A Survey of Adaptive Layered Video Multicast using MPEG-2 Streams

August Mayer and Hilmar Linder
Department of Scientific Computing
University of Salzburg, Austria
Email: [amayer | hlinder]@cosy.sbg.ac.at

***Abstract*— Adaptive Layered Video Multicast is a combination of layered video encoding and layered multicast. The layered video encoding method encodes (or transcodes) a video stream into multiple layers of data. With the layered multicast mechanism, these layers are transmitted over the network, automatically adapting to changes in the available channel resources, such as bandwidth, latency, or error rate. We present an overview over this technology, and some possible methods of using a pre-existing MPEG-2 bitstream with this transmission scheme. These methods do not require to fully decode and re-encode the bitstream, which results in considerably faster and less resource-intensive processing.**

***Index Terms*—Layered Multicast, Adaptive Video, Scalability, MPEG**

I. Introduction

In this paper, we will present methods to sub-divide pre-existing MPEG-1 or MPEG-2 video streams [4] into multiple layers, for layered multicast transmission. These methods are applied to the videos after encoding, in a separate step, whereas the traditional MPEG scalability features are applied during the encoding process. Thus, arbitrary pre-existing content in MPEG format can be used for layered video multicast, such as material from DVDs or TV broadcasts, without a need for resource-intensive and quality-decreasing stream transcoding. Also, it is possible to use one of the currently available, highly optimised MPEG encoders, which commonly do not provide scalability features per se, and to post-process this encoded stream for transmission over the network.

At current, videos are usually transmitted over the Internet via unicast. Multicast transmission is perceived to be not mature enough, and network providers do not offer multicast support for their end-user connectivity, nor do they commonly offer network Quality-of-Service (QoS) provisions necessary for real-time transmissions. There are also few concrete implementations of flexible multicast video distribution schemes, though there has been a lot of research in this area in recent years. However, the lack of support from providers can be overcome by using multicast "tunneling", multicast congestion control schemes such as WebRC [6] or FLID-DL [5], and schemes such as the ones presented below to prepare existing videos for distribution in a layered scheme.

Layered multicast distribution of video content has many advantages:

- Massive scalability. One layered video stream from the server can accommodate a large number of recipients, and more users can join the transmission without increasing the load. Schemes such as pyramid [7] or skyscraper broadcasting [8] allow the users to join the reception at virtually any time.
- Error resilience. FEC-based methods [10] such as rateless codes (for example, [11]; [9] presents an overview) allow to avoid retransmissions of lost packets; the information is recovered from additional received packets instead. Also, the transmission degrades gracefully if channel resources (bandwidth) are scarce, and the received video stream is still playable, albeit with a reduced quality. This is especially relevant for wireless networks, where the channel quality may change frequently.
- High efficiency. The video stream is only transmitted once over the network, and every client only receives as much data as it can handle. However, some encoding efficiency is lost because of the distribution to multiple video layers.
- Flexible. Multicasting permits sending a single video from multiple serving locations. Also, the same video transmission can accomodate diverse users or users with changing network connectivity, such as laptop and mobile phone users.

In principle, the presented methods can also be used with newer DCT[1]-based encoding formats such as MPEG-4 (DivX, H.264/AVC). There are additional problems in this case of usage, however. For instance, there are longer periods between intra-coded frames, which aggravates motion prediction errors introduced when dropping picture frames. Other new encoding features in these newer standards also introduce additional complexity and necessitate more functionality in the layer-generating code.

In the following chapters, we will first present more information about characteristics of Adaptive Layered Video Multicast, then present methods to provide temporal, qualitative and spatial layering of videos. We will present a short overview over methods of multicast transmission using advanced encodings, over our implementation, and finally conclude with a brief summary.

[1] Discrete Cosine Transform, see the MPEG2-Standard [4]

II. Adaptive Layered Video Multicast

Adaptive layered video multicast (ALVM) is the combination of layered encoding of video content, and a layered multicast transmission facility (Fig. 1). Transmission quality automatically adapts to the available resources, such as channel error rate or customer service subscription level. In case of scarce channel resources, for example due to TDD channel allocations or a high number of subscribers in a wireless segment, the quality also degrades gracefully.

