Network Working Group
Request for Comments: 2435
Obsoletes: 2035
Category: Standards Track

L. Berc
Digital Equipment Corporation
W. Fenner
Xerox PARC
R. Frederick
Xerox PARC
S. McCanne
Lawrence Berkeley Laboratory
P. Stewart
Xerox PARC
October 1998

RTP Payload Format for JPEG-compressed Video

Status of this Memo

Copyright Notice

Abstract

This memo describes the RTP payload format for JPEG video streams. The packet format is optimized for real-time video streams where codec parameters change rarely from frame to frame.

This document is a product of the Audio-Video Transport working group within the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group's mailing list at rem-conf@es.net and/or the author(s).

Changes from RFC 2035

Most of this memo is identical to RFC 2035. The changes made to the protocol are summarized in Appendix D.

Key Words

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 [9].

1. Introduction

The Joint Photographic Experts Group (JPEG) standard [1,2,3] defines a family of compression algorithms for continuous-tone, still images. This still image compression standard can be applied to video by compressing each frame of video as an independent still image and transmitting them in series. Video coded in this fashion is often called Motion-JPEG.

We first give an overview of JPEG and then describe the specific subset of JPEG that is supported in RTP and the mechanism by which JPEG frames are carried as RTP payloads.

The JPEG standard defines four modes of operation: the sequential DCT mode, the progressive DCT mode, the lossless mode, and the hierarchical mode. Depending on the mode, the image is represented in one or more passes. Each pass (called a frame in the JPEG standard) is further broken down into one or more scans. Within each scan, there are one to four components, which represent the three components of a color signal (e.g., "red, green, and blue", or a luminance signal and two chrominance signals). These components can be encoded as separate scans or interleaved into a single scan.

Each frame and scan is preceded with a header containing optional definitions for compression parameters like quantization tables and Huffman coding tables. The headers and optional parameters are identified with "markers" and comprise a marker segment; each scan appears as an entropy-coded bit stream within two marker segments. Markers are aligned to byte boundaries and (in general) cannot appear in the entropy-coded segment, allowing scan boundaries to be determined without parsing the bit stream.

Compressed data is represented in one of three formats: the interchange format, the abbreviated format, or the table-specification format. The interchange format contains definitions for all the tables used by the entropy-coded segments, while the abbreviated format might omit some assuming they were defined out-of-band or by a "previous" image.

The JPEG standard does not define the meaning or format of the components that comprise the image. Attributes like the color space and pixel aspect ratio must be specified out-of-band with respect to the JPEG bit stream. The JPEG File Interchange Format (JFIF) [4] is a de-facto standard that provides this extra information using an application marker segment (APP0). Note that a JFIF file is simply a JPEG interchange format image along with the APP0 segment. In the case of video, additional parameters must be defined out-of-band (e.g., frame rate, interlaced vs. non-interlaced, etc.).

While the JPEG standard provides a rich set of algorithms for flexible compression, cost-effective hardware implementations of the full standard have not appeared. Instead, most hardware JPEG video codecs implement only a subset of the sequential DCT mode of operation. Typically, marker segments are interpreted in software (which "re-programs" the hardware) and the hardware is presented with a single, interleaved entropy-coded scan represented in the YUV color space.

The scan contains an ordered sequence of Minimum Coded Units, or MCUs, which are the smallest group of image data coded in a JPEG bit stream. Each MCU defines the image data for a small rectangular block of the output image.

Restart markers in the JPEG data denote a point where the decoder should reset its state. As defined by JPEG, restart markers are the only type of marker that may appear embedded in the entropy-coded segment, and they may only appear on an MCU boundary. A "restart interval" is defined to be a block of data containing a restart marker followed by some fixed number of MCUs. An exception is made for the first restart interval in each frame, which omits the initial restart marker and just begins with the MCU data. When these markers are used, each frame is composed of some fixed number of back-to-back restart intervals.

