RTP payload Format for JVT Video

Status of this Memo

This document is an Internet-Draft and is in full conformance with 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 material 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.txt

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

Abstract

This memo describes an RTP Payload format for the ITU-T Recommendation H.264 codec. This codec was designed as a joint project of the ITU-T SG 16 VCEG, and the ISO/IEC JTC1/SC29/WG11 MPEG groups. The most up-to-date draft of the video codec was specified in late August 2002, is due for revision in late October 2002, and is available for public review [2]. Final versions carry the denomination H.264 and ISO/IEC 14496-10 and are technically identical.
1. The JVT codec

This memo specifies an RTP payload specification for a new video codec that is currently under development by the Joint Video Group (JVT), which is formed of video coding experts of MPEG and the ITU-T. After the likely approval by the two parent bodies, the codec specification will have the status of the ITU-T Recommendation H.264 and become part of the MPEG-4 specification (ISO/IEC 14496 Part 10). The current project timeline of the JVT project is such that a technically frozen specification (pending bug fixes) was finalized in July 2002 in the form of an ISO/IEC Final Committee Draft (FCD). In October, some editorial changes will be made, and a few technical changes can also be expected. However, it is believed that only very few, if any, technical details will be changed that directly affect this draft.

Before JVT was formed in late 2001, this project used the ITU-T project name H.26L and the JVT project inherited all the technical concepts of the H.26L project.

The JVT video codec has a very broad application range that covers the whole range from low bit rate Internet Streaming applications to HDTV broadcast and Digital Cinema applications with near lossless coding. Most, if not all, relevant companies in all of these fields (including TV broadcast) have participated in the standardization, which gives hope that this wide application range is more than an illusion and may materialize, probably in a relatively short time frame. The overall performance of the JVT codec is as such that bit rate savings of 50% or more, compared to the current state of technology, are reported. Digital Satellite TV quality, for example, was reported to be achievable at 1.5 Mbit/s, compared to the current operation point of MPEG 2 video at around 3.5 Mbit/s [1].

The codec specification [2] itself distinguishes conceptually between a video coding layer (VCL), and a network abstraction layer (NAL). The VCL contains the signal processing functionality of the codec, things such as transform, quantization, motion search/compensation, and the loop filter. It follows the general concept of most of today’s video codecs, a macroblock-based coder that utilizes inter picture prediction with motion compensation, and transform coding of the residual signal. The output of the VCL are slices: a bit string that contains the macroblock data of an integer number of macroblocks, and the information of the slice header (containing the spatial address of the first macroblock in the slice, the initial quantization parameter, and similar). Macroblocks in slices are ordered in scan order unless a different macroblock allocation is specified, using the so-called Flexible Macroblock Ordering syntax. In-picture prediction is used only within a slice.

The NAL encapsulates the slice output of the VCL into Network Abstraction Layer Units (NALUs), which are suitable for the transmission over packet networks or the use in packet oriented multiplex environments. JVT’s Annex B defines an encapsulation
process to transmit such NALUs over byte-stream oriented networks. In the scope of this memo Annex B is not relevant.

Neither VCL nor NAL are claimed to be media or network independent - the VCL needs to know transmission characteristics in order to appropriately select the error resilience strength, slice size, etc., whereas the NAL needs information like the importance of a bit string provided by the VCL to select the appropriate application layer protection.

Internally, the NAL uses NAL Units or NALUs. A NALU consists of a one-byte header and the payload byte string. The header co-serves as the RTP payload header and indicates the type of the NALU, the (potential) presence of bit errors in the NALU payload, and information regarding the relative importance of the NALU for the decoding process. This RTP payload specification is designed to be unaware of the bit string in the NALU payload.

One of the main properties of the JVT codec is the possibility of the use of Reference Picture Selection. For each macroblock the reference picture to be used can be selected independently. The reference pictures may be used in a first-in, first-out fashion, but it is also possible to handle the reference picture buffers explicitly. A consequence of this new feature (it was available before only in H.263++ [3]) is the complete decoupling of the transmission time, the decoding time, and the sampling or presentation time of slices and pictures. For this reason, the handling of the RTP timestamp requires some special considerations for those NALUs for which the sampling or presentation time is not defined, or, at transmission time, unknown.

