# H.26L/JVT CODING NETWORK ABSTRACTION LAYER AND IP-BASED TRANSPORT

*Thomas Stockhammer*
Institute for Communications Engineering
Munich University of Technology
80290 Munich, Germany

*Miska M. Hannuksela*
Nokia Mobile Software
Nokia Corporation
Tampere, Finland

*Stephan Wenger*
Communication and Operating Systems
Technical University Berlin
10587 Berlin, Germany

## ABSTRACT

The JVT/H.26L video coding scheme conceptually consists of a Video Coding Layer (VCL) responsible mainly for coding efficiency and a Network Abstraction Layer (NAL) that supports video specific transport features for a variety of networks. This paper describes the H.26L/JVT NAL in general, including the parameter set concept and the transport over packet-based and bit-stream oriented networks. Encapsulation of coded video data to RTP/UDP/IP transport is discussed, and applications of H.26L in fixed and wireless IP-based transmission environments are presented.

## 1 INTRODUCTION

Since 1997, the ITU-T's Video Coding Experts Group (VCEG) has been working on a new video coding standard with the internal denomination H.26L. In late 2001, MPEG's video group and VCEG decided to work together as a Joint Video Team (JVT), and to create a single technical design for a forthcoming ITU-T Recommendation and for a new part of the MPEG-4 standard based on the current working draft of JVT/H.26L [1][1] coding. Since the meeting in May 2002 the technical specification is almost frozen and the presented concepts in this work will very likely be part of the final standard. In addition to significant coding efficiency and simple syntax specification the third main design goal of the JVT/H.26L project was to allow a seamless and easy integration of the coded video into all current protocol and multiplex architectures – a goal summarized as "Network Friendliness". Transport protocols are very heterogeneous in terms of reliability, Quality-of-Service guarantees, encapsulation, and timing support. Moreover, transport systems provide differ in terms of internal setup and configuration protocol availability. Therefore, the JVT codec design distinguishes between two different conceptual layers, the Video Coding Layer (VCL), and the Network Abstraction Layer (NAL). Whereas the VCL mainly focuses on coding efficiency, the NAL provides the means to transport video data over a variety of networks.

In this work we present how JVT/H.26L coded video is transported over different networks. An overview of the JVT coding standard with special focus on the NAL features is presented. Packet-based and bit-stream oriented transport as well as the Parameter Set concept are discussed in more detail. As IP networks will carry a significant amount of JVT/H.26L video we focus in the second part on IP-based transport. It also serves as a concrete example on the integration of JVT/H.26L video into existing transport protocols.

[1] All referenced standard documents can be accessed via anonymous ftp at ftp://standard.pictel.com, ftp://ftp.imtc-files.org or ftp://ftp.ietf.org/.

## 2 TRANSPORT OF H.26L/JVT CODED VIDEO

### 2.1 Overview

According to Figure 1, the JVT codec design distinguishes between two different conceptual layers, the Video Coding Layer (VCL), and the Network Abstraction Layer (NAL). Both the VCL and the NAL are part of the JVT standard. Additionally, interface specifications are required to different transport protocols, to be specified by the responsible standardization bodies. Furthermore, the exact transport and encapsulation of JVT data for different transport systems, such as H.320, MPEG-2 Systems, and RTP/IP, are also outside the scope of the JVT standardization. The NAL decoder interface is normatively defined in the JVT coding standard, whereas the interface between the VCL and the NAL is conceptual and helps in describing and separating the tasks of the VCL and the NAL.

The VCL is not in the scope of this paper. The VCL specifies an efficient representation for the coded video signal. Coding tools, such as motion compensation, transform coding of coefficients, and entropy coding, are utilized. The highest level of representation in the VCL is a slice, which is a set of coded macroblocks for a picture. The NAL abstracts the VCL from the details of the transport layer used to carry the VCL data. It defines a generic and network independent representation for information above the level of the slice. Both, VCL and NAL are media-aware, i.e. they may know properties and constraints of the underlying networks, such as the prevailing or expected packet loss rate, MTU size, and transmission delay jitter. The VCL exploits this knowledge when it adjusts error resilience features, such as intra macroblock rate and coded slice size.