2. JPEG Over RTP

To maximize interoperability among hardware-based codecs, we assume the sequential DCT operating mode [1,Annex F] and restrict the set of predefined RTP/JPEG "type codes" (defined below) to single-scan, interleaved images. While this is more restrictive than even baseline JPEG, many hardware implementation fall short of the baseline specification (e.g., most hardware cannot decode non-interleaved scans).

In practice, most of the table-specification data rarely changes from frame to frame within a single video stream. Therefore RTP/JPEG data is represented in abbreviated format, with all of the tables omitted from the bit stream where possible. Each frame begins immediately with the (single) entropy-coded scan. The information that would otherwise be in both the frame and scan headers is represented entirely within the RTP/JPEG header (defined below) that lies between the RTP header and the JPEG payload.

While parameters like Huffman tables and color space are likely to remain fixed for the lifetime of the video stream, other parameters should be allowed to vary, notably the quantization tables and image size (e.g., to implement rate-adaptive transmission or allow a user to adjust the "quality level" or resolution manually). Thus explicit fields in the RTP/JPEG header are allocated to represent this information. Since only a small set of quantization tables are typically used, we encode the entire set of quantization tables in a small integer field. Customized quantization tables are accommodated by using a special range of values in this field, and then placing the table before the beginning of the JPEG payload. The image width and height are encoded explicitly.

Because JPEG frames are typically larger than the underlying network's maximum packet size, frames must often be fragmented into several packets. One approach is to allow the network layer below RTP (e.g., IP) to perform the fragmentation. However, this precludes rate-controlling the resulting packet stream or partial delivery in the presence of loss, and frames may be larger than the maximum network layer reassembly length (see [10] for more information). To avoid these limitations, RTP/JPEG defines a simple fragmentation and reassembly scheme at the RTP level.

3. RTP/JPEG Packet Format

The RTP timestamp is in units of 90000Hz. The same timestamp MUST appear in each fragment of a given frame. The RTP marker bit MUST be set in the last packet of a frame.

3.1. JPEG header

Each packet contains a special JPEG header which immediately follows the RTP header. The first 8 bytes of this header, called the "main JPEG header", are 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type-specific |              Fragment Offset                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |       Q       |     Width     |     Height    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

All fields in this header except for the Fragment Offset field MUST remain the same in all packets that correspond to the same JPEG frame.

A Restart Marker header and/or Quantization Table header may follow this header, depending on the values of the Type and Q fields.

3.1.1. Type-specific: 8 bits

Interpretation depends on the value of the type field. If no interpretation is specified, this field MUST be zeroed on transmission and ignored on reception.

3.1.2. Fragment Offset: 24 bits

The Fragment Offset is the offset in bytes of the current packet in the JPEG frame data. This value is encoded in network byte order (most significant byte first). The Fragment Offset plus the length of the payload data in the packet MUST NOT exceed 2^24 bytes.

3.1.3. Type: 8 bits

The type field specifies the information that would otherwise be present in a JPEG abbreviated table-specification as well as the additional JFIF-style parameters not defined by JPEG. Types 0-63 are reserved as fixed, well-known mappings to be defined by this document and future revisions of this document. Types 64-127 are the same as types 0-63, except that restart markers are present in the JPEG data and a Restart Marker header appears immediately following the main JPEG header. Types 128-255 are free to be dynamically defined by a session setup protocol (which is beyond the scope of this document).

3.1.4. Q: 8 bits

The Q field defines the quantization tables for this frame. Q values 0-127 indicate the quantization tables are computed using an algorithm determined by the Type field (see below). Q values 128-255 indicate that a Quantization Table header appears after the main JPEG header (and the Restart Marker header, if present) in the first packet of the frame (fragment offset 0). This header can be used to explicitly specify the quantization tables in-band.

3.1.5. Width: 8 bits

This field encodes the width of the image in 8-pixel multiples (e.g., a width of 40 denotes an image 320 pixels wide). The maximum width is 2040 pixels.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.