2. Changes relative to draft-wenger-avt-rtp-jvt-01.txt

[This section will be removed in a future version of this draft.]

2.1. Status of the JVT standardization, and recent changes to JVT

Since the last draft, JVT has met twice and each time a new JVT working draft was produced. The latest JVT working draft is currently in the second stage of the ISO/IEC approval process, the ballot on the so-called Final Committee Draft. Procedural provisions are taken by interested ISO/IEC members to ensure that changes relative to this draft are still possible, even after the ballot.

The meetings brought a lot of changes in the VCL, which do not have a direct influence to this memo. However, there were also numerous changes introduced to the NAL. Most of these changes can be considered bug fixes or cleanups that re-established the clean NAL design. In particular, the unreasonably high number of slice types were again reduced to the pre-Fairfax design (as presented in the Minneapolis IETF), and the picture header concept with its redundant carriage mechanism was removed.
Newly introduced was a mechanism that allows to signal the relative importance of a NALU for the decoding process. A two-bit field indicates the importance of a NALU. A value of 00 indicates that the decoding of the NALU is not necessary to maintain the integrity of the reference pictures. Values above 0 imply that the NALU is necessary for maintaining the integrity of the reference pictures. However, the impact of the loss, as determined by the encoder, is the higher the bigger the value of the field is. Intelligent network elements can use this information to discard NALUs in a controlled manner in order to produce the best possible picture at a given bit rate.

2.2. Changes relative to draft-wenger-avt-rtp-jvt-01.txt

This memo contains two significant changes relative to the previous I-D. The first change is the alignment with the current JVT WD, in particular with respect to the NALU types and the priority field in the NALU header. The second change was discussed in Yokohama and is concerned with the length of the timestamp offset field in the MTAP. In Yokohama it was felt that more flexibility is needed. Hence, now a total of 4 MTAPs are introduced, called MTAP8, MTAP16, MTAP24, and MTAP32, which differ from each other only by the length of the timestamp offset.

3. Scope

This payload specification can only be used to carry the "naked" JVT NALU stream over RTP. Likely, the first applications of a Standard Track RFC resulting from this draft will be in the conversational multimedia field, video telephone or video conference. The draft is not intended for the use in conjunction with the Byte Stream format of Annex B of the JVT working draft, the MPEG 4 system layer [4] or other multiplexing schemes.

4. NAL basics

Tutorial information on the NAL design can be found in [5], [6] and [14]. For the precise definition of the NAL it is referred to [2]. This section tries to provide a very short overview of the concepts used.

4.1. Parameter Set Concept

One very fundamental design concept of the JVT codec is to generate self-contained packets, to make mechanisms such as the header duplication of RFC2429 [7] or MPEG-4’s HEC [8] unnecessary. The way how this was achieved is to decouple information that is relevant for more than one slice from the media stream. This higher layer meta information should be sent reliably, asynchronously and in advance from the RTP packet stream that
contains the slice packets. The combination of the higher level parameters is called a Parameter Set. The Parameter Set contains information such as

- picture size,
- display window,
- optional coding modes employed,
- macroblock allocation map,
- and others.

In order to be able to change picture parameters (such as the picture size), without having the need to transmit Parameter Set updates synchronously to the slice packet stream, the encoder and decoder can maintain a list of more than one Parameter Set. Each slice header contains a codeword that indicates the Parameter Set to be used.

This mechanism allows to decouple the transmission of the Parameter Sets from the packet stream, and transmit them by external means, e.g. as a side effect of the capability exchange, or through a (reliable or unreliable) control protocol. It may even be possible that they get never transmitted but are fixed by an application design specification.

