers to aid in codestream recovery outside of this specification.

3. Payload Design

To design a payload format that maximizes JPEG 2000 features, the following are taken into consideration:

- Provisions for packet loss:

  On the Internet, 5% packet loss is common and this percentage may vary up to 20% or more. To split JPEG 2000 video streams into RTP packets, efficient packetization of the codestream is required to minimize problems in decoding due to missing packets. If the main header is lost, the image cannot be decoded.

- JPEG 2000 Scalability

  JPEG 2000 has powerful scalability features and markers in the payload header to indicate the specific meaning of the payload, such as:

  * Special markers for the headers, fragments of headers, etc.

* Tile numbering for association of packets.

* Since this is primarily for video applications, special markers are used to indicate format (i.e., interlace odd/even fields).

* Priority importance of the packet using methods described in RFC 5372 [RFC5372].

* Main header recovery using methods described in RFC 5372 [RFC5372].

Additional usage of the payload header is described in RFC 5372 [RFC5372].

4. Payload Format

4.1. RTP Fixed Header Usage

For each RTP packet, the RTP fixed header is followed by the JPEG 2000 RTP payload header, which is followed by the payload, a piece of a JPEG 2000 codestream, which is usually a JPEG 2000 packet.

The RTP header fields that have a meaning specific to a JPEG 2000 video stream are described as follows:

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

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 (i.e., Real Time Streaming Protocol (RTSP), SIP, etc.).

Timestamp: Timestamp indicates the presentation time of the frame contained in the RTP packet. The same timestamp value MUST appear in each RTP packet carrying a fragment of a given frame. When a JPEG 2000 image is in interlace format, the odd field and the corresponding even field MUST have the same timestamp value. Following the RTP specification [RFC3550], the initial value of the timestamp should be randomly chosen.

As for the clock rate, senders and receivers MUST support the 90kHz RTP timestamp rate, and MAY support other rates. RTP timestamp rates below 1000 Hz SHOULD NOT be used because they will result in insufficient resolution for RTP Control Protocol (RTCP) measurements based on the RTP timestamp, such as the interarrival

jitter. The clock rate MUST be negotiated at the start of the session. When using the Session Description Protocol (SDP), it MUST be expressed using the "rtpmap" attributes. If a non-90kHz clock rate is to be used, it is RECOMMENDED to present not only a preferable clock rate but also 90kHz clock rate with a different RTP payload type.

4.2. RTP Payload Header Format

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

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|tp |MHF|mh_id|T|     priority  |           tile number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|reserved       |             fragment offset                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3: RTP payload header format for JPEG 2000

tp (type): 2 bits

This field indicates how a JPEG 2000 image is scanned (progressive or interlace).

0: The payload is progressively scanned.

1: The payload is part of an odd field of an interlaced video frame. The height specified in the JPEG 2000 main header is half of the height of the entire displayed image. In a receiver, an odd field should be de-interlaced with the even field following it so that lines from each image are displayed alternately.

2: The payload is part of an even field of an interlaced video signal.

MHF (Main Header Flag): 2 bits

MHF indicates whether a main header or packet of a main header is in the RTP packet.

If there is no header, MHF has a value of 0. If there is just a part of a fragmented header, MHF has a value of 1. If there is the last part of a fragmented header, MHF has value of 2. If the whole header is in the packet, MHF has a value of 3.

MHF Value	Description
0	no main header in the payload
1	piece of fragmented header
2	last part of a fragmented header
3	a whole main header

Table 1: MHF Usage Values

mh_id (Main Header Identification): 3 bits

Main header identification value. This is used for JPEG 2000 main header recovery.

For implementations following only this specification, the sender SHOULD set this value to 0 and the receiver SHOULD ignore this field on processing.

T (Tile field invalidation flag): 1 bit

The T bit indicates whether the tile number field is valid or invalid. A sender MUST set the T bit to 1 when invalid and 0 when valid.

There are two cases where the tile number field is invalid:

* When an RTP packet holds only the main header. A sender cannot set any number in the tile number field, as no JPEG 2000 tile-part bitstream is included in the RTP packet.