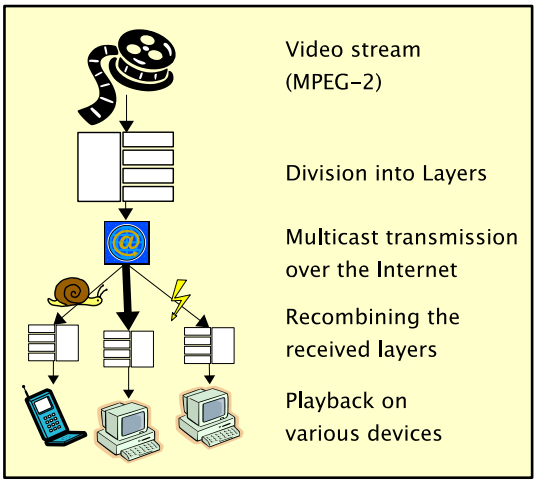

Fig. 1. Adaptive Layered Video Multicast

There are primarily three scalability methods to distribute MPEG video content onto several video layers: temporal, spatial, and qualitative or Signal-to-Noise-Ratio (SNR) scalability. With temporal scalability, the different video frames are distributed on different layers. Adding or dropping video layers thus affects the temporal resolution (frame rate) of the video. Spatial scalability distributes the pixels of each video frame, such that, for example, the base layer contains a movie with only the half frame size, and the enhancement layer delivers the remaining pixels for the full size. Finally, SNR scalability distributes qualitative picture components among the video layers, such as DCT coefficients. The effect is that the more layers are available at the decoding side, the more detailed the video will be after decoding. Combinations of these scalability methods allow for an even broader range of reception data rates and quality trade-offs.

Multiple video layers often depend on each other hierarchically. Layers of lower precedence successively extend the video information in the higher layers and can only be used if these higher layers have also been received. But there are also layering schemes where there are no such dependencies, for example with suitable spatial scalability methods. In this case, all layers can be used equally well to improve on the video quality. Such a non-hierarchical combination of scalability is more flexible and error-resilient, and any layer can be dropped in case of reception problems.

The methods presented below operate on existing MPEG-1/2 video streams and re-format them for use with layered multicast. Commonly, video scalability is achieved with a special encoder that divides a video stream into multiple layers, by writing multiple output streams, by annotating the output stream for a post-processing tool, or by defining error resilience checkpoints where a decoder can resume in case of missing data. But these features are rarely used, and most highly-optimized MPEG-1/2 encoders do not even implement them at all. Also, pre-existing video streams without these special provisions need to be re-encoded, which is resource-intensive and potentially degrades the video quality. In contrast, our methods avoid this re-encoding, with the price of additional, usually minor quality degradation in case of missing data.

III. Temporal Scalability

The MPEG standard [4] defines multiple flavors of frames: I-frames, P-frames, and B-frames. I-frames ("Intra-coded frames") contain all the data for a given frame and do not need other frames for decoding. P-frames ("Predictive") contain only the delta to another I-frame and thus need this frame's information for decoding. B-frames ("Bidirectionally predictive") even rely on two other I- or P-frames for correct decoding (but not on other B-frames). Temporal Scalability works by distributing these frames onto several transmission layers in various ways.

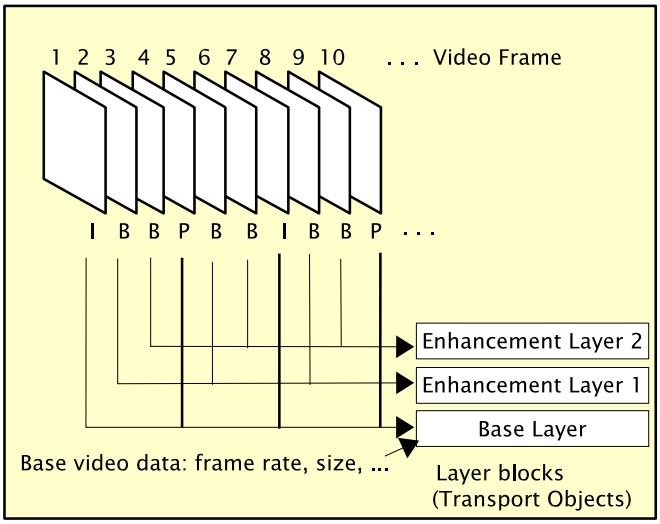

Fig. 2. Sample layering scheme using temporal scalability, by distributing the B-frames over multiple layers.