Although, conceptually, the Parameter Set updates are not designed to be sent in the synchronous packet stream, this memo contains means to convey them in the RTP packet stream.

4.2. Network Abstraction Layer Packet (NALU) Types

All NALUs consist of a single NALU Type octet, which also serves as the payload header. The payload of a NALU follows immediately.

The NALU type octet has the following format:

```
+------------------+
|0|1|2|3|4|5|6|7|
+------------------+
|F|NSI|  Type   |
+------------------+
```

F: 1 bit
The Forbidden bit, when zero, indicates a bit error free NAL unit. The JVT specification declares a value of 1 as a syntax violation. Hence, when set, the decoder is advised that bit errors may be present in the payload or in the NALU type octet. A prudent reaction of decoders that are incapable of handling bit errors is to discard such packets.

NSI: 2 bits
NAL Storage IDC. A value of 00 indicates that the content of the NALU is not used to reconstruct stored pictures (that can be
used for future reference). Such NALUs can be discarded without risking the integrity of the reference pictures. Values above 00 indicate that the decoding of the NALU is required to maintain the integrity of the reference pictures. Furthermore, values above 00 indicate the relative transport priority, as determined by the encoder. Intelligent network elements can use this information to protect more important NALUs better than less important NALUs. 11 is the highest transport priority, followed by 10, then by 01 and, finally, 00 is the lowest.

Type: 5 bits
The NAL Unit payload type as defined in table 7.1 of [2].

For a reference of all currently defined NALU types and their semantics please refer to section 7.1 in [2].

4.3. Aggregation Packets
Aggregation packets are the packet aggregation scheme of this payload specification. The scheme is introduced to reflect the dramatically different MTU sizes of two target networks -- wireline IP networks (with an MTU size that is often limited by the Ethernet MTU size -- roughly 1500 bytes), and IP or non-IP (e.g. H.324/M) based wireless networks with preferred transmission unit sizes of 254 bytes or less. In order to prevent media transcoding between the two worlds, and to avoid undesirable packetization overhead, a packet aggregation scheme is introduced.

Two types of Aggregation packets are defined by this specification:

- **Single-Time Aggregation Packet (STAP)** aggregate NALUs with identical NALU-time.
- **Multi-Time Aggregation Packets (MTAP)** aggregate NALUs with potentially differing NALU-time. Four different MTAPs are defined that differ in the length of the NALU timestamp offset.

The term NALU-time is defined as the value the RTP timestamp would have if that NALU would be transported in its own RTP packet.

MTAPs and STAP share the following packetization rules:

The NSI MUST be set to the maximum of the NSIs of all the NALUs to be aggregated.
The Type field of the NALU type octet MUST be set to the appropriate value as indicated in table xxx. The F bit MUST be cleared if all F bits of the aggregated NALUs are zero, otherwise it MUST be set.

**Table xxx: Type field for STAP and MTAPs**

<table>
<thead>
<tr>
<th>Type</th>
<th>Packet</th>
<th>Timestamp offset field length (in bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x18</td>
<td>STAP</td>
<td>0</td>
</tr>
<tr>
<td>0x19</td>
<td>MTAP8</td>
<td>8</td>
</tr>
<tr>
<td>0x20</td>
<td>MTAP16</td>
<td>16</td>
</tr>
<tr>
<td>0x21</td>
<td>MTAP24</td>
<td>24</td>
</tr>
</tbody>
</table>

The Marker bit in the RTP header MUST be set to the value the marker bit of the last NALU of the aggregated packet would have if it were transported in its own RTP packet.

The NALU Payload of an aggregation packet consists of one or more aggregation units. See section 4.3.1 and 4.3.2 for the two different types of aggregation units. An aggregation packet can carry as many aggregation units as necessary, however the total amount of data in an aggregation packet obviously MUST fit into an IP packet, and the size SHOULD be chosen such that the resulting IP packet is smaller than the MTU size.

4.3.1. Single-Time Aggregation Packet

Single-Time Aggregation Packet (STAP) SHOULD be used whenever aggregating NALUs that share the same NALU-time.