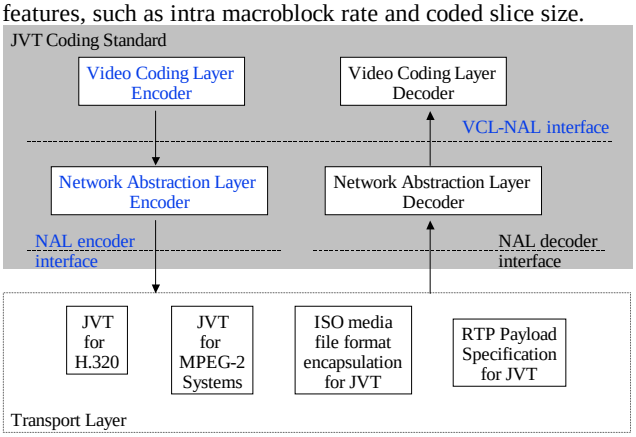


Figure 1 *JVT coding in transport environment*

Transport protocols are very heterogeneous in terms of reliability, Quality-of-Service guarantees, encapsulation, and timing support. Moreover, transport systems differ in terms of internal setup and configuration protocol availability. Hence, the NAL provides an abstract description of all the information to be conveyed by the JVT standard and also provides means to fulfill any kind of necessary network functionality within JVT by specifying the appropriate syntax. For example, H.320 systems do not provide encapsulation mechanisms whereas IP-based systems are entirely packet-based. In addition, the transport system may convey different NAL unit types on different logical channels with different QoS characteristics. The usage of setup and configuration protocols, such as SIP/SDP for IP-based applications and H.245 in H.324 systems, is relatively common. The mapping specification of JVT coded data to transport protocols is completely outside of the JVT standardization effort. However, on one hand, the NAL concept provides significant flexibility to integrate JVT coded data into existing and future networks, and, on the other hand, it also takes care of maintaining a sufficient common basis to facilitate gateway design between different transport layers. The NAL concept and the set of NAL functionalities will be discussed in the following in more detail.

## 2.2 Network Abstraction Layer Concept

The Network Abstraction Layer of JVT video defines the interface between the video codec itself and the outside world. It operates on Network Abstraction Layer Units (NALUs) which give support for the packet-based approach of most existing networks. At the NAL decoder interface it is assumed that the NALUs are delivered in transmission order and that packets are either received correctly, are lost, or an appropriate error flag in the NALU header is set if the payload contains bit errors. Any media-aware network element can raise this flag, if transmission errors were detected in the corresponding NALU. Therefore, erroneous payloads can be passed through a network, and the decoder or any gateway can decide whether this erroneous NALU is decoded or disposed.

A NALU consists of a one-byte header and a bit string that is, in most cases, the bits representing the macroblocks of a slice. The header byte itself consists of the aforementioned error flag, a disposable NALU flag, and the NALU type. The disposable flag can be used to signal whether the content of the NAL payload belongs to a picture that is not stored in the multi-picture buffer and allows the server, the network or any gateway to discard disposable NALUs without introducing error propagation. The NAL payload type either indicates the VCL "slice type" (e.g. an Intra Slice, P-Slice, or B-Slice), or high-level information, such as random access points, parameter set information, or supplemental enhancement information.

In addition to the transport of VCL data, the NAL includes additional features, which might partly be provided by underlying networks. Syntax and semantics of supplemental enhancement information is included in the NAL description. JVT has also considered a NALU aggregation scheme called compound packets and a fragmentation scheme to segment long NALUs into smaller pieces for transportation. It has to be clarified if there exist combinations of applications needing NALU aggregation and/or fragmentation and transport protocols not providing them or if the specification should be transport layer functionality as discussed in Section 3.2. Finally, the NAL provides means to transport high-level syntax, i.e., syntax which is assigned to more than one slice, e.g. to a picture, a group of pictures or to an entire sequence. As the applied parameter concept used in JVT is significantly different to previous video coding standards we will discuss it in greater detail in Section 2.4.

## 2.3 NAL for Bit-Stream Use

Systems using packet networks can employ NALUs directly, by using them as payloads for H.223 AL3 SDUs or RTP packets [11]. The mapping of NALUs to RTP packets is presented in detail in Section 3. However, some systems, such as the ITU-T Recommendation H.320 for videoconferencing and the MPEG-2 transport stream used in digital TV, are stream-oriented and therefore require a bit or byte stream format. Hence, the JVT standard defines a transformation of NALUs to such a format. NALUs are encapsulated by start codes, much in line with traditional video coding standards. The start code prefix is either 16 or 24 bits long, and its length depends on the "importance" of the payload of the NALU. Start code prefixes appear at byte-aligned positions only. Hence, a decoder can scan for start code prefixes and extract the NALUs by a simple, byte-oriented memory copy operation.