* Multiple tile-parts are packed together in a single payload. If there are multiple tiles packed into a single payload, there is no meaning to assign a number to the tile number field.

priority: 8 bits

The priority field indicates the importance of the JPEG 2000 packet included in the payload. Typically, a higher priority value is set in the packets containing JPEG 2000 packets that contain the lower sub-bands.

For implementations following only this specification, the sender SHOULD set this value to 255 and the receiver SHOULD ignore this field on processing.

tile number: 16 bits

This field shows the tile number of the payload. This field is only valid when the T bit is 0. If the T bit is set to 1, the receiver MUST ignore this field.

R (Reserved): 8 bits

This bit is reserved for future use. Senders MUST set this to 0. Receivers MUST ignore this field.

fragment offset: 24 bits

This value MUST be set to the byte offset of the current payload in relation to the very beginning of each JPEG 2000 codestream (JPEG 2000 frame).

Byte offsets are calculated from the start of each JPEG 2000 codestream up to the current position where the current payload would fit into the complete 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. It is then possible to deliver layered video using multiple RTP sessions; the fragment offset might not start from 0 in some RTP sessions even if the packet is the first one received in the RTP session.

5. RTP Packetization

The sender must packetize the JPEG 2000 appropriately according to initial media type parameters and/or details from SDP offer/answer parameters.

A "packetization unit" is defined as either a JPEG 2000 main header, a tile-part header, or a JPEG 2000 packet.

First, a sender divides the JPEG 2000 codestream into packetization units by parsing the codestream or by getting information from the encoder, and packs the packetization units into RTP packets. A sender can put an arbitrary number of packetization units into an RTP packet, but it MUST preserve the codestream order. An example of this kind of RTP packet format is shown in Figure 4:

```
+------+-------+---------------+---------------+
|RTP   |payload| packetization | packetization |
|header|header | unit          | unit          |
+------+-------+---------------+---------------+
```

Figure 4: An example with multiple packetization units

If a packetization unit with headers (IP header, RTP header, and payload header) is larger than the MTU size, it MAY be fragmented. To pack a fragmented packetization unit, the fragmented unit MUST NOT be packed with the succeeding packetization unit within the same RTP packet. An example of this kind of RTP packet format is shown in Figure 5:

```
+------+-------+--------------------------------------------------+
|RTP   |payload| packetization unit fragment                      |
|header|header |                                                  |
+------+-------+--------------------------------------------------+
+------+-------+--------------------------------------------------+
|RTP   |payload| packetization unit fragment                      |
|header|header |                                                  |
+------+-------+--------------------------------------------------+
           .
           .
           .
+------+-------+-------------------------------------+
|RTP   |payload| end of packetization unit fragment |
|header|header |                                     |
+------+-------+-------------------------------------+
```

Figure 5: An example with a fragmented packetization unit

6. Media Type Registration

This registration uses the template defined in [RFC4288] and follows [RFC4855].

Type name: video

Subtype name: jpeg2000

Required parameters:

rate: The RTP timestamp clock rate. The default rate is 90000, but other rates MAY be specified. Rates below 1000 Hz SHOULD NOT be used.

sampling: A list of values specifying the color space of the payload data.

Acceptable values:

RGB: standard Red, Green, Blue color space.

BGR: standard Blue, Green, Red color space.

RGBA: standard Red, Green, Blue, Alpha color space.

BGRA: standard Blue, Green, Red, Alpha color space.

YCbCr-4:4:4: standard YCbCr color space; no subsampling.

YCbCr-4:2:2: standard YCbCr color space; Cb and Cr are subsampled horizontally by 1/2.

YCbCr-4:2:0: standard YCbCr color space; Cb and Cr are subsampled horizontally and vertically by 1/2.

YCbCr-4:1:1: standard YCbCr color space; Cb and Cr are subsampled vertically by 1/4.

GRAYSCALE: basically, a single component image of just multilevels of grey.

EXTENSION VALUE: Additional color samplings can be registered with the current listing of registered color samplings at: Color Sampling Registration Authority. Please refer to RTP Format for Uncompressed Video [RFC4175].