The NALU payload of an STAP consists of Single-Picture Aggregation units.

A Single-Picture Aggregation Unit consists of 16-bit unsigned size information that indicates the size of the following NALU in bytes (excluding these two octets, but including the NALU type octet of the NALU), followed by the NALU itself including its NALU type byte.

4.3.2. Multi-Time Aggregation Packets (MTAPs)

An MTAP has a similar architecture as an STAP. It consists of the NALU header byte and one or more Multi-Picture Aggregation Units. The choice between the different MTAP fields is application dependent -- the larger the timestamp offset is the higher is the flexibility of the MTAP, but the higher is also the overhead.

This Memo does not specify how the NALUs within an MTAP are ordered. In most cases, the natural "decoding order" SHOULD be used, in particular in conjunction with bi-predicted pictures that use a forward reference picture. However, all other NALU ordering schemes that are legal in JVT video MAY be used as well.

Four different Multi-Time Aggregation Unit are defined in this specification. They all consist of 16 bits unsigned size information of the following NALU (same as the size information of in the STAP). These 16 bits are followed by n bits of timing information for this NALU, whereby n can be 8, 16, 24, or 32. The timing information field MUST be set so that the RTP timestamp of an RTP packet of each NALU in the MTAP (the NALU-time) can be generated by subtracting the timing information from the RTP timestamp of the MTAP.

For the "latest" multi-picture Aggregation Unit in an MTAP the timing offset MUST be zero. Hence, the RTP timestamp of the MTAP itself is identical to the latest NALU-time.
5. RTP Packetization Process

The RTP packetization process of the JVT codec is straightforward and follows the general principles outlined in RFC1889. When using one NALU per RTP packet, the RTP payload consists of the bit buffer containing the NALU. The RTP payload (and the settings for some RTP header bits) for aggregation packets were already defined in section 4.3 above. There is no specific RTP payload header – the NALU type byte double-functions in this task. The RTP header information is set as follows:

Timestamp: 32 bits
The RTP timestamp is set to the presentation/sampling timestamp of the content. If the NALU has no own timing properties (e.g. PSIs, SEI), or if the presentation/sampling time is unknown, the RTP timestamp is set to the RTP timestamp of the last transmitted RTP packet in the session. The setting of the RTP Timestamp for MTAPs is defined in section 4.3.2 above.

Marker bit (M): 1 bit
Set for the very last packet of the picture indicated by the RTP timestamp, in line with the normal use of the M bit and to allow an efficient playout buffer handling. Decoders MAY use this bit as an early indication of the last packet of a coded picture, but MUST not rely on this property because the last packet of the picture may get lost, and because the use of MTAPs does not always preserve the M bit.

Sequence No (Seq): 16 bit
Increased by one for each sent packet. Set to a random value during startup as per RFC1889

Version (V): 2 bits
set to 2

Padding (P): 1 bit
set to 0

Extension (X): 1 bit
set to 0

Payload Type (PT): 8 bits
established dynamically during connection establishment

All other RTP header fields are set as per RFC1889.

6. Packetization Rules

Two cases of packetization rules have to be distinguished by the possibility to put packets belonging to more than a single picture into a single aggregated packet (using STAPs or MTAPs).
6.1. Unrestricted Mode (Multiple Picture Model)

This mode MAY be supported by some receivers. Usually, the capability of a receiver to support this mode is indicated by one of the profiles of the JVT codec (this is not yet defined in [2]). The following packetization rules MUST be enforced by the sender:

- Single slice packets belonging to the same picture (and hence share the same RTP timestamp value) MAY be sent in any order, although, for delay critical systems, they SHOULD be sent in their original coding order to minimize the delay. Note that the coding order is not necessarily the scan order, but the order the NAL packets become available to the RTP stack.

- Both MTAPs and STAPs MAY be used.

- SEI packets MAY be sent anytime.