In order to prevent start code prefix emulations in the byte stream format, most video coding standards employ a carefully designed entropy coding. Since JVT video contains two different entropy coding modes, such a start-code-emulation-free environment would be very difficult to achieve. Instead, JVT relies on a byte-stuffing mechanism that inserts non-zero bytes into NALU bit strings in such positions where start code emulation occurs. In order to facilitate gateway designs, this byte stuffing is performed even in such environments where it seems to be unnecessary, especially in the packet environments. Note that as the VCL-NAL interface is only conceptual, the start code emulation prevention may also be implemented conventionally as part of the entropy coding of the VCL.

## 2.4 Parameter Set Concept

One of the key problems of robust video transmission in packet-lossy environments results from the layered nature of all current video coding standards. Information in the slice/picture/GOP/sequence headers is necessary for the reconstruction of the whole slice/picture/GOP/sequence, but traditionally coded only once at the start of each slice/picture/GOP/sequence in order to save bits. A loss of a packet containing a header renders all following packets containing data that rely on the lost header useless. Traditionally, three different strategies to handle this problem are used. The possible loss of this data might be ignored completely by transmitter and receiver and, therefore, all data relying on the lost header might be incorrectly interpreted. This leads to fatal consequences in even moderately error-prone environments. If the receiver is capable to recognize the loss of one or more packets, the video decoder might invoke concealment for the lost header. This works well in some environments, but becomes increasingly difficult the more information changes there are in the headers [20]. If the transmitter is aware of possible header losses, redundancy coding techniques are commonly used, especially at loss rates above 5%. Such techniques reside either in the video coding (MPEG-4 HEC [17] or H.263 Annex W [18], [20]), or in the RTP payload specifications (e.g. RFC 2429 [12]). Redundancy coding techniques add bit rate – sometimes to an amount that make error resilient video practically unfeasible – and also have a negative impact on the design complexity. A major disadvantage of all these approaches is that it is unpredictable whether the header data is accessible at the decoder or not. Repetition of the entire header

information in each and every transport packet is obviously too costly in terms of bit rate.

Therefore, a fundamentally new approach was taken in H.26L/JVT video. The synchronous, real-time media transmission consists entirely of packets that are completely self-contained. That is, each packet can be reconstructed without relying on information of other packets. The slice layer was identified as the appropriate smallest self-contained unit[2] (unless data partitioning is employed), because the size of slices can be adjusted to be small enough to fit into the MTU size of the most demanding system in this regard, namely H.324 mobile.

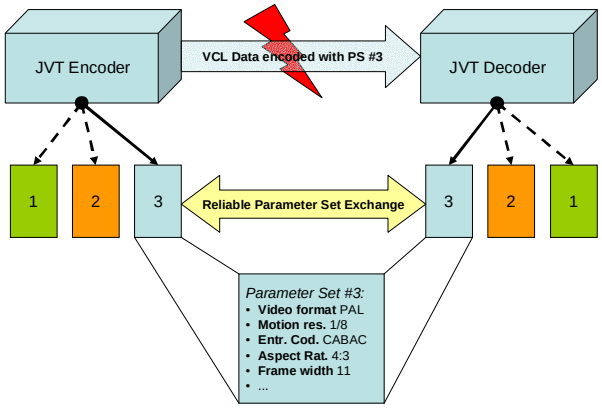


Figure 2 *The Parameter Set Concept*

All information at the higher layers is conveyed asynchronously. Of course, there are a few picture and GOP parameters that may change during the existence of a session. Parameters that change very frequently are added to the slice layer – this is in particularly true for the Frame Number (that signals the decoding order of pictures and is used as a picture identifier in multi-picture buffer management operations). All other parameters are collected in a so-called Parameter Set. Encoder and Decoder(s) maintain a synchronized set of such Parameter Sets. Which of the (potentially many) Parameter Sets the decoder should use to reconstructed on coded slice is defined by a codeword in the slice header of the VCL data. Figure 2 depicts this concept. The JVT standard specifies a method to convey Parameter Sets in a special NAL unit type, the Parameter Set information packet. The semantics are specified in the JVT standard, the syntax is still to be finalized. However, the Parameter Set concept allows using different logical channels or even a different out-of-band control protocols to convey parameter sets from the encoder to the decoder. In-band parameter set information and out-of-band control protocols should not be used in combination. As the high-level information included in Parameter Sets is usually only system-dependent, but not content-dependent, it is available in advance at the transmitter. Therefore, each parameter set can be transmitted in the session setup or during the session in an asynchronous and reliable way well before the synchronous video data references it. For example, SDP [15] can be used to define parameter sets that are conveyed reliably using SIP [14] or RTSP [21]. In H.323 and H.324 systems [19], the H.245 control protocol could be used to define and convey parameter sets.