Optional parameters:

interlace: Interlace scanning. If the payload is in interlace format, the acceptable value is "1"; otherwise, the value should be "0". Each complete image forms, vertically, half the display. The tp value MUST properly specify the field the image represents: odd(tp=1) or even(tp=2). If this option is

not present, the payload MUST be in progressive format and the tp MUST be set to 0.

width: A parameter describing the maximum width of the video stream. This parameter MUST appear when height is present. Acceptable values: -- an integer value between 0 -- 4,294,967,295.

height: A parameter describing the maximum height of the video stream. This parameter MUST appear when width is present. Acceptable values: -- an integer value between 0 -- 4,294,967,295.

The receiver MUST ignore any unspecified parameters.

Encoding considerations:

This media type is framed and binary, see Section 4.8 of [RFC4288].

Security considerations: See Section 9 of this document.

Interoperability considerations:

The JPEG 2000 video stream is a sequence of JPEG 2000 still images. An implementation compliant with [JPEG2000Pt_1] can decode and attempt to display the encoded JPEG 2000 video stream.

Published specification: ISO/IEC 15444-1 | ITU-T Rec. T.800

Applications that use this media type:

video streaming and communication

Person and email address to contact for further information:

Eisaburo Itakura, Satoshi Futemma, Andrew Leung
Email: itakura@sm.sony.co.jp, satosi-f@sm.sony.co.jp, andrew@ualberta.net

Intended usage: COMMON

Restrictions on Usage:

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

Author/Change Controller:

Author:

Eisaburo Itakura, Satoshi Futemma, Andrew Leung
Email: itakura@sm.sony.co.jp, satosi-f@sm.sony.co.jp,
andrew@ualberta.net

Change controller:

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

7. SDP Parameters

7.1. SDP Mapping

The media type video/jpeg2000 string is mapped to fields in the Session Description Protocol (SDP) [RFC4566] as follows:

- The media name in the "m=" line of SDP MUST be video.

- The encoding name in the "a=rtpmap" line of SDP MUST be jpeg2000 (the subtype).

- The clock rate in the "a=rtpmap" line is set according to the "rate" parameter. Senders that wish to use a non-90kHz rate SHOULD also offer the same stream using a 90kHz timestamp rate with a different RTP payload type, allowing graceful fallback to 90kHz for compatibility.

- The REQUIRED parameter, "sampling", MUST be included in the "a=fmtp" line of SDP.

- The OPTIONAL parameters, if presented, MUST be included in the "a=fmtp" line of SDP.

These parameters are expressed as a media type string, in the form of a semicolon separated list of parameter=value pairs.

Therefore, an example of media representation in SDP using typical parameters is as follows:

```
m=video 49170/2 RTP/AVP 98
a=rtpmap:98 jpeg2000/90000
a=fmtp:98 sampling=YCbCr-4:2:0;width=128;height=128
```

An example for using non-90kHz timestamp is as follows:

```
m=video 49170/2 RTP/AVP 98 99
a=rtpmap:98 jpeg2000/27000000
a=rtpmap:99 jpeg2000/90000
a=fmtp:98 sampling=YCbCr-4:2:0;width=128;height=128
a=fmtp:99 sampling=YCbCr-4:2:0;width=128;height=128
```

7.2. Usage with the SDP Offer/Answer Model

When offering JPEG 2000 over RTP using SDP in an Offer/Answer model [RFC3264], the following rules and limitations apply:

- o All parameters MUST have an acceptable value for the parameter.

- o All parameters MUST correspond to the parameters of the payload.

- o The parameter "sampling" with an acceptable answer MUST appear in the offer and in the answer if accepted by the receiver. The receiver SHOULD do its best to handle the received codestream in the color space offered. If the receiver cannot handle the offered color space for whatever reason, it should reply with its preferred color space in the answer and gracefully end the session. Senders do not need to conform to the color space in the answer, but they should take note that the session ended due to color sampling issues.