- PSIs MUST NOT be sent in an RTP session whose Parameter Sets were already changed by control protocol messages during the lifetime of the RTP session. If PSIs are allowed by this condition, they MAY be sent at any time.

- All NALUs of all types MAY be mixed freely, provided that above rules are obeyed. In particular, it is allowed to mix slices in data-partitioned and single-slice mode.

- Network elements MAY convert multiple RTP packets carrying individual NALUs into one aggregated RTP packet, convert an aggregated RTP packet into several RTP packets carrying individual NALUs, or mix both concepts. However, when doing so they SHOULD take into account at least the following parameters: path MTU size, unequal protection mechanisms (e.g. through packet duplication, packet-based FEC carried by RFC2198, especially for header and Type A Data Partitioning packets), bearable latency of the system, and buffering capabilities of the receiver.

- NALUs of all types MAY be conveyed as aggregation units of an STAP or MTAP rather than individual RTP packets. Special care SHOULD be taken (particularly in gateways) to avoid more than a single copy of identical NALUs in a single STAP/MTAP in order to avoid unnecessary data transfers without any improvements of QoS.

6.2. Restricted Mode (Single Picture Model)

This mode MUST be supported by all receivers. It is primarily intended for low delay applications. Its main difference from the Unrestricted Mode is to forbid the packetization of data belonging to more than one picture in a single RTP packet. Hence, MTAPs MUST
NOT be used. The following packetization rules MUST be enforced by the sender:

- All rules of the Unrestricted Mode above, with the following additions
- only STAPs MAY be used, MTAPs MUST NOT be used. This implies that aggregated packets MUST NOT include slices or data partitions belonging to different pictures.

7. De-Packetization Process

The de-packetization process is implementation dependent. Hence, the following description should be seen as an example of a suitable implementation. Other schemes MAY be used as well. Optimizations relative to the described algorithms are likely possible.

The general concept behind these de-packetization rules is to collect all packets belonging to a picture, bringing them into a reasonable order, discard anything that is unusable, and pass the rest to the decoder. Aggregation packets are handled by unloading their payload into individual RTP packets carrying NALUs. Those NALUs are processed as if they were received in separate RTP packets, in the order they were arranged in the Aggregation Packet.

The following de-packetization rules MAY be used to implement an operational JVT de-packetizer:

- NALUs are presented to the JVT decoder in the order of the RTP sequence number.
- NALUs carried in an Aggregation Packet are presented in their order in the Aggregation packet. All NALUs of the Aggregation packet are processed before the next RTP packet is processed.
- Intelligent RTP receivers (e.g. in Gateways) MAY identify lost DPAs. If a lost DPA is found, the Gateway MAY decide not to send the DPB and DPC partitions, as their information is meaningless for the JVT Decoder. In this way a network element can reduce network load by discarding useless packets, without parsing a complex bit stream.
- Intelligent receivers MAY discard all packets that have the Disposable Flag set. However, they SHOULD process those packets if possible, because the user experience may suffer if the packets are discarded.

8. MIME Considerations

This section is to be completed later.
9. Security Considerations

So far, no security considerations beyond those of RFC1889 have been identified.

Currently, the JVT CD does not allow carrying any type of active payload. However, the inclusion of a "user data" mechanism is under consideration, which could potentially be used for mechanisms such as remote software updates of the video decoder and similar tasks.

10. Informative Appendix: Application Examples

This payload specification is very flexible in its use, to cover the extremely wide application space that is anticipated for the JVT codec. However, such a great flexibility also makes it difficult for an implementer to decide on a reasonable packetization scheme. Some information how to apply this specification to real-world scenarios is likely to appear in the form of academic publications and a Test Model in the near future. However, some preliminary usage scenarios should be described here as well.

10.1. Video Telephony, no Data Partitioning, no packet aggregation

The RTP part of this scheme is implemented and tested (though not the control-protocol part, see below).