For example, the I- and P-frames could be placed on the base layer, and the B-frames on the single extension layer (this is used in [1] for simplicity). However, this does not produce a very continuous scaling behaviour. In MPEG-2 videos, there are usually 3 or 4 P-frames per I-frame, and two B-frames per I- or P-frame, creating a frame sequence like "IBB PBB PBB PBB IBB PBB PBB PBB ...". Consequently, such an encoded video stream consists of about 2/3 B-frames, and 1/3 I- and P-frames (for the actual data sizes, see also Fig. 5 below). If the B-frame layer needs to be dropped, the video's frame rate

decreases to 1/3 of the original value, which is commonly 25 frames/sec / 3 $\approx$ 8 frames/sec.

A better, but still simple method would be to distribute only the B frames onto several layers. So for example, every even B-frame could be put onto extension layer 1, and every odd B-frame onto layer 2, as shown in Fig. 2. The base layer still contains the base video data and the I- and P-frames. Also, the resulting two extension layers do not depend on each other, and either one could be dropped if necessary.

Evaluation

There is only little additional overhead required to be able to correctly re-combine the layers, such as frame sequence number or frame length. There is also little additional processing power needed, as the receiver simply re-sorts the frames received on each multicast layer into the single output stream. In case of dropped layers, replacement frames must be inserted for the lost video frames to keep the frame rate constant; for MPEG movies, this can be achieved simply by inserting frames that re-display the previous frame. However, the video becomes more and more choppy as more layers and thus frames are dropped.

Also, missing updates for the motion compensation predictors create additional artifacts. Motion compensation is a method to describe in P- and B-frames where pixel blocks from the reference frame should be moved when copying from the reference frame. This reduces the amount of required delta information considerably (for example, this is effective in scenes when the camera slowly pans across a view, or where objects such as cars slowly move through the picture). Motion compensation information is further condensed by using a prediction model; movements are assumed to be continous, and only the difference to this prediction is stored in the motion compensation updates in each frame. When such updates are missing, blocks will be copied to the wrong places, resulting in artifacts.

However, these additional visual errors turn out to be tolerable. The encoded motion delta is usually small, and motion compensation is restarted frequently at every I-frame. This problem is worse for the newer standards like MPEG-4, with longer stretches of predictive frames between the intra-coded frames.

IV. Qualitative (SNR) scalability

Video streams can also be separated into layers by distributing qualitative factors, such as, for MPEG, the encoded DCT coefficient information (Fig. 3).

For example, the base layer could contain the audio and the basic video data. There could be ten extension layers which would each contain 10% of the encoded coefficients. To separate the coefficients into these layers, the sender needs to parse the MPEG data slices and extract the bit positions and lengths of the coefficient runs[2]. Therefore, this method

[2]The DCT coefficients are compressed with an RLE (Run Length Encoding) method, with multiple bit runs per set of coefficients. For details, please see the MPEG specification [4] for details.

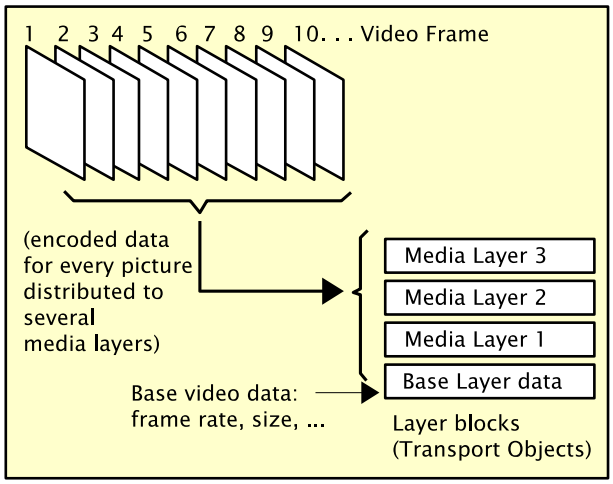

Fig. 3. Layering using Qualitative Scalability.

is more complicated and resource-intensive than the temporal scalability layering presented above, but it has the advantage of scaling much more smoothly. If only some coefficients are removed, video quality will be nearly identical to the original. If more layers are dropped, the amount of detail of the pictures will degrade smoothly and decoding artifacts increase slowly. This behaviour is also more similar to that of traditional TV broadcasts, where degrading signal reception quality continuously introduces more noise to the picture.