[2] Note that at press time of this paper the JVT committee draft (CD) contains a slightly different concept. A picture layer is part of the CD, which can be sent either as a distinct and separate layer or as part of the slice layer. When this data is included at the slice layer, each slice becomes an independently decodable packet of data (relative to the content of other data for the current picture). There appears to be a significant chance that the picture layer will be removed as a distinct entity in the final draft and therefore we focus on the syntax structure that enables independent slice decoding, as this is the more robust structure of the two.

# 3 IP-BASED TRANSPORT

## 3.1 IP-Based JVT Applications

Previous video coding standards, such as H.261, MPEG-2, and H.263, were mainly designed for special applications and transport protocols usually in a circuit-switched or bit-stream oriented environment although they have been adapted to different transport protocols later on. JVT experts have taken into account transmission over packet-based networks in the video codec design from the very beginning. In addition to fixed IP networks, the JVT has considered IP-based video transmission over the third generation mobile systems. Applications that have been considered include streaming and conversational services, such as video telephony and videoconferencing.

The JVT acknowledged the importance of IP-based transmission over fixed and wireless networks by adopting a set of common test conditions for fixed and mobile IP based transmission, in [6] and [7], respectively. These test conditions allow for selecting appropriate coding features, testing and evaluating error resilience features, and producing meaningful anchor results. The common defined test case combinations include fixed Internet conversational services as well as packet-switched conversational services and packet-switched streaming services over 3G mobile networks. Also included is simplified offline network simulation software, which uses appropriate error patterns captured under realistic transmission conditions. Anchor video sequences, appropriate bit rates and evaluation criteria are specified. Recent results for fixed [3][5] and mobile [2] conversational services show the suitability of the JVT/H.26L coding standard to IP-based fixed and wireless applications.

## 3.2 RTP and RTP payload formats

Video over IP is usually implemented either by downloading complete bit streams, or by real-time transmission. The latter one for conversational or streaming applications over IP networks usually employs IP [8] on the network layer, UDP [9] on the transport layer, and RTP [10] and accompanying RTP payload specifications on the application layer.

IP and UDP together offer an unreliable datagram transport service. RTP adds to this functionality a few features that make the transport of media possible. Sequence numbers are used for restoring of the transmit order of out-of-order delivered IP packets at the receiver. RTP Timestamp are used for the synchronization of more than one media stream, e.g. to provide synchronization between audio and video. The timestamp is normally the sampling instant of the last bit of media in the RTP payload. It is not an absolute time, but coded relative to a random offset that is chosen during the session establishment. Receivers can self-tune themselves to identify the timestamp offset. The RTP Payload Type contains information about the media coding used by the payload. Some additional information can be conveyed by RTP to support extensions and multipoint operations through mixers and translators. Above IP, the media

bits may be encapsulated into a layer that hides some media-specific properties from RTP, and hides some RTP specific properties from the media coding. The documents specifying this "glue layer" are known as RTP payload specifications. For many simpler media coding schemes (e.g. for most sample based audio coding schemes), the RTP payload specifications do not add any information to the bit stream, but specify only the way how the media should be interpreted – e.g. that the timestamp should be 90 kHz.

For more complex media coding schemes such as most video coding schemes, however, the RTP payload specifications describe the whole operation of the packetization and depacketization process (including the fragmentation rules – the mapping of a bit-stream to packet boundaries), the timestamp, and any necessary header redundancy coding techniques, e.g. RFC2429 for H.263 [12] and RFC3016 for MPEG-4 [13].

One of the goals of the JVT was to design a simple coding scheme that can be encapsulated for transport in a straightforward manner. While the standardization process of both the JVT codec and the RTP payload specification for JVT [11] is still an ongoing process, it seems that the goal will be reached. The draft RTP payload specification expects that NALUs are transmitted directly as the RTP payload. No redundancy coding techniques for headers are necessary, because there are no headers above the slice header due to the Parameter Set concept.