In most real-world video telephony applications, the picture parameters such as picture size or optional modes never change during the lifetime of a connection. Hence, all necessary Parameter Sets (usually only one) are sent as a side effect of the capability exchange/announcement process. An example for such a capability exchange with an SDP-like syntax can be found in [9], but other schemes such as ASN.1 are possible as well. Since all necessary Parameter Set information is established before the RTP session starts, there is no need for sending any PSIs. Data Partitioning is not used either. Hence, the RTP packet stream consists basically of NALUs that carry single slices of video information.

The size of those single-slice NALUs is chosen by the encoder such that they offer the best performance. Often, this is done by adapting the coded slice size to the MTU size of the IP network. For small picture sizes this may result in a one-picture-per-one-packet strategy. The loss of packets and the resulting drift-related artifacts are cleaned up by Intra refresh algorithms.

10.2. Video Telephony, Interleaved Packetization using Packet Aggregation
This scheme allows better error concealment and is widely used in H.263 based designs using RFC2429 packetization. It is also implemented and good results were reported [5].

The source picture is coded by the VCL such that all MBs of one MB line are assigned to one slice. All slices with even MB row addresses are combined into one STAP, and all slices with odd MB row addresses into another STAP. Those STAPs are transmitted as RTP packets. The establishment of the Parameter Sets is performed as discussed above.

Note that the use of STAPs is essential here, because the high number of individual slices (18 for a CIF picture) would lead to unacceptably high IP/UDP/RTP header overhead (unless the source coding tool FMO is used, which is not assumed in this scenario). Furthermore, some wireless video transmission systems, such as H.324M and the IP-based video telephony specified in 3GPP, are likely to use relatively small transport packet size. For example, a typical MTU size of H.223 AL3 SDU is around 100 bytes [10]. Coding individual slices according to this packetization scheme provides a further advantage in communication between wired and wireless networks, as individual slices are likely to be smaller than the preferred maximum packet size of wireless systems. Consequently, a gateway can convert the STAPs used in a wired network to several RTP packets with only one NALU that are preferred in a wireless network and vice versa.

10.3. Video Telephony, with Data Partitioning

This scheme is implemented and was shown to offer good performance especially at higher packet loss rates [5]. Data Partitioning is known to be useful only when some form of unequal error protection is available. Normally, in single-session RTP environments, even error characteristics are assumed -- statistically, the packet loss probability of all packets of the session is the same. However, there are means to reduce the packet loss probability of individual packets in an RTP session. One simple way is known as Packet Duplication: simply send the to-be-protected packet twice, with the same sequence number. If both packets survive, the receiver will assume a packet duplication by UDP and discard one of the two packets. Other means of unequal protection within the same RTP session include the use of RFC 2198 [11] (for this application it is essentially a packet duplication process as well, with some saved bytes for the second RTP header), or packet-based Forward Error Correction [12] carried in RFC2198.

The implemented software uses the simple packet duplication process to increase the probability of all DPA NALUs. The incurred overhead is substantial, but in the same order of magnitude as the number of bits that have otherwise been spent for intra information. However, this mechanism is not adding any delay to the system.

Again, the complete Parameter Set establishment is performed through control protocol means.
10.4. MPEG-2 Transport to RTP Gateway

This example is not implemented completely, but the basic mechanisms are part of the interim file format the JVT group uses and, hence, well tested.

When using JVT video in satellite/cable broadcast environments, there is no control protocol available that can be used for the transmission of Parameter Sets. Furthermore, a receiver has to be able to "tune" into an ongoing packet stream at any time, without much delay and artifacts. For this reason, PSIs that contain all Parameter Set information are included in the packet stream at any Instantaneous Decoder Refresh Point (which are similar to Key Frames in earlier coding standards). IDERP packets are used to signal these "key frames" so that a decoder can most easily determine where to start in its decoding process.

Since the byte stream format used in satellite/cable broadcast environments does not include timing information in the video stream, the gateway needs to use external timing information (e.g. from the MPEG-2 system layer) to generate the RTP timestamp. Please note that this timestamp is also a 90 kHz clock -- hence, in most cases, the conversion should be relatively simple.