Evaluation

This method is quite flexible, and it is possible for the receivers to tune the reception data rate accurately to the available resources. As only about half of the video stream data consists of the DCT coefficients, SNR scaling needs to be combined with other methods such as temporal layering to effect larger reductions of the video size.

There is also some additional encoding overhead. Additional marker bits (two or four, depending on the encoding mode) are required for the base and every enhancement layer with coefficients to signal the end of the data on this layer. For example, if there are 10 enhancement layers with enough data for every layer, the additional data would be at least $10 \cdot 2 = 20$ bits. Also, additional computing power is required for parsing the MPEG picture data structures at both the sender, to find the coefficients to distribute, and also the receiver, for minimal overhead.

With this method of qualitative scalability, the layers depend on each other hierarchically. When higher layers are dropped, the effect is that some coefficient runs towards the end of the encoded frequency scale are lost. This results in gradually more decompression artifacts and less image detail, until the image only consists of the base layer information, which is basically an array of single-color 8×8 pixel blocks.

V. Spatial scalability

Spatial layering is a method which distributes the pixels of the video frames over the transmission layers, creating

multiple layers with partial resolution. For example, two layers could be created by putting every even pixel on the first layer, and every odd on the second, as shown in Fig. 4.

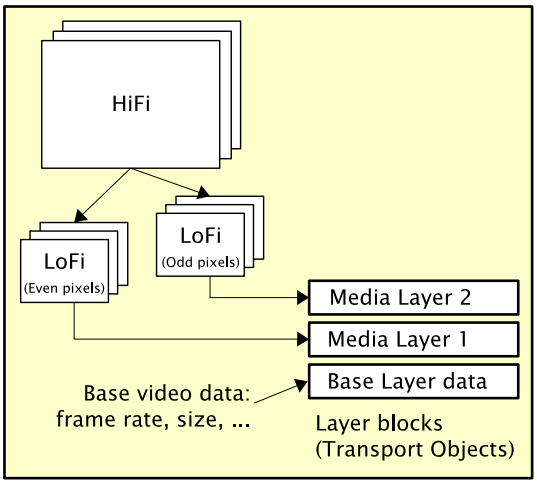

Fig. 4. Layering using Spatial Scalability.

This layering method operates in the spatial domain, whereas MPEG videos are encoded in the frequency domain. However, there are methods to re-scale an image in the frequency domain (presented for example in [2] and [3]), which is much less resource intensive computationally than full video transcoding.

Evaluation

The advantages of this spatial layering method are that it allows to decrease the base layer data rate of the video layer even further. The layers do not (necessarily) depend on each other and can be decoded independently, in a non-hierarchical way. Still, it is considerably more complex and resource intensive than the temporal and qualitative layering methods presented above; in addition to decoding the macroblocks and DCT coefficients, it also requires a number of mathematical transformations at both the sender side, to separate every frame apart, and at the client side, to recombine them if enough layers are available.

If there are insufficient resources, and consequently some dropped layers, the client requires additional computational power for scaling the video pictures up to the full size. However, for devices such as mobile phones, the availability of a scaled-down version could even be an advantage. Such devices may be unable to process the full rate anyway, or they may have a small target display for which the video would have to be downscaled in any event.

VI. Transmission over the Network

After the video has been distributed onto several video layers, every video layer is assigned to a corresponding multicast layer, which are then distributed over the network. A client only receives as many layers as its resources allow (network, computational, or other), and adapts to changes in these conditions automatically. To achieve fairness to other traffic on the same line, such as TCP, and to not disturb other multicast traffic, additional technologies need to be employed, such as multicast congestion control schemes and FEC encoding methods.