The single additional concept introduced in the draft RTP payload specification is the so-called Compound Packet. A compound packet allows aggregating more than one NALU into a single RTP packet, by simply adding a 16-bit size information field of the following NALU at the end of the previous NALU. This technique is primarily helpful in gateway designs between networks with very different MTU sizes, because it allows saving the bits spent for the IP/UDP/RTP header overhead. Two forms of Compound Packets are currently under discussion. The simple form aggregates only packets with the same RTP timestamp (belonging to the same frame). The second form adds as administrative information an integer that codes the timestamp difference between the NALUs in the Compound packet. This more complex mechanism was introduced as a response to the demands of the streaming industry. For media streaming, especially at low bit rates, it is often advisable to put information belonging to more than one picture into the same packet. Since the JVT bit stream does not carry its own timing information, there is a need to handle such information externally, in the RTP timestamp.

## 4 CONCLUSIONS

In this paper, the current state of standardization of the Network Abstraction Layer of the JVT/H.26L video codec and its transport over RTP was outlined. The main new features of the NAL are the Parameter Set concept that decouples the (synchronous) video stream from the (asynchronous) transmission of picture header information, the introduction of Network Abstraction Layer Units and the definition of a bit-stream oriented video sytntax. A brief overview over an RTP packetization scheme was described. The authors and the JVT are confident that the transport issues presented will allow a simple integration of highly efficient JVT/H.26L coded video in many different networks and applications and will simplify gateway operations significantly.

## 5 REFERENCES


[1] T. Wiegand (ed.), "Committee Draft Number 1, Revision 0 (CD-1)," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-C167, May 2002.

[2] T. Stockhammer, M. Hannuksela, and T. Wiegand, "JVT/H.26L in 3G Wireless Environments", submitted for IEEE CSVT, Special Issue on H.26L/JVT, April 2002.

[3] S. Wenger, "H.26L over IP: The IP-Network Adaptation Layer", Proc. Packet Video Workshop 2002, Pittsburgh, PY, April 2002.

[4] S. Wenger, "Common Test Conditions for the H.323/Internet case," ITU-T Standardization Sector Q15-K45, October 1999.

[5] T. Stockhammer, T. Wiegand and S. Wenger, "Optimized transmission of H.26L/JVT coded video over packet-lossy networks," Proc. ICIP 2002, Rochester, NY, Sept. 2002.

[6] S. Wenger, "Common Conditions for wireline, low delay IP/UDP/RTP packet loss resilient testing", VCEG-N79r1, available from http://standard.pictel.com/ftp/video-site/0109_San/VCEG-N79r1.doc, September 2001.

[7] G. Roth, R. Sjöberg, G. Liebl, T. Stockhammer, V. Varsa, and M. Karczewicz, "Common Test Conditions for RTP/IP over 3GPP/3GPP2," ITU-T SG16 Doc. VCEG-N80, Santa Barbara, CA, USA, Sept. 2001.

[8] J. Postel, "Internet Protocol", RFC 791, September 1981.

[9] J. Postel, "User Datagram Protocol", RFC 768, August 1980.

[10] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.

[11] S. Wenger, T. Stockhammer, M. Hannuksela, "RTP payload Format for JVT Video", draft-wenger-avt-rtp-jvt-00.txt, Internet Draft, Work in Progress, February 2002.

[12] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", RFC2429, October 1998.

[13] Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata, "RTP Payload Format for MPEG-4 Audio/Visual Streams", RFC3016, November 2002.

[14] M. Handley, H.Schulzrinne, E.Schooler, and J.Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[15] M. Handley, V. Jacobson, "SDP: Session Description Protocol", RFC 2327, April 1998.

[16] Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T. and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

[17] ISO/IEC JTC1/SC 29/WG 11, "Overview of the MPEG-4 Standard (V.18)," ISO/IEC JTC1/SC 29/WG 11 N4030, March, 2001. Current version available at http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.htm.

[18] ITU-T, "Additional Supplemental Enhancement Information Specification", Annex W to ITU-T Recommendation H.263, Apr. 2001.

[19] D. Lindbergh, "The H.324 Multimedia Communication Standard," IEEE Communications Magazine, vol. 34, no. 12, pp. 46-51, Dec. 1996.

[20] M. Hannuksela, "H.263 Picture Header Recovery in H.324 Videophone", Proc. EUSIPCO 2000, Tampere, Finland, September 2000.

[21] A. Rao and R. Lanphier, "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.