The simplest possible MPEG-2 transport to RTP gateway could take the NALUs as they come from the MPEG-2 transport stream (after de-framing), and send them, each NALU in one RTP packet, with increasing RTP sequence numbers. However, less than perfect packet loss rates would lead to a very poor performance of such a system. However, a Gateway could use the protection mechanisms discussed above to unequally protect the most important packets, e.g. all PSIs (very strong protection) IDERPVs (weak protection), and transmit everything else best effort. The Gateway can do this without parsing the bit stream, by simply using the NALU type byte. A more sophisticated Gateway may be able to combine some small NALUs to a big STAP or MTAP in order to save the bytes used for the IP/UDP/RTP headers.

A similar mechanism is, of course, also possible in H.320 to RTP gateways. Here, however, the system environment does not include any timing information, and exact presentation timing is carried in the form of SEIs. Hence, in the H.320 to IP data path, the gateway has the additional duty to filter out SEIs containing timing information and setting the RTP timestamp of the following video packets accordingly. In the reverse direction, SEIs need to be generated using the RTP timestamp as a guideline.

10.5. Low-Bit-Rate Streaming

This scheme has been implemented with H.263 and gave good results [13]. There is no technical reason why similarly good results could not be achievable using the JVT codec.
In today's Internet streaming, some of the offered bit-rates are relatively low in order to allow terminals with dial-up modems to access the content. In wired IP networks, relatively large packets, say 500 - 1500 bytes, are preferred to smaller and more frequently occurring packets in order to reduce network congestion. Moreover, use of large packets decreases the amount of RTP/UDP/IP header overhead. For low-bit-rate video, the use of large packets means that sometimes up to few pictures should be encapsulated in one packet.

However, loss of such a packet would have drastic consequences in visual quality, as there is practically no other way to conceal a loss of an entire picture than to repeat the previous one. One way to construct relatively large packets and maintain possibilities for successful loss concealment is to construct MTAPs that contain slices from several pictures in an interleaved manner. An MTAP should not contain spatially adjacent slices from the same picture or spatially overlapping slices from any picture. If a packet is lost, it is likely that a lost slice is surrounded by spatially adjacent slices of the same picture and spatially corresponding slices of the temporally previous and succeeding pictures. Consequently, concealment of the lost slice is likely to succeed relatively well.

11. Open Issues
There are several open issues on which the authors would like to receive opinions. They are listed below.

We have now five xTAPs, with 0, 8, 16, 24, and 32 bit timestamps offset per aggregation unit. This is per response to the last AVT meeting. However, neither the 8 bit nor the 32 bit offset make a lot of sense. JVT does not allow frame rates that make 8 bit offsets useful, and a 32 bit offset at 90 kHz is only necessary for frame intervals longer than 186 seconds. Hence, we believe we should remove the 8 bit and 32 bit timestamp offsets to save the two codepoints.

Since JVT will likely be approved as the advanced video codec of MPEG-4, it may be desirable to align this payload specification with other payload specifications for MPEG 4. The authors of this I-D and some authors of the MPEG-4 packetization I-Ds are discussing the issue, and there is a chance that in the future changes to this I-D will be proposed to AVT to reflect the outcome of these discussions.

12. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published
13. Bibliography


[2] JVT Joint Final Committee Draft, available from


Author’s Addresses

Stephan Wenger                                      Phone: +49-172-300-0813
TU Berlin / Teles AG                                Email: stewe@cs.tu-berlin.de
Franklinstr. 28-29                                  D-10587 Berlin
Germany

Thomas Stockhammer                                 Phone: +49-89-28923474
Institute for Communications Eng.                   Email: stockhammer@ei.tum.de
Munich University of Technology                     D-80290 Munich
Germany

Miska M. Hannuksela                                 Phone: +358 40 5212845
Nokia Corporation                                  Email: miska.hannuksela@nokia.com
P.O. Box 68                                         33721 Tampere
Finland