For good real-time performance, network Quality-of-Service (QoS) provisions would be optimal. However, such QoS mechanisms are uncommon at current, and usually proprietary to specific vendors. They are usually not available for end-users, and the infrastructure is not easy to configure and maintain. Instead, it is necessary to use client- and/or server-driven multicast congestion control schemes, which do not need provider support, but operate only with software agents at the multicast senders and receivers. Two prominent schemes are FLID-DL [5] and WebRC [6]. However, such schemes have some drawbacks, because they use the network as an opaque medium that does not act autonomously.

To improve the behaviour of multicast traffic in the case of transmission errors, FEC-based transmission methods [9] [10] using Erasure Codes such as the rateless Online Codes [11] have been developed. These are special encodings of the layer data sent over the multicast channels, with the advantage of a drastically increased error resilience. There is only a small amount of additional data, and the methods do not use retransmissions, which have generally turned out to be not very efficient for multicast. Also, packet ordering is not important with these schemes; a video chunk can be decoded when a large enough amount of data blocks has been received, independent of which blocks these are. As a consequence, there does not have to be a separate channel for receivers with a slower connection, because it is sufficient that the router before the slow connection drops random excess packets.

VII. Implementation

For testing the characteristics of the presented adaptive layered video multicast methods, we have created a prototype tool called avcast. It distributes videos over layered multicast, using temporal or qualitative scalability. With temporal scalability, avcast currently separates a video into four layers containing base data (structure and audio), I-Frames, P-Frames and B-Frames (Figure 5).

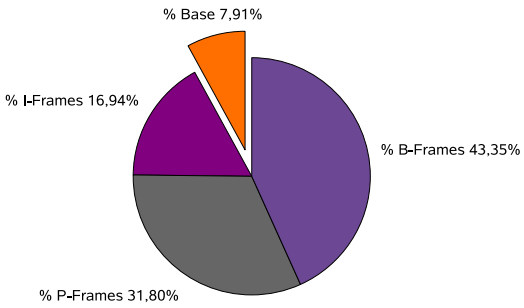

Fig. 5. Example for the relative layer sizes for temporal scalability with avcast. Experimental results from a clip of the DVD version of “Contact”.

Qualitative scalability is implemented as outlined in chapter III above, by parsing the MPEG video data and distributing the encoded DCT coefficients over multiple layers. For layered multicast distribution, we use the method presented by [1], which transmits video chunks of one minute length over the network.

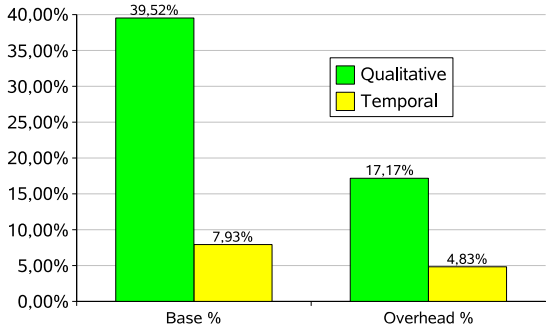

Fig. 6. Typical overhead and base layer sizes for temporal and qualitative scalability with avcast, relative to the video size. Results also from the "Contact" video clip.

As expected, the transmission overhead is tolerable, but it is necessary to transmit relatively much data on the base layer (see also Figure 6); about 40% are needed for our qualitative scalability method. Better results should be attainable with a combination of temporal and qualitative scalability, which we intend to examine as a next step.

VIII. Conclusion and Outlook

In this paper, we have presented some methods to distribute pre-encoded MPEG-1 and MPEG-2 video data onto several video layers, for transmission over multiple multicast channels. Layered multicast distribution has many advantages, such as better scalability, more possible receiver diversity, and better error resilience. Dividing the video into layers as a separate step after encoding is more complex, but offers the advantage of being able to use existing video data and highly-efficient standard encoders. In principle, the presented methods can also be applied to newer MPEG video coding methods, such as MPEG-4 AVC [13]. However, the additional compression features of these encodings increase the complexity of the layering step considerably, and bear the risk of even more visual errors in case of dropped layers.