- o For optional parameter "interlace", if this option is used, it MUST appear in the offer and, if accepted, it SHOULD appear in the answer. Receivers should do their best to handle interlace or progressive codestreams but, if for some reason, receivers cannot accommodate, receivers should reply with preferred settings in the answer, then gracefully end the session. Senders do not need to adjust settings upon this answer, but they should take note that the session ended due to interlace or progressive issues.

- o For optional parameters "width" and "height", the following applies:

  - * if "width" appears in the offer or answer, "height" MUST be present.

  - * if "height" appears in the offer or answer, "width" MUST be present.

- o Width and height should appear in the offer as the maximum dimensions the sender can offer. In the answer, it SHOULD represent the maximum the receiver can accommodate. If there is a

difference between the offer and answer, the sender should re-offer a new width and height and appropriately scale down the codestream for the receiver.

- In a multicast environment, [RFC1112] receivers should do their best to conform to parameters in the offer from the sender. Senders should use recommended settings in multicast environments and take note of answers. For width and height, the sender should accommodate to the lowest values it receives from all answers.

- Any unknown options in the offer should be ignored and deleted from the answer.

7.2.1. Examples

Example offer/answer exchanges are provided.

Alice offers YCbCr 4:2:2 color space, interlace image with 720-pixel width and 480-pixel height as below:

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.example
s=
c=IN IP4 host.example
t=0 0
m=video 49170 RTP/AVP 98
a=rtpmap:98 jpeg2000/90000
a=fmtp:98 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
```

Bob accepts YCbCr-4:2:2 color space, interlace image and replies:

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example
s=
c=IN IP4 host.example
t=0 0
m=video 49920 RTP/AVP 98
a=rtpmap:98 jpeg2000/90000
a=fmtp:98 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
```

7.2.2. Examples: Non-90kHz Timestamp

Example offer/answer exchanges, where an offerer wishes to use non-90kHz timestamp, are provided.

Alice offers an RTP payload type with 27MHz clock rate as well as with 90kHz clock rate, and each payload type includes: YCbCr 4:2:2 color space, interlace image, 720-pixel width and 480-pixel height.

She puts 27MHz clock rate attributes prior to 90kHz because she wants to use 27 MHz rather than 90kHz.

```
v=0
o=alice 2890844526 2890844526 IN IP4 host.example
s=
c=IN IP4 host.example
t=0 0
m=video 49170 RTP/AVP 98 99
a=rtpmap:98 jpeg2000/27000000
a=rtpmap:99 jpeg2000/90000
a=fmtp:98 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
a=fmtp:99 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
```

If Bob can accept 27MHz clock rate, he replies as below:

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example
s=
c=IN IP4 host.example
t=0 0
m=video 49920 RTP/AVP 98
a=rtpmap:98 jpeg2000/27000000
a=fmtp:98 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
```

If Bob doesn't accept 27MHz clock rate, he replies as below:

```
v=0
o=bob 2890844730 2890844731 IN IP4 host.example
s=
c=IN IP4 host.example
t=0 0
m=video 49920 RTP/AVP 99
a=rtpmap:99 jpeg2000/90000
a=fmtp:99 sampling=YCbCr-4:2:2; interlace=1; width=720;height=480
```

8. IANA Considerations

A new media subtype (video/jpeg2000) has been registered by IANA. For details, see Section 6 of this document.

9. Security Considerations

RTP packets using the payload format defined in this specification are subject to the security considerations discussed in the RTP specification [RFC3550], and in any applicable RTP profile. The main security considerations for the RTP packet carrying the RTP payload format defined within this memo are confidentiality, integrity, and

source authenticity. Confidentiality is achieved by encryption of the RTP payload. Integrity of the RTP packets is through the use of suitable cryptographic integrity protection mechanism. A cryptographic system may also allow the authentication of the source of the payload. A suitable security mechanism for this RTP payload format should provide confidentiality, integrity protection, and at least a source authentication method capable of determining whether or not an RTP packet is from a member of the RTP session.

Note that the appropriate mechanism to provide security to RTP and payloads following this memo may vary. It is dependent on the application, the transport, and the signaling protocol employed. Therefore, a single mechanism is not sufficient, although if suitable, the usage of SRTP [RFC3711] is recommended. Other mechanism that may be used are IPsec [RFC4301] and Transport Layer Security (TLS) [RFC5246] (RTP over TCP), but other alternatives may also exist.

10. Congestion Control

If Quality of Service (QoS) enhanced service is used, RTP receivers SHOULD monitor packet loss to ensure that the service that was requested is actually being delivered. If it is not, then they SHOULD assume that they are receiving best-effort service and behave accordingly.

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

11. References

11.1. Normative References

[JPEG2000Pt_1] ISO/IEC JTC1/SC29, ISO/IEC 15444-1 | ITU-T Rec. T.800, "Information Technology - JPEG 2000 Image Coding System - Part 1: Core Coding System", December 2000.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, February 2007.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, July 2006.

[RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, June 2002.

11.2. Informative References

[JPEG2000Pt_3] ISO/IEC JTC1/SC29, ISO/IEC 15444-1 | ITU-T Rec. T.800, "Information Technology - JPEG 2000 Image Coding System - Part 3: Motion JPEG 2000", July 2002.

[RFC5372] Leung, A., Futemma, S., and E. Itakura, "Payload Format for JPEG 2000 Video: Extensions for Scalability and Main Header Recovery", RFC 5372, October 2008.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC4175] Gharai, L. and C. Perkins, "RTP Payload Format for Uncompressed Video", RFC 4175, September 2005.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, August 1989.

Appendix A. Informative Appendix

A.1. Recommended Practices

As the JPEG 2000 coding standard is highly flexible, many different but compliant data streams may be produced and still be compliant JPEG 2000 codestreams.

The following is a set of recommendations set forth from our experience in developing JPEG 2000 and this payload specification. Implementations of this standard must handle all possibilities mentioned in this specification. The following is a listing of items an implementation may optimize.

Error Resilience Markers: The use of error resilience markers in the JPEG 2000 data stream is highly recommended in all situations. Error recovery with these markers is helpful to the decoder and saves external resources (e.g., markers such as RESET, RESTART, and ERTERM).

YCbCr Color Space: The YCbCr color space provides the greatest amount of compression in color with respect to the human visual system. When used with JPEG 2000, this color space can provide excellent visual results at low bit rates.

Progression Ordering: JPEG 2000 offers many different ways to order the final code stream to optimize the transfer with the presentation. We have found that the most useful codestream ordering is layer progression and resolution progression ordering.

Tiling and Packets: JPEG 2000 packets are formed regardless of the encoding method. The encoder has little control over the size of these JPEG 2000 packets as they may be large or small. Tiling splits the image into smaller areas and each is encoded separately. With tiles, the JPEG 2000 packet sizes are also reduced. When using tiling, almost all JPEG 2000 packet sizes are an acceptable size for transmission (i.e., smaller than the MTU size of most networks).

Sender Processing: There are no limitations as to how the sender should pack the payload. In general, the sender should pack headers separately from the rest of the codestream to make header recovery simple. Payloads should generally begin with a Start of Packet (SOP) marker and end with an End of Packet Header (EPH) marker for easier decoder processing.

A.2. Sample Headers in Detail

This section has various sample headers in various configurations for reference.

For reference, the payload header is as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|tp |MHF|mh_id|T|     priority  |           tile number         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|reserved       |             fragment offset                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: JPEG 2000 Payload Header

A.2.1. Sample 1: Progressive Image with Single Tile, 3500 Bytes (i.e., thumbnail)

First Packet: This packet will have the whole main header 210 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 3 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                       0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF4F FF51 002F 000 ....                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Header Sample 1-1 (First Packet)

Second Packet: This packet will have a tile header and the first tile part LLband 1500 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 3 |  0  |0|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                      210                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0000 0000 2DB3 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Header Sample 1-2 (Second Packet)

Third Packet: This packet will have the next part in the tile, no tile header 1500 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     1710                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|E841 4526 4556 9850 C2EA ...                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Header Sample 1-3 (Third Packet)

Fourth Packet: Last packet for the image 290 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     3210                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A55D 8B73 3B25 25C7 B9EB ...                           2FBE B153|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Header Sample 1-4 (4th Packet)

A.2.2. Sample 2: Image with 4 Tiles

First Packet: This packet will have the whole main header. 210 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 3 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                       0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF4F FF51 002F 000 ...                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Header Sample 2-1 (First Packet)