In recent years, there has been a lot of research activity in the field of layered video multicast. Many ways of providing scalability and layering features for encoded video have been examined thoroughly, and there has also been a lot of activity in the field of layered multicast transmission. It seems, though, that there are only a few experimental solutions that combine the two topics. Moreover, there is currently no standard protocol for layered video multicast transmissions, which would enable different software or hardware agents to interoperate.

To fill this gap, an interesting approach would be to adapt the IETF FLUTE protocol [12] for this application area. FLUTE is primarily designed for efficiently sending files over uni-directional links; it could be used to carry the signaling information needed to automatically detect and use the different media layers at the receivers. It could perhaps also be used to distribute the actual video stream layers. However, real-time video transmissions have some special characteristics, such as that packets are usually rather short, and that timing and transmission latency are important factors. Video packets that would arrive "too late" are not useful and could be discarded inside the network. Therefore, video stream data could also be sent over multicast RTP (Real-time Transport Protocol, see [14]) streams, if network QoS provisions are available. Both methods would be one further important step towards a standard for efficient transmission of videos over the Internet.

Acknowledgment

This work was funded by the European Commission 6th Framework programme IST Project BROADWAN[3].

References

[1] Ch. Neumann and V. Roca, *Scalable Video Streaming over ALC (SVSoA): a Solution for the Large Scale Multicast Distribution of Videos*, Research Report, INRIA Rhône-Alpes, Planète project, March 2003.

[2] R. Dugad and N. Ahuja, *A Fast Scheme for Image Size Change in the Compressed Domain*, IEEE Trans. on Circuits and Systems for Video Technology, Vol.11, No.4, pp.461-474, April 2001.

[3] S. Suh, S. S. Chun, M. Lee and S. Sull, *Efficient image down-conversion for mixed field/frame-mode macroblocks*, Proceedings of the International Conference on Image Processing (ICIP) 2003, Vol. 1, 14–17, pp. 177–180, Sept. 2003.

[4] ISO/IEC Standard 13818-2, *Generic coding of moving pictures and associated audio information: Video*, Revision 2000-12-21 (This is commonly referred to as the MPEG-2 video standard).

[5] J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter and W. Shaver, *FLID-DL: Congestion Control for Layered Multicast*, Proceedings of NGC 2000, pages 71–81, November 2000. http://citeseer.ist.psu.edu/byers00fliddl.html

[6] M. Luby and V. Goyal, *Wave and Equation Based Rate Control (WEBRC) Building Block*, RFC 3738, Internet Engineering Task Force (IETF), April 2004.

[7] S. Viswanathan and T. Imielinski, *Pyramid Broadcasting for Video-on-Demand Service*, Proceedings of the SPIE Multimedia Computing and Networking Conference, Vol. 2417, pp. 66–77, San Jose, CA, February 1995.

[8] K. A. Hua and S. Sheu, *An Efficient Periodic Broadcast Technique for Digital Video Libraries*, Multimedia Tools and Applications, Vol. 10, Number 2/3, pp. 157–177. 1998.

[9] Ch. Neumann and V. Roca, *Analysis of FEC Codes for Partially Reliable Media Broadcasting Schemes*, 2nd ACM International Workshop on Multimedia Interactive Protocols and Systems (MIPS'04), Grenoble, France, Nov. 2004.

[10] L. Rizzo, *Effective Erasure Codes for Reliable Computer Communication Protocols*, Computer Communication Review, 27(2):24–36, April 1997.

[11] P. Maymounkov, *Online Codes*, Technical Report TR2002-833, New York University, Nov. 2002.

[12] T. Paila, M. Luby, R. Lehtonen, V. Roca and R. Walsh, *FLUTE – File Delivery over Unidirectional Transport*, RFC 3926, Internet Engineering Task Force, October 2004.

[13] ISO/IEC Standard 14496-10:2004, *Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*, Revision 2004-09-28 (This is commonly referred to as MPEG-4 AVC or H.264/AVC).

[14] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, *RTP: A Transport Protocol for Real-Time Applications*, RFC 3550, Internet Engineering Task Force, July 2003.

[3] http://www.broadwan.org