Second Packet: This packet will have a first tile part (tile 0) 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                      210                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0000 0000 0578 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Header Sample 2-2 (Second Packet)

Third Packet: This packet will have a second tile part (tile 1) 1423 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               1               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     1610                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0001 0000 058F 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Header Sample 2-3 (Third Packet)

Fourth Packet: This packet will have a third tile part (tile 2) 1355 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               2               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     3033                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0002 0000 054B 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Header Sample 2-4 (4th Packet)

Fifth Packet: This packet will have a fourth tile part (tile 3) 1290 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               3               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     4388                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0003 0000 050A 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Header Sample 2-5 (5th Packet)

A.2.3. Sample 3: Packing Multiple Tiles in Single Payload, Fragmented Header

First Packet: This packet will have the first part of the main header 110 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 1 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                       0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF4F FF51 002F 000 ...                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Header Sample 3-1 (First Packet)

Second Packet: This packet has the second part of the header 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 2 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                      110                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF64 00FF ...                                                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: Header Sample 3-2 (Second Packet)

Third Packet: This packet has two tiles, tile 0 and tile 1 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     1510                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0000 0000 02BC 0001 FF93 ...                         |
//                                                             //
|FF90 000A 0001 0000 02BC 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Header Sample 3-3 (Third Packet)

Fourth Packet: This packet has one tile, tile 2 1395 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 | 0 |  0  |0|      255      |               2               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     2910                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0002 0000 0573 0001 FF93 ...                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Header Sample 3-4 (4th Packet)

A.2.4. Sample 4: Interlace Image, Single Tile

First packet: This packet will have the whole main header for the odd field 210 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 3 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                       0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF4F FF51 002F 000 ...                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: Header Sample 4-1 (First Packet)

Second packet: This packet will have the first part of the odd field's tile 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                      210                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0000 0000 0578  0001 FF93 ...                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: Header Sample 4-2 (Second Packet)

Third packet: This packet will have the second part of the odd field's tile 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     1610                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|7F04 E708 27D9 D11D 22CB ...                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: Header Sample 4-3 (Third Packet)

Fourth packet: This packet will have the third part of the odd field's tile 1300 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 1 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     3010                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|98BD EC9B 2826 DC62 D4AB ...                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Header Sample 4-4 (4th Packet)

Fifth packet: This packet will have the whole main header for the even field 210 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 2 | 3 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                       0                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF4F FF51 002F 000 ...                                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: Header Sample 4-5 (5th Packet)

Sixth packet: This packet will have the first part of the even field's tile 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 2 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                      210                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|FF90 000A 0000 0000 0578  0001 FF93 ...                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 25: Header Sample 4-6 (6th Packet)

Seventh packet: This packet will have the second part of the even field's tile 1400 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 2 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     1610                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|626C 42F0 166B 6BD0 F8E1 ...                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 26: Header Sample 4-7 (7th Packet)

Eighth packet: This packet will have the third part of the even field's tile 1300 bytes

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 2 | 0 |  0  |1|      255      |               0               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       0       |                     3010                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|8114 41D5 18AB 4A1B ...                                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 27: Header Sample 4-8 (8th Packet)

Authors' Addresses

Satoshi Futemma
Sony Corporation
1-7-1 Konan
Minato-ku
Tokyo 108-0075
Japan

Phone: +81 3 6748-2111
EMail: satosi-f@sm.sony.co.jp
URI: http://www.sony.net/

Eisaburo Itakura
Sony Corporation
1-7-1 Konan
Minato-ku
Tokyo 108-0075
Japan

Phone: +81 3 6748-2111
EMail: itakura@sm.sony.co.jp
URI: http://www.sony.net/

Andrew Leung
Sony Corporation

EMail: andrew@ualberta.net

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

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

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

Intellectual Property

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

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

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